Short Blocklength Wiretap Channel Codes via Deep Learning: Design and Performance Evaluation

Vidhi Rana, *Member, IEEE*, and Rémi A. Chou, *Member, IEEE*

Abstract—We design short blocklength codes for the Gaussian wiretap channel under information-theoretic security guarantees. Our approach consists in decoupling the reliability and secrecy constraints in our code design. Specifically, we handle the reliability constraint via an autoencoder, and handle the secrecy constraint with hash functions. For blocklengths smaller than or equal to 128, we evaluate through simulations the probability of error at the legitimate receiver and the leakage at the eavesdropper for our code construction. This leakage is defined as the mutual information between the confidential message and the eavesdropper's channel observations, and is empirically measured via a neural network-based mutual information estimator. Our simulation results provide examples of codes with positive secrecy rates that outperform the best known achievable secrecy rates obtained non-constructively for the Gaussian wiretap channel. Additionally, we show that our code design is suitable for the compound and arbitrarily varying Gaussian wiretap channels, for which the channel statistics are not perfectly known but only known to belong to a pre-specified uncertainty set. These models not only capture uncertainty related to channel statistics estimation, but also scenarios where the eavesdropper jams the legitimate transmission or influences its own channel statistics by changing its location.

Index Terms—Wiretap channel, information-theoretic security, autoencoder, deep learning, compound and arbitrarily varying wiretap channel.

I. INTRODUCTION

THE wiretap channel [2] is a basic model to account for eavesdroppers in wireless communication. In this model, a sender (Alice) encodes a confidential message M into a codeword X^n and transmits it to a legitimate receiver (Bob) over n uses of a channel in the presence of an external eavesdropper (Eve). Bob's estimate of M from his channel output observations is denoted by $\hat{M}$, and Eve's channel output observations are denoted by Z^n. In [2], the constraints are that Bob must be able to recover M, i.e., $\lim_{n\to\infty} \mathbb{P}[M \neq \hat{M}] = 0$, and the leakage about M at Eve, quantified by $I(M; Z^n)$, is not

too large in the sense that $\lim_{n\to\infty} \frac{1}{n} I(M; Z^n) = 0$. Note that the stronger security requirement $\lim_{n\to\infty} I(M; Z^n) = 0$ can also be considered [3], meaning that Eve's observations Z^n are almost independent of M for large n. The secrecy capacity has been characterized for degraded discrete memoryless channels in [2], for arbitrary discrete memoryless channels in [4], and for Gaussian channels in [5].

While [2], [4], [5] provide non-constructive achievability schemes for the wiretap channel, constructive coding schemes have also been proposed. Specifically, coding schemes based on low-density parity-check (LDPC) codes [6], [7], [8], polar codes [9], [10], [11], [12], and invertible extractors [13], [14] have been constructed for degraded or symmetric wiretap channel models. Moreover, the method in [13] and [14] has been extended to the Gaussian wiretap channel [15]. Coding schemes based on random lattice codes have also been proposed for the Gaussian wiretap channel [16]. Subsequently, constructive [17], [18], [19] and random [20] polar coding schemes have been proposed to achieve the secrecy capacity of non-degraded discrete wiretap channels. Coding schemes that combine polar codes and invertible extractors have also been proposed to avoid the need for a pre-shared secret under strong secrecy [21], [22]. All the references above consider the asymptotic regime, i.e., the regime where n approaches infinity. However, many practical applications require short packet lengths or low latency [23]. To fulfill this need, non-asymptotic and second-order asymptotics achievability and converse bounds on the secrecy capacity of discrete and Gaussian wiretap channels have been established in [24], [25], and [26]. Note that [24], [25], and [26] focus on deriving fundamental limits and not on code constructions. We will review the works that are most related to our study and focus on code constructions at finite blocklength for the wiretap channel in Section II.

