Short Blocklength Secret Coding via Helper-Assisted Learning over the Wiretap Channel

Vidhi Rana
University of Texas at Arlington
Arlington, TX
Email: vidhi.rana@uta.edu

Rémi A. Chou
University of Texas at Arlington
Arlington, TX
Email: remi.chou@uta.edu

Taejoon Kim
University of Kansas
Lawrence, KS
Email: taejoonkim@ku.edu

***Abstract*—Consider the Gaussian wiretap channel, where a legitimate transmitter wishes to send a confidential message to a legitimate receiver in the presence of an eavesdropper. Unfortunately, in this setting, it is well known that if the eavesdropper experiences less channel noise than the legitimate receiver, then it is impossible for the transmitter to achieve positive secrecy rates. A known solution to this issue consists in involving a second transmitter, referred to as a helper, to help the first transmitter to achieve security. While such a solution has been studied for the asymptotic blocklength regime and via non-constructive coding schemes, in this paper, for the first time, we design explicit and short blocklength codes using deep learning and cryptographic tools to demonstrate the benefit and practicality of cooperation between two transmitters over the wiretap channel. Specifically, our proposed codes show strict improvement in terms of information leakage compared to existing point-to-point codes that do not consider a helper, even when the transmitter has adverse channel conditions, in the sense that the eavesdropper experiences less channel noise than the legitimate receiver. Our code design approach relies on a reliability layer, implemented with an autoencoder architecture inspired by the successive interference cancellation method developed for broadcast channels, and a security layer implemented with universal hash functions.**

I. Introduction

Physical layer security exploits physical characteristics of the wireless channels to transmit confidential messages. An information theoretic approach to physical layer security has been first proposed by Wyner with the wiretap channel model in [1], where a transmitter wishes to send a confidential message to a legitimate receiver in the presence of an eavesdropper. Unfortunately, if the channel conditions are such that the channel gain between the transmitter and the eavesdropper is better than the channel gain between the transmitter and the legitimate receiver, then the secrecy capacity is zero, meaning that it is impossible for the transmitter to send a secret message to the legitimate receiver [1], [2].

To overcome this impossibility, settings that involve multiple users need to be considered to go beyond point-to-point transmission, and corresponding codes need to be designed for such settings. Works in this direction include the Gaussian multiple-access wiretap channel [3], [4], where multiple users communicate secret messages with the receiver in the presence of an eavesdropper, and the helper-assisted wiretap channel, e.g., [5], [6], where the transmitter can benefit from the help of jammers. Specifically, these works demonstrate that cooperation between transmitters can be beneficial to enable positive secrecy rates at the transmitters who could not achieve positive secrecy rates with point-to-point codes.

While existing works have mainly focused on the asymptotic blocklength regime and non-constructive coding schemes to derive achievability secrecy rates, in this paper, we propose to design explicit and short blocklength codes (< 24) for the Gaussian wiretap channel in the presence of a helper that cooperates with the transmitter. Our constructed short blocklength codes demonstrate the benefit and practicality of user cooperation over a wiretap channel as follows:

- We show that for a transmitter with adverse channel conditions (i.e., his channel gain with the eavesdropper is better than his channel gain with the legitimate receiver), a second transmitter, called a helper, can help decrease information leakage at the eavesdropper. Note that such a result is impossible to achieve by solely relying on point-to-point Gaussian wiretap channel codes and therefore new codes that enable cooperation need to be designed.
- We show that for a transmitter with favorable channel conditions (i.e., his channel gain with the eavesdropper is worse than his channel gain with the legitimate receiver), a helper, can also help decrease information leakage at the eavesdropper.

Our proposed framework for code designs decouples the reliability and secrecy constraints. Specifically, our codes rely on one reliability layer and one security layer that can be designed separately to allow a flexible code design at short blocklength. The reliability layer is implemented with an autoencoder architecture inspired by the well-known successive interference cancellation (SIC) idea, first introduced for broadcast channels in [7], and the security layer is implemented with universal hash functions.

Related works: To the best of our knowledge, prior to the present work, no finite-length code constructions have been proposed for cooperating users over a wiretap channel under an information-theoretic leakage security metric, i.e., information leakage at the eavesdropper is measured in terms of the mutual information between the confidential message and the eavesdropper's channel observations. The most related works are point-to-point wiretap code constructions at finite blocklengths under the same security metric and include (i)

978-1-7281-9054-9/24/$31.00 ©2024 IEEE

coding-theoretic constructions such as [8], which relies on punctured systematic irregular LDPC codes, [9], which utilizes LDPC codes, and [10], which relies on randomized Reed-Muller codes, and (ii) deep learning based constructions such as [11], whose approach consists in training an autoencoder to optimize the reliability and secrecy constraints simultaneously, and [12]–[14], whose approach aims to separately handle the reliability constraint via an autoencoder and the secrecy constraint via hash functions for better modularity and a fine information leakage control.

