Secure Transmission in Wireless Semantic Communications With Adversarial Training

Muhammed Yasinhan Yaşar
Computer Science
Ankara Yıldırım Beyazıt University
Ankara, Turkey
myyasar2001@gmail.com

Abstract—The Burgeoning technology of deep learning based semantic communications has significantly enhanced the efficiency and reliability of wireless communication systems by facilitating the transmission of semantic features. However, security threats, notably the interception of sensitive data, remain a sigfnificant challenge for secure communications. To safeguard the confidentiality of transmitted sematics and effectively counteract eavesdropping threats, this letter propose a secure deep learning-based semantic communication system, SecureDSC. It comprises semantic encoder/decoder, channel encoder/decoder, and encryption/decryption modules with a key processing network.

Index Terms—Semantic communications, symmetric encryption, secure transmission

I. Introduction

DEEP learning-based semantic communications have attracted much attention due to the focus on transmisson and interaction at the smeantic level [1]. Compared to traditional communication systems, sematic communications achieve higher transmission efficiency and lower bandwith with less susceptible to noise or other interference [2]. Nevertheless, the inherent openness and accessibility of wireless channels present a significant security risk to extracted semantic features transmitted in plaintext, enabling eavesdropping attacks and resulting in the disclosure of sensitive information. With the development of semantic communications, ensuring secure transmission is crucial for fostering trustworthy interactions within semantic communication systems and reliable evolution of wireless communication technologies.
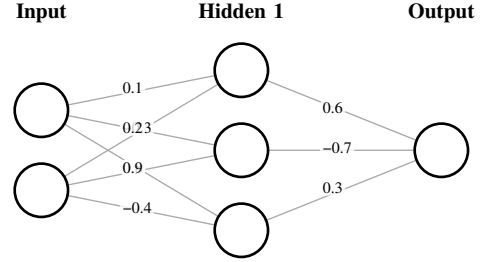
Fig. 1. The model of proposed secure text semantic communication system.

II. System Model

As shown in Fig. 1, Alice sends a message $\mathbf{m}$ to the legitimate receiver Bob, and both of them use the same shared session key $\mathbf{k}$ for encryption and decryption. The eavesdropping attacker Eve aims to reconstruct the original information communicated between legitimate users. As a black-box attacker, Eve lacks the knowledge of specific parameters used in the models employed by Alice and Bob. However, it can eavesdrop and analyze messages transmitted within the channel and leverage a large number of input-output pairs to invert the original network for private information.

Without loss of generality, we take the example of transmitting a text message i.e., $\mathbf{m} = [w_1, w_2, ..., w_L]$ where w_l denotes the lth word in the sentence. At the sender end, the message passes through a semantic encoder to obtain a semantic feature $\mathbf{f}$, then $\mathbf{f}$ is input into a cipher encoder to generate ciphertext $\mathbf{c}$, and $\mathbf{c}$ is converted into asymbol stream $\mathbf{x}$ using a channel encoder, which is expressed as

$$\mathbf{x} = Ce_\gamma\big(En_\beta(Se_\alpha(\mathbf{m}), K_\kappa(\mathbf{k}))\big), \tag{1}$$

where $Se_\alpha(\cdot)$ is the semantic encoder network with parameter set $\boldsymbol{\alpha}$, $En_\beta(\cdot)$ is the encrypt network with parameter set β, $Ce_\gamma(\cdot)$ is the channel encoder network with parameter set γ and $K_\kappa(\cdot)$ is key processing network with parameter set κ.

As a legitimate receiver, Bob receives the signal $\hat{\mathbf{y}}$ with the interference and noise which is represented as

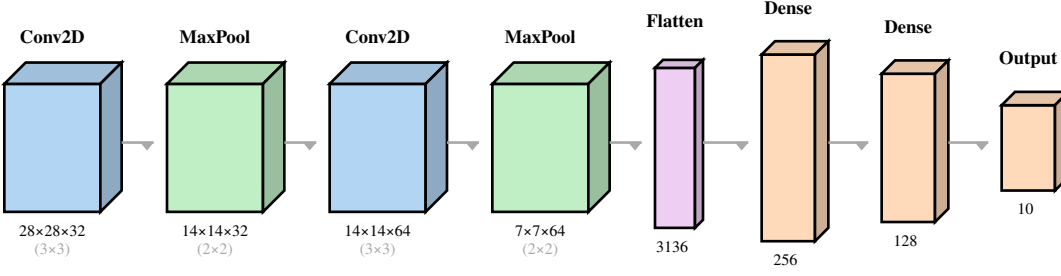$$\hat{\mathbf{y}} = h\mathbf{x} + \mathbf{n}, \tag{2}$$