In this paper, we propose to design short blocklength codes (smaller than or equal to 128) for the Gaussian wiretap channel under information-theoretic security guarantees. Such an information-theoretic approach enables coding solutions robust against computationally unbounded adversaries, and are thus technology independent and, in particular, quantum proof. Specifically, we quantify security in terms of the leakage $I(M; Z^n)$, i.e., the mutual information between the confidential message and the eavesdropper's channel observations. The main idea of our approach is to decouple the reliability and secrecy constraints. Specifically, we use a deep learning approach based on a feed-forward neural

network autoencoder [27] to handle the reliability constraint and cryptographic tools, namely, hash functions [28], to handle the secrecy constraint.[1] Then, to evaluate the performance of our constructed code, we empirically estimate the leakage $I(M; Z^n)$. Note that even for small values of n this estimation is challenging with standard techniques such as binning of the probability space [29], k-nearest neighbor statistics [30], or maximum likelihood estimation [31]. Unlike [24], [25], [26], which analytically derive upper bounds on the leakage, we consider a practical approach to estimate the leakage via the mutual information neural estimator (MINE) from [32], which is provably consistent and offers better performances than other known mutual information estimators in high dimension. We also compare the performances of our codes with the best-known achievability and converse bounds on optimal secrecy rates for the Gaussian wiretap channel [24].

Our main contributions are as follows.

1) We propose a framework based on neural networks that enables a flexible design of finite blocklength codes for the Gaussian wiretap channel. Additionally, as seen in our simulations, our code design provides examples of wiretap codes that outperform the best known achievable secrecy rates from [24] obtained non-constructively for the Gaussian wiretap channel.

2) We demonstrate that our proposed framework is also able to handle compound [33], [34] and arbitrarily varying [35], [36] settings, when uncertainty holds on both the legitimate users' channel and the eavesdropper's channel, as demonstrated by our simulations results in Section V. These models are particularly useful to capture uncertainty about the channel statistics of the eavesdropper channel or to model an active eavesdropper who can influence its channel statistics by changing its location.

3) We propose a coding scheme design able to precisely control the level of information leakage at the eavesdropper through the independent design of a reliability coding layer and a secrecy coding layer. By contrast, as elaborated on in Section II, deep learning approaches that seek to simultaneously design codes for reliability and secrecy do not seem to offer good control over the information leakage at the eavesdropper.

Additionally, our proposed code design offers the following features.

- A modular approach that separates the code design into a secrecy layer and a reliability layer. The secrecy layer only deals with the secrecy constraint and only depends on the statistics of the eavesdropper's channel, whereas

the reliability layer only deals with the reliability constraint and only depends on the statistics of the legitimate receiver's channel. This approach allows a simplified code design, for instance, if only one of the two layers needs to be (re)designed.

- A universal way of dealing with the secrecy constraint through the use of hash functions. This is beneficial, for instance, for compound [33], [34] and arbitrarily varying [35], [36] settings, as our results show that it becomes sufficient to design our code with respect to the best eavesdropper's channel.

- A method that can be applied to an arbitrary channel model as the conditional probability distribution that defines the channel is not needed and only input and output channel samples are needed to design the reliability and secrecy layers.

Note that it is difficult to analytically characterize optimal secrecy rates for the Gaussian wiretap channel in the finite blocklength regime. In this study, we adopt a practical approach based on deep learning to better understand this regime.

The remainder of the paper is organized as follows. Section II reviews related works. Section III introduces the Gaussian wiretap channel model. Section IV describes our proposed code design and our simulation results for the Gaussian wiretap channel model. Section V discusses the compound and arbitrarily varying Gaussian wiretap channel models and presents our simulation results. Finally, Section VI provides concluding remarks.

II. RELATED WORKS

As elaborated on in the introduction, several code constructions have already been proposed for Gaussian wiretap channel coding in the asymptotic blocklength regime. Another challenging task is code designs in the finite blocklength regime. Next, we review known finite-length code constructions based on coding theoretic tools and deep learning tools in Sections II-A and II-B, respectively.

A. Works Based on Coding Theory

In the following, we distinguish the works that consider a non-information-theoretic secrecy metric from the works that consider an information-theoretic secrecy metric.

1) Non-Information-Theoretic Secrecy Metric: A non-information-theoretic security metric called security gap, which is based on an error probability analysis at the eavesdropper, is used to evaluate the secrecy performance in [37], [38], [39], [40], and [41]. Specifically, randomized convolutional codes for Gaussian and binary symmetric wiretap channels are studied in [37], and randomized turbo codes for the Gaussian wiretap channel are investigated in [38]. Coding schemes for the Gaussian wiretap channel based on LDPC codes are proposed in [39] and [40]. Additionally, another non-information-theoretic security approach called practical secrecy is investigated in [42], where a leakage between Alice's message and an estimate of the message at Eve is estimated.