The remainder of the paper is organized as follows. Section II introduces the Gaussian wiretap channel model with a helper. Section III describes our proposed code design, and Section IV presents our code performance evaluation for the Gaussian wiretap channel model with a helper. Finally, Section V provides concluding remarks.

II. Model

As depicted in Figure 1, we consider a Gaussian wiretap channel with a helper defined by

$$Y = \sqrt{h_1}X_1 + \sqrt{h_2}X_2 + N_Y, \tag{1}$$
$$Z = \sqrt{g_1}X_1 + \sqrt{g_2}X_2 + N_Z, \tag{2}$$

where h_1 and h_2 are the channel gains of the transmitter and helper, respectively, to the intended receiver, g_1 and g_2 are the channel gains of the transmitter and helper, respectively, to the eavesdropper, and N_Y and N_Z are zero-mean Gaussian random variables with variances σ_Y^2 and σ_Z^2, respectively. The legitimate receiver and the eavesdropper observe the sequences Y^n and Z^n, respectively, given by (1) and (2), and all the above channel parameters are known to everyone. In this

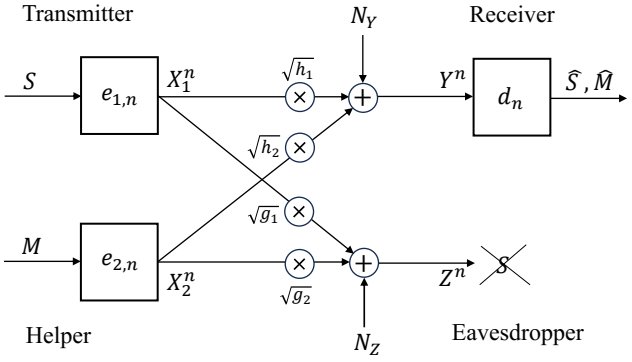


Figure 1: Gaussian wiretap channel with a helper.

model, the transmitter wishes to transmit a secret message S to the legitimate receiver by cooperating with the helper who wishes to transmit an unprotected message M to the legitimate receiver.

Definition 1. *Let $\mathbb{B}_0^n(r)$ be the ball of radius r centered at the origin in $\mathbb{R}^n$ under the Euclidian norm. A (k_1, k_2, n, P_1, P_2) Gaussian wiretap channel code with a helper consists of*

- *an encoder for the transmitter*

$$e_{1,n} : \{0,1\}^{k_1} \to \mathbb{B}_0^n(\sqrt{nP_1}),$$

which, for a message $S \in \{0,1\}^{k_1}$, forms the codeword $X_1^n \triangleq e_{1,n}(S)$;
- *an encoder for the helper*

$$e_{2,n} : \{0,1\}^{k_2} \to \mathbb{B}_0^n(\sqrt{nP_2}),$$

which, for a message $M \in \{0,1\}^{k_2}$, forms the codeword $X_2^n \triangleq e_{2,n}(M)$;
- *a decoder for the legitimate receiver*

$$d_n : \mathbb{R}^n \to \{0,1\}^{k_1} \times \{0,1\}^{k_2},$$

which, from the channel observations Y^n, forms an estimate of the messages (S, M) as $(\widehat{S}, \widehat{M}) \triangleq d_n(Y^n)$;

The codomain of the encoders reflects the following power constraints for the transmitter and helper

$$\sum_{t=1}^{n}(X_i(t))^2 \leq nP_i,\ i \in \{1,2\}, \tag{3}$$

where $X_i(t)$ is the t-th entry of X_i^n. Throughout the paper, the unit of power is Watts.

The performance of a (k_1, k_2, n, P_1, P_2) code is measured in terms of

1) The average probability of error for the secret message S

$$\mathbf{P}_e^{(S)} \triangleq \frac{1}{2^{k_1}} \sum_{s=1}^{2^{k_1}} \mathbb{P}[\widehat{S} \neq s | s \text{ is sent}]; \tag{4}$$

2) The average probability of error for the unprotected message M

$$\mathbf{P}_e^{(M)} \triangleq \frac{1}{2^{k_2}} \sum_{m=1}^{2^{k_2}} \mathbb{P}[\widehat{M} \neq m | m \text{ is sent}]; \tag{5}$$

3) The information leakage about the message S at the eavesdropper

$$\mathbf{L}_e \triangleq I(S; Z^n). \tag{6}$$

Definition 2. *A (k_1, k_2, n, P_1, P_2) code is said (ϵ_S, ϵ_M)-reliable if $\mathbf{P}_e^{(S)} \leq \epsilon_S$ and $\mathbf{P}_e^{(M)} \leq \epsilon_M$, and δ-secure if $\mathbf{L_e} \leq \delta$. Moreover, a rate pair $(\frac{k_1}{n}, \frac{k_2}{n})$ is $(\epsilon_S, \epsilon_M, \delta)$-achievable with power constraint (P_1, P_2) if there exists an (ϵ_S, ϵ_M)-reliable and δ-secure (k_1, k_2, n, P_1, P_2) code.*

Note that the encoders $(e_{1,n}, e_{2,n})$ and the decoder d_n are public knowledge and known to the eavesdropper.