Fig. 2.
The whole neural network and components for the proposed SecureDSC system.

where h is the channel coefficient assumed to be constant for a quasi-static channel, and $\mathbf{n}$ is the additive white Gaussian noise (AWGN) with a mean of zero and a variance σ_n^2. With the support of channel, chiper and semantic decoders, $\hat{\mathbf{y}}$ is decoded to reconstruct the original message as

$$\hat{\mathbf{m}} = Sd_{\boldsymbol{\theta}}\big(De_{\boldsymbol{\delta}}(Cd_{\boldsymbol{\chi}}(\hat{\mathbf{y}}), K_{\boldsymbol{\kappa}}(\mathbf{k}))\big), \tag{3}$$

in which $Sd_{\boldsymbol{\theta}}(\cdot)$ is the semantic decoder network with parameter set $\boldsymbol{\theta}$, $De_{\boldsymbol{\delta}}(\cdot)$ is the decrypt network with parameter set $\boldsymbol{\delta}$ and $Cd_{\boldsymbol{\chi}}(\cdot)$ is the channel decoder network with parameter set $\boldsymbol{\chi}$. Moreover, it is straightforward that Alice and Bob utilize the same key processing network $K_{\boldsymbol{\kappa}}(\cdot)$ to obtain the same session key in the symmetric encryption.

The atacker Eve eavesdrops on the open wireless channel, intercepting the transmitted signal $\bar{\mathbf{y}}$. As the session key $\mathbf{k}$ secretly shared between legitimate users is inaccessible, Eve replaces it with a random number $\mathbf{r}$ to reconstruct the original message as

$$\bar{\mathbf{m}} = Sd_{\bar{\boldsymbol{\theta}}}\big(De_{\bar{\boldsymbol{\delta}}}(Cd_{\bar{\boldsymbol{\chi}}}(\hat{\mathbf{y}}), \mathbf{r})\big). \tag{4}$$

It is noteworthy that despite the similar network structure between Bob and Eve, the parameter sets differ due to Eve's absence from the joint training between Alice and Bob.

III. Propesed SecreDSC System

This section presents the basic model of the proposed semantic communication system SecureDSC, including the composition of the network layers and the procedures of encryption and decryption. We then discuss the design of loss function and the adversarial training process that provides the confidentiality protection of transmitted messages.

A. Basic Model

Fig. 2 illustrates the whole sturcture of the neural network. Tje embeding layer first performs a transformation on the input message $\mathbf{M}$, mapping it to a dense vector representation. Then, the semantic encoder $Se_{\boldsymbol{\alpha}}(\cdot)$ extracts semantic features from the embedded input. Subsequently, extracted features $\mathbf{F}$ and keys $\mathbf{K}$ serve as inputs for the encryptor $En_{\boldsymbol{\beta}}(\cdot)$ to generate the ciphertext $\mathbf{C}$. Before being transmitted over the channel, the ciphertext is input into the channel encoder $Ce_{\gamma(\cdot)}$

to generate symbol streams. The network on the receiver end is symmetrical to that on the transmitter end, except for a prediction layer with Softmax as the activation function at the end for outouttşng the decoded results $\mathbf{M}$
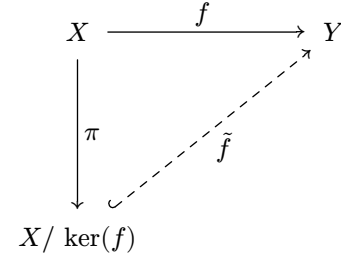

Fig. 3.
The detailed encryption and decryption process and network interconnection schematics.

B. Loss Function Design and Training Process

Given that the data is coveyed in textual format, cross-entropy (CE) is employed as the loss function to quantify the discrepancy between two sentences as follows

$$\begin{aligned}&\mathcal{L}_{CE}(\mathbf{m}_1, \mathbf{m}_2)\\&= -\sum_{l=1} p(w_l)\log q(w_l) + (1 - p(w_l))\log(1 - q(w_l))\end{aligned} \tag{5}$$

where $p(w_l)$ and $q(w_l)$ denote the real probability of the lth word w_l appearing in message $\mathbf{m}_1$ and the predicted probability of it appearing in $\mathbf{m}_2$, respectively.