---

[1]Note that a coding strategy that separately handles the reliability and secrecy constraints with two separate coding layers is also used for the discrete wiretap channel in [13] and [14], and for the Gaussian wiretap channel in [15]. In these works, an asymptotic regime is considered, i.e., the blocklength n tends to infinity. Further, in [13], [14], and [15], the security layer relies on the random choice of a hash function in a family of universal hash functions, and therefore, the coding scheme is non-constructive. In this paper, we also consider a family of hash functions for the security layer but only select a specific function in this family. This choice is deterministic and part of the coding scheme design, thus making it constructive, as elaborated on in our simulation results.

2) Information-Theoretic Secrecy Metric: Next, we review works that consider the leakage $I(M; Z^n)$ as a secrecy metric. In [43], punctured systematic irregular LDPC codes are proposed for the binary phase-shift-keyed-constrained Gaussian wiretap channel, and a leakage as low as 11 percent of the message length has been obtained for a blocklength $n = 10^6$. In [44], LDPC codes for the Gaussian wiretap channel have also been developed, and a leakage as low as 20 percent of the message length has been obtained for a blocklength $n = 50,000$. Most recently, in [45], randomized Reed-Muller codes are developed for the Gaussian wiretap channel, and a leakage as low as 0.2 percent of the message length has been obtained for a blocklength $n = 16$.

B. Works Based on Deep Learning

Artificial neural networks have gained attention in communication system design because they approach the performance of state-of-the-art channel coding solutions. In [46] and [47], neural networks (autoencoder) are used to learn the encoder and decoder for a channel coding task without secrecy constraints. Other machine learning approaches for channel coding without secrecy constraints have also been investigated in [48] and [49] with reinforcement learning, in [50] with mutual information estimators, and in [51] with generative adversarial networks.

Recently, deep learning approaches for channel coding have been extended to wiretap channel coding. In [52] and [53], a coding scheme that imitates coset coding by clustering learned signal constellations is developed for the Gaussian wiretap channel under a non-information-theoretic secrecy metric, which relies on a cross-entropy loss function. In [54], neural networks are used to learn optimal precoding for the MIMO Gaussian wiretap channel. In [55], a coding scheme for the Gaussian wiretap channel is developed under the information-theoretic leakage $I(M; Z^n)$ with an autoencoder approach that seeks to simultaneously optimize the reliability and secrecy constraints. A leakage as low as 15 percent of the message length is obtained in [55] for a blocklength $n = 16$. It seems that precisely controlling and minimizing the leakage is challenging with such an approach. By contrast, in this paper, we propose an approach that separates the code design into a part that only deals with the reliability constraint (by means of an autoencoder) and another part that only deals with the secrecy constraint (by means of hash functions). As supported by our simulation results, one of the advantages of our approach is a better control of how small the leakage can be made.

III. GAUSSIAN WIRETAP CHANNEL MODEL

Notation: Unless specified otherwise, capital letters represent random variables, whereas lowercase letters represent realizations of associated random variables, e.g., x is a realization of the random variable X. $|\mathcal{X}|$ denotes the cardinality of the set $\mathcal{X}$. $\|\cdot\|_2$ denotes the Euclidean norm. $\mathrm{GF}(2^q)$ denotes a finite field of order 2^q, $q \in \mathbb{N}^*$.

For $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \mathbb{R}$, consider a memoryless Gaussian wiretap channel $(\mathcal{X}, P_{YZ|X}, \mathcal{Y} \times \mathcal{Z})$ defined by

$$Y \triangleq X + N_Y, \tag{1}$$

$$Z \triangleq X + N_Z, \tag{2}$$

where N_Y and N_Z are zero-mean Gaussian random variables with variances σ_Y^2 and σ_Z^2, respectively. As formalized next, the objective of the sender is to transmit a confidential message M to a legitimate receiver by encoding it into a sequence X^n, which is then sent over n uses of the channels (1), (2) and yields the channel observations Y^n and Z^n at the legitimate receiver and eavesdropper, respectively.