III. Coding Scheme

We first describe, at a high level, our coding scheme in Section III-A. Our coding approach consists of two coding layers: a reliability layer, whose design is described in Section III-B, and a security layer, whose design is described in Section III-C. Finally, we provide simulation results and examples of our code designs in Section IV. Our simulation

results show a significant advantage in terms of information leakage having a helper compared to having no helper.

A. High-level description of our coding scheme

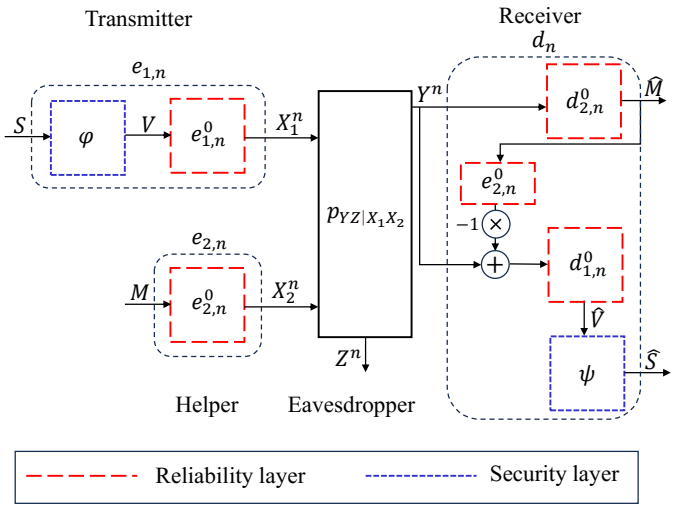


Figure 2: Our code design consists of a reliability layer and a security layer. The reliability layer is implemented using two encoders $(e^0_{1,n}, e^0_{2,n})$ and two decoders $(d^0_{1,n}, d^0_{2,n})$, and the security layer is implemented using the functions φ and ψ.

As shown in Figure 2, our code construction consists of (i) a reliability layer implemented with an ϵ-reliable (n, q_1, q_2, P_1, P_2) code described by the encoders $e^0_{1,n}$ for the transmitter and $e^0_{2,n}$ for the helper, and a decoder pair $(d^0_{1,n}, d^0_{2,n})$ for the legitimate receiver,[1] and (ii) a security layer for the transmitter, which consists of an encoding function φ and a decoding function ψ. Note that for the helper there is no secrecy layer, as the security constraint (6) concerns only the transmitter. As detailed in Sections III-B and III-C, we will design the reliability layer using a deep learning approach based on a neural network autoencoder with SIC, and the secrecy layer using universal hash functions.

Encoding at the transmitter: The transmitter first generates a sequence B of $q_1 - k_1$ bits uniformly at random in $\{0,1\}^{q_1-k_1}$, which represents local randomness used to randomize the output of the function φ to confuse the eavesdropper. Then, the transmitter encodes the message S uniformly distributed in $\{0,1\}^{k_1}$ as $e^0_{1,n}(\varphi(S,B))$. The overall encoding map $e_{1,n}$ for the transmitter that describes both secrecy and reliability layer encoding is described by

$$e_{1,n} : \{0,1\}^{k_1} \times \{0,1\}^{q_1-k_1} \to \mathbb{B}^n_0(\sqrt{nP_1}),$$
$$(s,b) \mapsto e^0_{1,n}(\varphi(s,b)).$$

Encoding at the helper: The helper encodes the message M uniformly distributed in $\{0,1\}^{k_2}$ as $e^0_{2,n}(M)$.

Decoding: From the channel observations Y^n, the legitimate receiver successively decodes M and S. Specifically, the message M is first decoded as $\widehat{M} \triangleq d^0_{2,n}(Y^n)$, then the message S is decoded as $\psi(d^0_{1,n}(Y^n - \sqrt{h_2}\widehat{X}^n_2))$, where $\widehat{X}^n_2 \triangleq e^0_{2,n}(\widehat{M})$.

[1]This code is designed without any security requirement, i.e., its performance is solely measured in terms of the average probability of errors (4), (5).

B. Design of the reliability layer

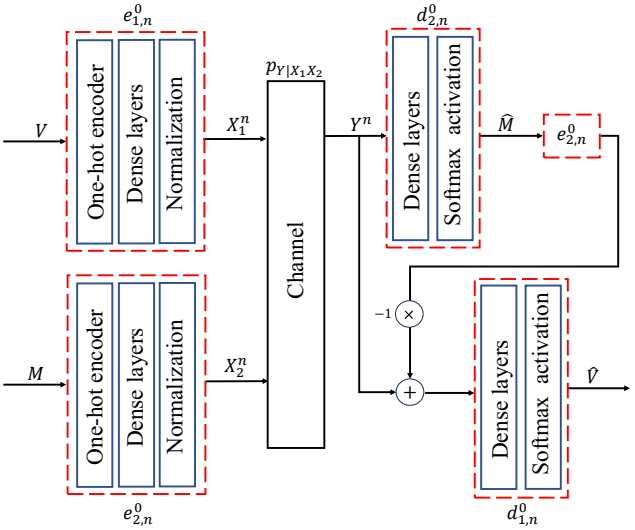


Figure 3: Architecture of the autoencoder based on successive interference cancellation.