To accomplish successful transmission and fulfill aforementioned goals, we perform joint and independent training of net, modules for legitimate transmitters and receivers. Additionally, we devis an adversarial training mechanism that account for the impact of adversaries throughout the training process.

For the independent training of the semantic encoder and decoder, the loss function is designed as

$$\mathcal{L}_{sem}(\mathbf{s}, \hat{\mathbf{s}}; \boldsymbol{\alpha}, \boldsymbol{\theta}) = \mathcal{L}_{CE}(\mathbf{s}, Sd_{\boldsymbol{\theta}}(Se_{\boldsymbol{\alpha}}(\mathbf{s}))), \tag{6}$$

where $\mathbf{s}$ is the batch data. Likewise, to train the encryptor, decryptor, and the key processing module independently, the loss function follows the subsequent expression

$$\mathcal{L}_{cip}(\mathbf{s}, \hat{\mathbf{s}}; \mathbf{B}, \boldsymbol{\delta}, \boldsymbol{\kappa})$$
$$= \mathcal{L}_{CE}(\mathbf{s}, De_{\delta}(En_{\mathrm{B}}(\mathbf{s}, K_{\kappa}(\mathbf{k})), K_{\kappa}(\mathbf{k}))). \quad (7)$$

Hence the basic modules in the network can operate independently using the aforementioned two loss functions.

At the same time, to jointly train the entire network, the loss function for the legitimate receiver Bob is designed as

$$\mathcal{L}_B(\mathbf{m}, \hat{\mathbf{m}}; \boldsymbol{\chi}, \boldsymbol{\delta}, \boldsymbol{\theta}, \boldsymbol{\kappa}) = \mathcal{L}_{CE}(\mathbf{m}, \hat{\mathbf{m}}), \quad (8)$$

where $\hat{\mathbf{m}}$ is the received message defşned in (3). Considering the adversary Eve in the channel with the sane network structure as the legitimate receiver, the loss function is defined similarly as

$$\mathcal{L}_E\left(\mathbf{m}, \overline{\mathbf{m}}; \overline{\boldsymbol{\chi}}, \overline{\boldsymbol{\delta}}, \overline{\boldsymbol{\theta}}, \overline{\boldsymbol{\kappa}}\right) = \mathcal{L}_{CE}(\mathbf{m}, \overline{\mathbf{m}}). \quad (9)$$

Utilizing adversarial cryptography, we combine the above two loss functions to design the joint loss function as

$$\mathcal{L}_B(\mathbf{m}, \hat{\mathbf{m}}, \overline{\mathbf{m}}; \alpha, \beta, \gamma, \boldsymbol{\chi}, \boldsymbol{\delta}, \boldsymbol{\theta}, \boldsymbol{\kappa})$$
$$= \mathcal{L}_B + \mathrm{abs}(\mathcal{L}_E - \lambda) \quad (10)$$

where $\mathrm{abs}(\cdot)$ is the absolute value function and λ is the hper parameter for adversarial training to restrict Eve's ability of information reconstruction.

As shown in Algorithm 1, the entire network is updated sequentially according to the above loss functions. Simce the primary focus of the SecreDSC system is to protect the confidentiality of semantic features, the training of channel codecs is ommited here and the parameters are updated throughout the training process. In this algorithm, the keys shared between Alice and Bob are randomized for each message, awhile Eve replaces them with random sequences as mentioned above.

Algorithm 1. Training Algorithm of SecureDSC

1: **procedure** ENCRYPT_DECRYPT(A, n, v)
2:    ▷ Initialize the search range
3:    $l \leftarrow 1$
4:    $r \leftarrow n$
5:
6:    **while** $l \leq r$ **do**
7:      mid $\leftarrow$ **floor**$(\frac{l+r}{2})$
8:      **if** $A[\mathrm{mid}] < v$ **then**
9:        $l \leftarrow \mathrm{mid} + 1$
10:      **else if** $A[\mathrm{mid}] > v$ **then**
11:        $r \leftarrow \mathrm{mid} - 1$
12:      **else**
13:        **return** mid
14:      **end**
15:    **end**
16:    **return null**
17: **end**

IV. PERFORMANCE EVALUATION

A. Performance Evaluation of the Propesed SecreDSC

REFERENCES

[1] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Y. Li, "Semantic Communications: Principles and Challenges." [Online]. Available: https://arxiv.org/abs/2201.01389

[2] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep Learning Enabled Semantic Communication Systems," *IEEE Transactions on Signal Processing*, vol. 69, no. , pp. 2663–2675, 2021, doi: 10.1109/TSP.2021.3071210.