Definition 1: Let $\mathbb{B}_0^n(\sqrt{nP})$ be the ball of radius $\sqrt{nP}$ centered at the origin in $\mathbb{R}^n$ under the Euclidian norm. An (n, k, P) code consists of

- *a message set $\{0, 1\}^k$;*
- *an encoder $e : \{0, 1\}^k \rightarrow \mathbb{B}_0^n(\sqrt{nP})$, which, for a message $M \in \{0, 1\}^k$, forms the codeword $X^n \triangleq e(M)$;*
- *a decoder $d : \mathbb{R}^n \rightarrow \{0, 1\}^k$, which, from the channel observations Y^n, forms an estimate of the message as $d(Y^n)$.*

The codomain of the encoder e reflects the power constraint $\|e(m)\|_2^2 \leq nP, \; \forall m \in \{0, 1\}^k$.
The performance of an (n, k, P) code is measured in terms of

1) The average probability of error $\mathbf{P_e} \triangleq \frac{1}{2^k} \sum_{m=1}^{2^k} \mathbb{P}[d(Y^n) \neq m | m \text{ is sent}]$;
2) The leakage at the eavesdropper $\mathbf{L_e} \triangleq I(M; Z^n)$.

Definition 2: An (n, k, P) code is ϵ-reliable if $\mathbf{P_e} \leq \epsilon$ and δ-secure if $\mathbf{L_e} \leq \delta$. Moreover, a secrecy rate $\frac{k}{n}$ is (ϵ, δ)-achievable with power constraint P if there exists an ϵ-reliable and δ-secure (n, k, P) code.

IV. CODING SCHEME

We first describe, at a high level, our coding scheme in Section IV-A. Specifically, our coding approach consists of two coding layers, one reliability layer, whose design is described in Section IV-B, and one security layer, whose design is described in Section IV-C. We then comment on the communication rate of our proposed coding scheme when considering multiplexing of protected and unprotected messages in Section IV-D. Finally, we provide simulation results and examples of our code design for the Gaussian wiretap channel in Sections IV-E and IV-F.

A. High-Level Description of Our Coding Scheme

Our code construction consists of (i) a reliability layer with an ϵ-reliable (n, q, P) code, described by the encoder/decoder pair (e_0, d_0) (this code is designed without any security requirement, i.e., its performance is solely measured in terms of average probability of error), and (ii) a security layer implemented with hash functions. We design the encoder/decoder pair (e_0, d_0) of the reliability layer using a deep learning approach based on neural network autoencoders as described in Section IV-B. We will then design two functions φ_s and ψ_s in Section IV-C to perform the encoding and decoding, respectively, at the secrecy layer. The encoder/decoder pair (e, d) for the encoding and decoding process of the reliability and secrecy layers considered jointly is described as follows:

Encoding: Assume that a fixed sequence of bits $s \in \mathcal{S} \triangleq \{0, 1\}^q \backslash \{\mathbf{0}\}$, called seed, is known to all parties. Alice
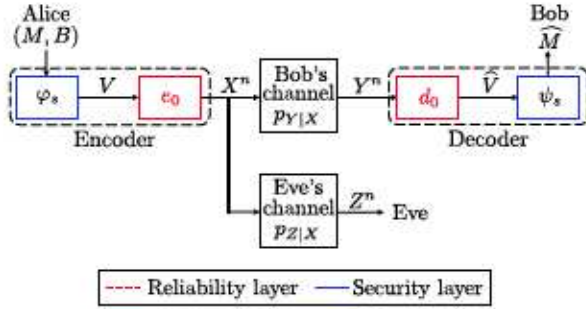
Fig. 1. Our code design consists of a reliability layer and a security layer. The reliability layer is implemented using an autoencoder (e_0, d_0) described in Section IV-B, and the security layer is implemented using the functions φ_s and ψ_s described in Section IV-C.1.

Fig. 2. Architecture of the autoencoder (e_0, d_0) via feed-forward neural networks.

generates a sequence B of $q - k$ bits uniformly at random in $\{0, 1\}^{q-k}$ (this sequence represents local randomness used to randomize the output of the function φ_s) and encodes the message $M \in \{0, 1\}^k$ as $e_0(\varphi_s(M, B))$, where $\varphi_s(M, B) \in \{0, 1\}^q$. The overall encoding map e that describes the encoding at the secrecy and reliability layers is described by

$$e : \{0, 1\}^k \times \{0, 1\}^{q-k} \to \mathbb{B}_0^n(\sqrt{nP})$$
$$(m, b) \mapsto e_0(\varphi_s(m, b)).$$

Decoding: Given Y^n and s, Bob decodes the message as $\psi_s(d_0(Y^n))$. The overall decoding map d that describes the decoding at the reliability and secrecy layers is

$$d : \mathbb{R}^n \to \{0, 1\}^k$$
$$y^n \mapsto \psi_s(d_0(y^n)).$$

For a given code design, described by the encoder/decoder pair (e, d), we will then evaluate the performance of this code by empirically measuring the leakage using a neural network-based mutual information estimator as described in Section IV-C.2. Our code design is summarized in Figure 1.