The design of the reliability layer consists in designing an (ϵ_S, ϵ_M)-reliable (n, q_1, q_2, P_1, P_2) code described by two encoders $(e^0_{1,n}, e^0_{2,n})$ and two decoders $(d^0_{1,n}, d^0_{2,n})$ for the channel described by (1). The input message $V \in \mathcal{V} \triangleq \{1, 2, \ldots, 2^{q_1}\}$ is encoded using a neural network encoder with an encoding function $e^0_{1,n}$ to obtain the codeword X^n_1, and the input message $M \in \mathcal{M} \triangleq \{1, 2, \ldots, 2^{q_2}\}$ is encoded using a neural network encoder with an encoding function $e^0_{2,n}$ to obtain the codeword X^n_2. As depicted in Figure 3, the encoders $e^0_{1,n}$ and $e^0_{2,n}$ consist of (i) an input layer where the message is fed to a one-hot encoder, which is followed by (ii) dense layers with the ReLU activation function, followed by (iii) a dense layer that returns a vector of dimension n, followed by (iv) a normalization layer that ensures that the average power constraints (3) are met for the codewords. The decoder receives the channel output Y^n and applies the decoder pair $(d^0_{1,n}, d^0_{2,n})$ to successively estimate the messages M and V, as shown in Figure 3, which is inspired by the well-known SIC method, e.g., [7]. Specifically, upon receiving Y^n, the decoder $d^0_{2,n}$ recovers M as $\widehat{M}$, while treating the signal $\sqrt{h_1}X^n_1$ from the transmitter as noise. Then, the receiver subtracts $\sqrt{h_2}X^n_2$ from Y^n and the decoder $d^0_{1,n}$ decodes V as $\widehat{V}$. Note that we assume that $\mathbb{P}[\widehat{M} \neq M] \leq \mathbb{P}[\widehat{V} \neq V]$, otherwise the decoder would first decode V, then M, and Figures 2 and 3 would need to be modified accordingly.

As depicted in Figure 3, the neural network decoders $(d^0_{1,n}, d^0_{2,n})$ consist of dense layers with ReLU activation and a final layer with the softmax activation function whose output is a probability vector over all possible messages. Finally, the

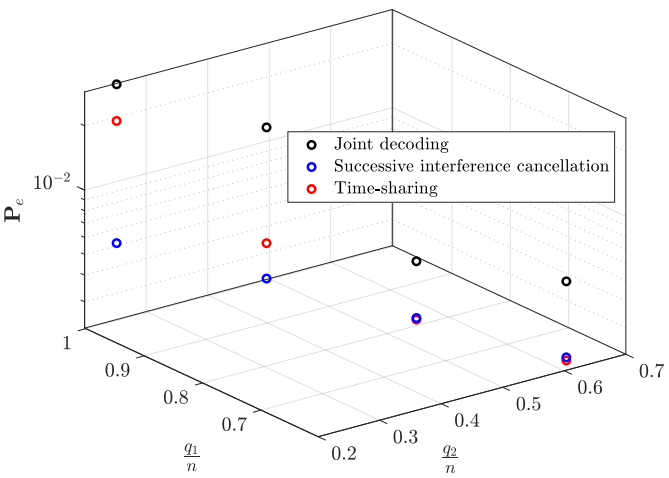


Figure 4: Comparison of schemes based on joint decoding, successive interference cancellation, and time-sharing when $h_1 = 1$, $h_2 = 1$, $n = 8$, $P_1 = 2$, $P_2 = 2$, and $\sigma_Y^2 = 6$.

decoded messages correspond to the index associated with the highest probability. The autoencoder is trained over all possible messages $v \in \mathcal{V}, m \in \mathcal{M}$ using an ADAM optimizer and the categorical cross-entropy loss function.

Comparison between the successive interference cancellation approach and other coding approaches: Figure 4 compares the probability of error $\mathbf{P}_e \triangleq \mathbb{P}\left[(\widehat{V}, \widehat{M}) \neq (V, M)\right]$ of our proposed code design with (i) time sharing, and (ii) joint decoding. We observe that, depending on the rate pair $(\frac{q_1}{n}, \frac{q_2}{n})$, our proposed code design has similar or better performance, in terms of probability of error, than time-sharing. We also observe that our approach outperforms joint decoding-based code designs. For the time-sharing approach, we divided the time frame into two subframes, one of length n_1 and the other of length n_2, which are optimized to minimize the probability of error. During the first subframe, transmitter encodes V as $X_1^{n_1}$, and the receiver observes Y^{n_1} and decodes V as $\widehat{V}$. During the second subframe, the helper encodes M as $X_2^{n_2}$, and the receiver observes Y^{n_2} and decodes M as $\widehat{M}$. Note that for a fair comparison with our proposed code design, we chose the power constraints $P_1^{TS} = \frac{P_1}{\alpha}$ and $P_2^{TS} = \frac{P_2}{1-\alpha}$, for the transmitter and the helper, respectively, with $\alpha \triangleq \frac{n_1}{n}$ and $n_1 + n_2 = n$. For the code design based on joint decoding, the neural network decoder consists of one decoder d_n^0 instead of two decoders $(d_{1,n}^0, d_{2,n}^0)$, i.e., the receiver simultaneously estimates V and M.