B. Design of the Reliability Layer (e_0, d_0)

The design of the reliability layer consists in designing an ϵ-reliable (n, q, P) code described by the encoder/decoder pair (e_0, d_0) for the channel (1). Define $Q \triangleq 2^q$ and let $\mathcal{V} \triangleq \{1, 2, \ldots, Q\}$ be the message set of this code. (e_0, d_0) is obtained with an autoencoder as in [46]. Specifically, the goal of the autoencoder is here to learn a representation of the encoded message that is robust to the channel noise, so that the received message at Bob can be reconstructed from its noisy channel observations with a small probability of error. In other words, the encoding part (denoted by e_0) of this autoencoder adds redundancy to the message to ensure recoverability by Bob in the presence of noise. As depicted in Figure 2, the encoder consists of (i) a one-hot encoder where the input $v \in \mathcal{V}$ is mapped to a one-hot vector $\mathbf{1}_v \in \mathbb{R}^Q$, i.e., a vector whose components are all equal to zero except the v-th component which is equal to one, followed by (ii) dense hidden layers (with rectified linear unit (ReLU) or linear activation functions [46]) that take v as input and return an n-dimensional vector, followed by
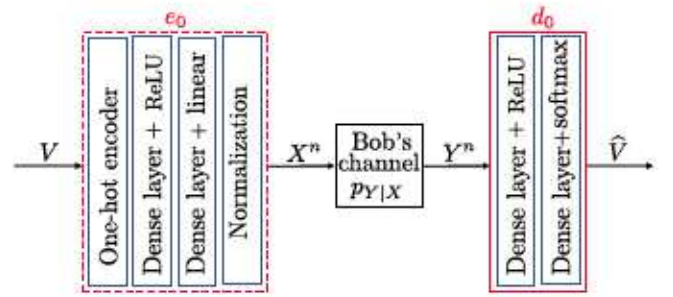
(iii) a normalization layer that ensures that the average power constraint $\frac{1}{n}\|e_0(v)\|_2^2 \leq P$ is met for the codeword $e_0(v)$. Note that, without loss of generality, one can assume that $P = 1$ since one can rewrite $\frac{1}{n}\|e_0(v)\|_2^2 \leq P$ as $\frac{1}{n}\|\tilde{e}_0(v)\|_2^2 \leq 1$, where $\tilde{e}_0(v) \triangleq e_0(v)/\sqrt{P}$. As depicted in Figure 2, the decoder consists of dense hidden layers and a softmax layer. More specifically, let $\mu^{|\mathcal{V}|}$ be the output of the last dense layer in the decoder. The softmax layer takes $\mu^{|\mathcal{V}|}$ as input and returns a vector of probabilities $p^{|\mathcal{V}|} \in [0, 1]^{|\mathcal{V}|}$, whose components p_v, $v \in \mathcal{V}$, are $p_v \triangleq \exp(\mu_v) \left(\sum_{i=1}^{|\mathcal{V}|} \exp(\mu_i)\right)^{-1}$. Finally, the decoded message $\hat{v}$ corresponds to the index of the component of $p^{|\mathcal{V}|}$ associated with the highest probability, i.e., $\hat{v} \in \arg\max_{v \in \mathcal{V}} p_v$. The autoencoder is trained over all possible messages $v \in \mathcal{V}$ using a stochastic gradient descent (SGD) [56] and the categorical cross-entropy loss function.

C. Design of the Security Layer (φ_s, ψ_s)

The objective is now to design (φ_s, ψ_s) such that the total amount of leaked information about the original message is small in the sense that $I(M; Z^n) \leq \delta$, for some $\delta > 0$. For a given choice of (φ_s, ψ_s), the performance of our code construction will be evaluated using a mutual information neural estimator (MINE) [32]. Before we describe the construction of (φ_s, ψ_s), we review the definition of 2-universal hash functions.