C. Design of the security layer

The objective of the secrecy layer (φ, ψ) is to limit the total amount of leaked information about the message S in the sense that $I(S; Z^n) \leq \delta$, for some $\delta > 0$. To this end, we will use 2-universal hash functions, whose definition is reviewed next.

Definition 3. *[15] Given two finite sets $\mathcal{X}$ and $\mathcal{Y}$, a family $\mathcal{G}$ of functions from $\mathcal{X}$ to $\mathcal{Y}$ is 2-universal if $\forall x_1, x_2 \in \mathcal{X},\ x_1 \neq x_2 \implies \mathbb{P}[G(x_1) = G(x_2)] \leq |\mathcal{Y}|^{-1}$, where G is the random variable that represents the choice of a function $g \in \mathcal{G}$ uniformly at random in $\mathcal{G}$.*

Let $\mathcal{L} \triangleq \{0,1\}^{q_1} \backslash \{\mathbf{0}\}$. For $k_1 \leq q_1$, consider the 2-universal family of hash functions $\mathcal{F} \triangleq (\psi_\lambda)_{\lambda \in \mathcal{L}}$, where for $\lambda \in \mathcal{L}$,

$$\psi_\lambda : \{0,1\}^{q_1} \to \{0,1\}^{k_1}, \tag{7}$$

$$v \mapsto (\lambda \odot v)_{k_1}, \tag{8}$$

where $\odot$ is the multiplication in GF(2^{q_1}) and $(\cdot)_{k_1}$ selects the k_1 left-most bits. Then, we define

$$\varphi_\lambda : \{0,1\}^{k_1} \times \{0,1\}^{q_1 - k_1} \to \{0,1\}^{q_1},$$
$$(s, b) \mapsto \lambda^{-1} \odot (s \| b), \tag{9}$$

where $(\cdot\|\cdot)$ denotes the concatenation of two strings. Note that for any $\lambda \in \mathcal{L}$, $s \in \{0,1\}^{k_1}$, $b \in \{0,1\}^{q_1 - k_1}$, we have $\psi_\lambda \circ \varphi_\lambda(s, b) = s$.

In our proposed code construction, the design of the security layer consists in carefully choosing the seed $\lambda \in \mathcal{L}$. Additionally, the performance of the security layer will be evaluated using a mutual information neural estimator (MINE) [16].

D. Coding scheme summary

When the secrecy layer is combined with the reliability layer, our coding scheme can be summarized as follows. The input of the encoder $e_{1,n}^0$ is obtained by computing $V \triangleq \varphi_\lambda(S, B)$, where $S \in \{0,1\}^{k_1}$ is the message, and $B \in \{0,1\}^{q_1 - k_1}$ is a sequence of $q_1 - k_1$ random bits generated uniformly at random. After computing V, the trained encoder $e_{1,n}^0$ generates the codeword $X_1^n \triangleq e_{1,n}^0(V)$. The input of the encoder $e_{2,n}^0$ is the message $M \in \{0,1\}^{q_2}$, and the output is the codeword of the heplper $X_2^n \triangleq e_{2,n}^0(M)$. Then, the codewords X_1^n and X_2^n are sent over the channel, and the intended receiver and the wiretapper observe Y^n and Z^n, respectively, as described by (1) and (2). The legitimate receiver decodes Y^n as $\widehat{M} \triangleq d_{2,n}^0(Y^n)$ and $\widehat{V} \triangleq d_{1,n}^0(Y^n - \sqrt{h_2}\widehat{X}_2^n)$, where $\widehat{X}_2^n \triangleq e_{2,n}^0(\widehat{M})$. Finally, the receiver creates an estimate $\widehat{S}$ of S as $\widehat{S} \triangleq \psi_\lambda(\widehat{V})$.

IV. Simulations

We now provide examples of code designs that follow the guidelines described in Sections III-B, III-C, and evaluate their performance in terms of the average probability of error at the receiver and information leakage at the eavesdropper. Note that no other finite-length codes have been proposed for our specific setting. Therefore, our performance evaluation serves to quantify the gains introduced by the helper in terms of probability of error and information leakage when contrasted with a scenario where no helper is present. The neural networks are implemented in Python 3.8 using Tensorflow 2.6.2. Based on the channel parameters in (1) and (2), we consider two cases.

Case 1: $\frac{h_1}{\sigma_Y^2} \leq \frac{g_1}{\sigma_Z^2}$. In this case, the eavesdropper has a channel advantage over the legitimate receiver. Intuitively, this means that the legitimate receiver experiences more channel

noise than the eavesdropper does, and it is well known that the secrecy capacity is zero in this case [2]. Therefore, point-to-point codes cannot allow secure transmission of the message S for the transmitter, and additional resources are needed to achieve security. Here, a helper represents such a resource that can help the transmitter, provided that the helper and the legitimate receiver have a channel advantage over the eavesdropper in the sense that $\frac{h_2}{\sigma_Y^2} > \frac{g_2}{\sigma_Z^2}$. In Figures 5a and 5b, we evaluate the performance of our code design, and demonstrate this benefit of cooperation among the transmitter and the helper at finite blocklength.

Case 2: $\frac{h_1}{\sigma_Y^2} > \frac{g_1}{\sigma_Z^2}$. In this case, the legitimate receiver has a channel advantage over the eavesdropper. In Figures 5a and 5c, we evaluate the performance of our code design and, similar to Case 1, demonstrate that the helper can help the transmitter decreasing the information leakage at the eavesdropper, provided that the helper and the legitimate receiver have a channel advantage over the eavesdropper in the sense that $\frac{h_2}{\sigma_Y^2} > \frac{g_2}{\sigma_Z^2}$.

Note that if $\frac{h_2}{\sigma_Y^2} \leq \frac{g_2}{\sigma_Z^2}$, then the helper could also help to improve the information leakage for the transmitter but would also negatively affect the probability of error of the secret message.

A. Average probability of error

a) With helper: Training: We consider the channel model (1) with $\sigma_Y^2 = 1$. For the design of the reliability layer (Section III-B), the autoencoder is trained for $(n, q_1, q_2) = (12, 4, 4)$, $(n, q_1, q_2) = (16, 6, 6)$, $(n, q_1, q_2) = (20, 8, 8)$, and $(n, q_1, q_2) = (24, 12, 12)$ at an initial learning rate of 0.005 over 600 epochs of $2 \cdot 10^5$ random encoder input messages with a batch size of 100 and 200, respectively.

Testing: To evaluate the average probability of error for the unprotected message $\mathbf{P}_e^{(M)}$, we first generate the inputs $V \in \{0,1\}^{q_1}$ and $M \in \{0,1\}^{q_2}$. Then, V is passed through the trained encoder $e_{1,n}^0$ and M is passed through the trained encoder $e_{2,n}^0$, which generates the codewords X_1^n and X_2^n, respectively, and the channel output Y^n. Finally, the trained decoder $d_{2,n}^0$ forms an estimate of M from Y^n, as described in Section III-B.

Consider φ and ψ with $q_1 \in \{4, 6, 8, 10\}$ and $n \in \{12, 16, 20, 24\}$. We chose the seeds as $\lambda \in \{0011, 000011, 00000011, 0000000011\}$ for the different values of q_1 and set the secret length $k_1 = 1$. To evaluate the average probability of error for the secret message $\mathbf{P}_e^{(S)}$, the trained encoder $e_{1,n}^0$ encodes the message $S \in \{0,1\}^{k_1}$ as $e_{1,n}^0(\varphi(S, B))$, as described in Section III-C, where $B \in \{0,1\}^{q_1 - k_1}$ is a sequence of $q_1 - k_1$ bits generated uniformly at random. The trained decoder $d_{1,n}^0$ forms $\widehat{S} \triangleq \psi(d_{1,n}^0(Y^n - \sqrt{h_2}\widehat{X}_2^n))$, as described in Section III-C.

b) Without helper: This scenario corresponds to $P_2 = 0$ and $M = \emptyset$ in the setting with a helper and can be implemented with point-to-point codes. Hence, to evaluate $\mathbf{P}_e^{(S)}$, we used the best known point-to-point code construction for the Gaussian wiretap channel, i.e., the code design from [12].

Figure 5a shows the average probability of error versus blocklength n. We observe a similar probability of error $\mathbf{P}_e^{(S)}$ for the secret message for both code designs with or without a helper, which shows that the presence of the helper does not degrade performance in terms of probability of error. Figure 5a also shows the probability of error $\mathbf{P}_e^{(M)}$ for the unprotected message, which is only present in code designs with a helper.

Note that the probability of error increases as blocklength n increases because the rates $\frac{q_1}{n}$ and $\frac{q_2}{n}$ are not fixed. Note that due to different power constraints (i.e., $P_2 > P_1$), the probability of error for the unprotected message is smaller than the probability of error for the secret message.