Definition 3: [28] *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 $\mathcal{G}$.*

1) Design of (φ_s, ψ_s): Let $\mathcal{S} \triangleq \{0, 1\}^q \backslash \{0\}$. For $k \leq q$, consider the 2-universal hash family of functions $\mathcal{G} \triangleq \{\psi_s\}_{s \in \mathcal{S}}$, where for $s \in \mathcal{S}$,

$$\psi_s : \{0, 1\}^q \to \{0, 1\}^k$$
$$v \mapsto (s \odot v)_k,$$

where $\odot$ is the multiplication in $\text{GF}(2^q)$ and $(\cdot)_k$ selects the k most significant bits. In our proposed code design, the security layer is handled via a specific function $\psi_s \in \mathcal{G}$ indexed by the seed $s \in \mathcal{S}$. Then, we define

$$\varphi_s : \{0, 1\}^k \times \{0, 1\}^{q-k} \to \{0, 1\}^q$$
$$(m, b) \mapsto s^{-1} \odot (m\|b), \qquad (3)$$

where $(\cdot\|\cdot)$ denotes the concatenation of two strings.

When the secrecy layer is combined with the reliability layer, our coding scheme can be summarized as follows. The input of the encoder e_0 is obtained by computing $V \triangleq \varphi_s(M, B)$, where $M \in \{0,1\}^k$ is the confidential message, and $B \in \{0,1\}^{q-k}$ is a sequence of $q - k$ random bits generated uniformly at random. After computing V, the encoder e_0, trained as described in Section IV-B, generates the codeword $X^n \triangleq e_0(V)$, which is sent over the channel by Alice. Bob and Eve observe Y^n and Z^n, respectively, as described by (1) and (2). The decoder d_0, trained as described in Section IV-B, decodes Y^n as $\widehat{V} \triangleq d_0(Y^n)$. Then, the receiver performs the multiplication of $\widehat{V}$ and s, which is followed by a selection of the k most significant bits to create an estimate $\widehat{M}$ of M, i.e., $\widehat{M} \triangleq \psi_s(\widehat{V})$.

2) Leakage Evaluation via Mutual Information Neural Estimator (MINE) [32]: Let $\{T_\theta : \{0,1\}^k \times \mathbb{R}^n \to \mathbb{R}\}_{\theta \in \Theta}$ be the set of functions parameterized by a deep neural network with parameters $\theta \in \Theta$. Define

$$I_\Theta(p_{MZ^n}) \triangleq \sup_{\theta \in \Theta} \mathbb{E}_{p_{MZ^n}} T_\theta(M, Z^n) - \log \mathbb{E}_{p_M p_{Z^n}} e^{T_\theta(M, Z^n)},$$

where p_{MZ^n} is the joint probability distribution of (M, Z^n). By [32], $I_\Theta(p_{MZ^n})$ can approximate the mutual information $I(M; Z^n)$ with arbitrary accuracy. Note that because the true distribution p_{MZ^n} is unknown, one cannot directly use $I_\Theta(p_{MZ^n})$ to estimate $I(M; Z^n)$. However, by estimating the expectations in $I_\Theta(p_{MZ^n})$ with samples from p_{MZ^n} and p_M and p_{Z^n}, one can rewrite $I_\Theta(p_{MZ^n})$ as

$$\widehat{I}(M; Z^n) \triangleq \sup_{\theta \in \Theta} \frac{1}{l} \sum_{i=1}^{l} T_\theta(m(i), z^n(i))$$
$$- \log \left[\frac{1}{l} \sum_{i=1}^{l} e^{T_\theta(\bar{m}(i), z^n(i))} \right],$$

where the term $\frac{1}{l} \sum_{i=1}^{l} T_\theta(m(i), z^n(i))$ represents a sample mean using l samples $(m(i), z^n(i))_{i \in \{1,...,l\}}$ from p_{MZ^n}, and the term $\frac{1}{l} \sum_{i=1}^{l} e^{T_\theta(\bar{m}(i), z^n(i))}$ represents a sample mean using l samples $(\bar{m}(i), \bar{z}^n(i))_{i \in \{1,...,l\}}$ from $p_M p_{Z^n}$.

The goal of MINE, whose architecture is depicted in Figure 3, is to design T_θ such that $\widehat{I}(M; Z^n)$ approaches the mutual information $I(M; Z^n)$. By [32], the estimator $\widehat{I}(M; Z^n)$ converges to $I(M; Z^n)$ when the number of samples is sufficiently large [32]. Guidelines to implement the estimator $\widehat{I}(M; Z^n)$ are also provided in [32].