B. Information leakage at the eavesdropper

a) With helper: We consider the model in (2) with $\sigma_Z^2 = 1$. Set the secret length $k_1 = 1$, and set the unprotected message length $q_2 \in \{4, 6, 8, 10\}$. Consider φ and ψ with $q_1 \in \{4, 6, 8, 10\}$ and $n \in \{12, 16, 20, 24\}$. We chose the seeds as $\lambda \in \{0011, 000011, 00000011, 0000000011\}$ for the different values of q_1. Generate uniformly at random S and B that are fed to the encoder $e_{1,n}^0$ and outputs X_1^n. Similarly, generate uniformly at random M that is fed to the encoder $e_{2,n}^0$ and outputs X_2^n. The output of encoders produces the channel outputs Z^n at the eavesdropper. To evaluate the leakage $I(S; Z^n)$, we use the Mutual Information Neural Estimator (MINE) from [16], which is known to be consistent. In particular, we use a fully connected feed-forward neural network with 4 hidden layers, each having 400 neurons, and used rectified linear unit (ReLU) as an activation function. The input layer has $k_1 + n$ neurons, and the ADAM optimizer with a learning rate of 0.0001 is used for the training. We train the neural network over $100,000$ epochs of $20,000$ messages with a batch size of 2500. The samples of joint and marginal distributions are produced as described in Section IV-B.

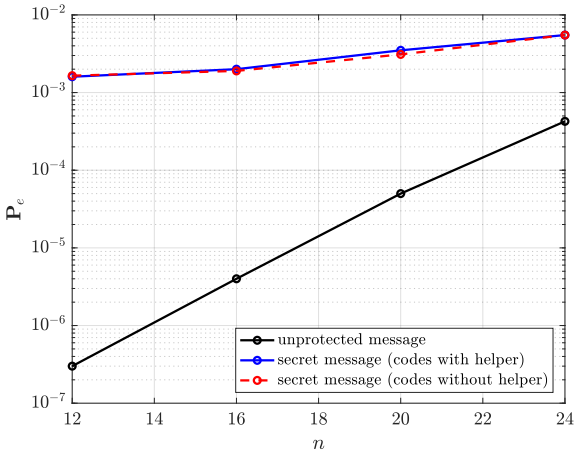
b) Without helper: To evaluate the leakage $I(S; Z^n)$ for point-to-point codes, we used the code design from [12], and it corresponds to $M = \emptyset$ and $P_2 = 0$ in the setting with a helper, that we consider in this paper.

Figures 5b and 5c show the estimated information leakage versus blocklength n for Cases 1 and 2, respectively.

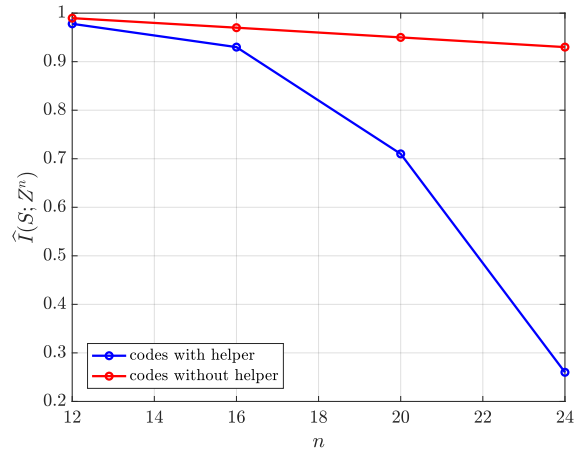
C. Discussion

As seen in Figures 5b and 5c, there is a significant improvement in terms of information leakage for codes with a helper compared to codes without a helper. Moreover, Figure 5a shows that the probability of error of the secret message remains unchanged with or without the helper.

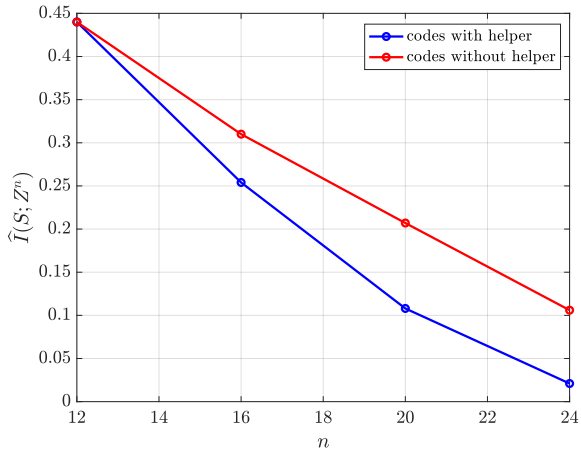
From Figures 5a and 5b, we have designed codes (corresponding to Case 1) that show that the rate pair $(\frac{1}{24}, \frac{10}{24})$ is $(\epsilon_S = 5.5 \cdot 10^{-3}, \epsilon_M = 4.2 \cdot 10^{-4}, \delta = 2.6 \cdot 10^{-1})$- achievable with power constraint $(2, 12)$. Also, from Figures 5a and 5c, we designed codes (corresponding to Case 2) that show that the rate pair $(\frac{1}{24}, \frac{10}{24})$ is $(\epsilon_S = 5.5 \cdot 10^{-3}, \epsilon_M = 4.2 \cdot 10^{-4}, \delta = 2.1 \cdot 10^{-2})$- achievable with power constraint $(2, 12)$. Since our proposed approach is modular, we only need to redesign the secrecy layer for Case 1 and Case 2 since, in both cases, the



(a)



(b)



(c)

Figure 5: (a) The average probability of error versus blocklength. (b) Case 1: The leakage versus blocklength obtained for $g_1 = 1$ and $g_2 = 0.3$. (c) Case 2: The leakage versus blocklength obtained for $g_1 = 0.2$ and $g_2 = 0.3$. When $(\frac{q_1}{n}, \frac{q_2}{n}) \in \{(\frac{4}{12}, \frac{4}{12}), (\frac{6}{16}, \frac{6}{16}), (\frac{8}{20}, \frac{8}{20}), (\frac{10}{24}, \frac{10}{24})\}$, and $k_1 = 1$. $\sigma_Y^2 = 1$, $\sigma_Z^2 = 1$, $P_1 = 2$ and $P_2 = 12$. The channel gains of the transmitter are $h_1 = 1$ and $h_2 = 1$.

channel gains h_1 and h_2 of the legitimate receiver's channel are unchanged.

Our codes' performance in Case 1 and Case 2 at short blocklengths are consistent with the previous work that considered non-constructive coding schemes and the asymptotic regime, e.g., [4], where a helper can improve the secrecy rate of a transmitter when either $\frac{h_1}{\sigma_Y^2} > \frac{g_1}{\sigma_Z^2}$ or $\frac{h_1}{\sigma_Y^2} \leq \frac{g_1}{\sigma_Z^2}$.

V. Concluding Remarks

We designed explicit and short blocklength codes for the Gaussian wiretap channel in the presence of a helper that cooperates with the transmitter. Our proposed codes showed significant improvement in information leakage compared to existing point-to-point codes, even when the transmitter has adverse channel conditions, i.e., the eavesdropper experiences less channel noise than the legitimate receiver. We proposed a framework that separates the code design into two layers: a reliability layer and a secrecy layer. We implemented the reliability layer with an autoencoder based on SIC and the secrecy layer with universal hash functions.

References

[1] A. Wyner, "The wire-tap channel," *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[2] S. Cheong and M. Hellman, "The Gaussian wire-tap channel," *IEEE Trans. Inf. Theory*, vol. 24, no. 4, pp. 451–456, 1978.

[3] E. Tekin and A. Yener, "The Gaussian multiple access wire-tap channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5747–5755, 2008.

[4] ——, "The general Gaussian multiple-access and two-way wiretap channels: Achievable rates and cooperative jamming," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2735–2751, 2008.

[5] J. Xie and S. Ulukus, "Secure degrees of freedom of the Gaussian wiretap channel with helpers," in *50th Annual Allerton Conf. Commun., Control, and Computing*, 2012, pp. 193–200.

[6] X. He and A. Yener, "Providing secrecy with structured codes: Two-user Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 4, pp. 2121–2138, 2014.

[7] T. Cover, "Broadcast channels," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 2–14, 1972.

[8] C. Wong, T. Wong, and J. Shea, "LDPC code design for the BPSK-constrained Gaussian wiretap channel," in *IEEE Globecom Workshops*, 2011, pp. 898–902.

[9] M. Baldi, G. Ricciutelli, N. Maturo, and F. Chiaraluce, "Performance assessment and design of finite length LDPC codes for the Gaussian wiretap channel," in *IEEE Int. Conf. Commun. Workshop*, 2015, pp. 435–440.

[10] A. Nooraiepour, S. Aghdam, and T. Duman, "On secure communications over Gaussian wiretap channels via finite-length codes," *IEEE Commun. Letters*, vol. 24, no. 9, pp. 1904–1908, 2020.

[11] K. Besser, P. Lin, C. Janda, and E. Jorswieck, "Wiretap code design by neural network autoencoders," *IEEE Trans. Inf. Forensics and Security*, vol. 15, pp. 3374–3386, 2020.

[12] V. Rana and R. A. Chou, "Short blocklength wiretap channel codes via deep learning: Design and performance evaluation," *IEEE Trans. Commun.*, vol. 71, no. 3, pp. 1462–1474, 2023.

[13] V. Rana and R. Chou, "Design of short blocklength wiretap channel codes: Deep learning and cryptography working hand in hand," in *IEEE Inf. Theory Workshop*, 2021, pp. 1–6.

[14] R. Sultana, V. Rana, and R. A. Chou, "Secret sharing over a Gaussian broadcast channel: Optimal coding scheme design and deep learning approach at short blocklength," in *IEEE Int. Symp. Inf. Theory*, 2023, pp. 1961–1966.

[15] J. Carter and M. Wegman, "Universal classes of hash functions," *J. Computer and System Sciences*, vol. 18, no. 2, pp. 143–154, 1979.

[16] M. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. Hjelm, "MINE: Mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.