D. Discussion on the Communication Rate When Multiplexing Protected and Unprotected Messages

Note that our approach incurs no rate loss compared to a traditional channel code. Our proposed design of an (n, k, P) code with power constraint P, blocklength n, and secret message length k consists of two layers: (i) a reliability layer implemented with a (n, q, P) channel code (e_0, d_0), and (ii) a secrecy layer. As described in Section IV-C, the secrecy layer takes as input a sequence of q bits, out of which k bits correspond to the secret message M and $q - k$ bits correspond to random bits (denoted by B in Section IV-C). By construction, the sequence B can be reconstructed at Bob
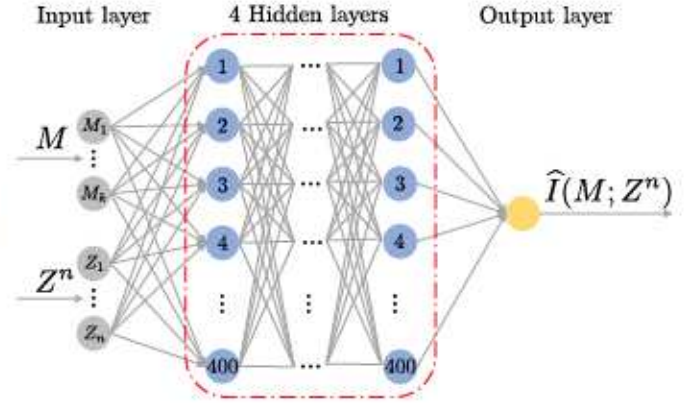

Fig. 3. The security performance is evaluated in terms of the leakage $I(M; Z^n)$ via the mutual information estimator described in Section IV-C.2, where $M \triangleq (M_i)_{i \in \{1,2,...,k\}}$, $Z^n \triangleq (Z_j)_{j \in \{1,2,...,n\}}$.

with an average probability of error $\mathbf{P}_e(e_0, d_0)$. However, the security constraint only holds on M and not on B. To summarize, our code design transforms a channel code with rate $\frac{q}{n}$ into a wiretap code able to transmit a confidential message M with rate $\frac{k}{n}$ and an unprotected message B with rate $\frac{q-k}{n}$. Hence, there is no loss compared to a channel code, as the overall transmission rate is $\frac{q}{n}$.

### E. Simulations and Examples of Code Designs for $n \leq 16$

We now provide examples of code designs that follow the guidelines described in Sections IV-B, IV-C, and evaluate their performance in terms of average probability of error at Bob and leakage at Eve. The neural networks are implemented in Python 3.7 using Tensorflow 2.5.0.

1) Autoencoder Training for the Design of the Reliability Layer (e_0, d_0): We consider the channel model (1) with $\sigma_Y^2 \triangleq 10^{-\mathrm{SNR}_B/10}$ and $\mathrm{SNR}_B = 9$dB, where, as explained in Section IV-B, without loss of generality, we choose $P = 1$. The autoencoder is trained for $q = n - 1$ using SGD with the Adam optimizer [56] at a learning rate of 0.001 over 600 epochs of 10^5 random encoder input messages with a batch size of 1000. Due to the exponential growth of the complexity with q, we changed the value of q to $n - 2$ when $n = 16$. Specifically, to evaluate $\mathbf{P}_e(e_0, d_0)$, we first generate the input $V \in \{0,1\}^q$. Then, V is passed through the trained encoder e_0, which generates the codewords X^n and the channel output Y^n. Finally, the trained decoder d_0 forms an estimate of V from Y^n.

Figure 4a compares the achievable rate $\frac{q}{n}$ of our reliability layer (e_0, d_0) with the best known achievability and converse bounds [57, Section III.J] for channel coding. We observe that the rate of our reliability layer outperforms the achievability bounds from [57] for blocklengths smaller than or equal to 16 when $\mathrm{SNR}_B = 9$dB. Note that for each value of n, this comparison is made for a given average probability of error $\mathbf{P}_e(e_0, d_0)$ as specified in Figure 4.

2) Design of the Secrecy Layer and Leakage Evaluation: The seeds selected for the simulations are given in Table I. All possible seeds have been tested for the values of n smaller than or equal to eight to minimize the leakage, and only one seed is tested for the values of n greater than eight.

(a)
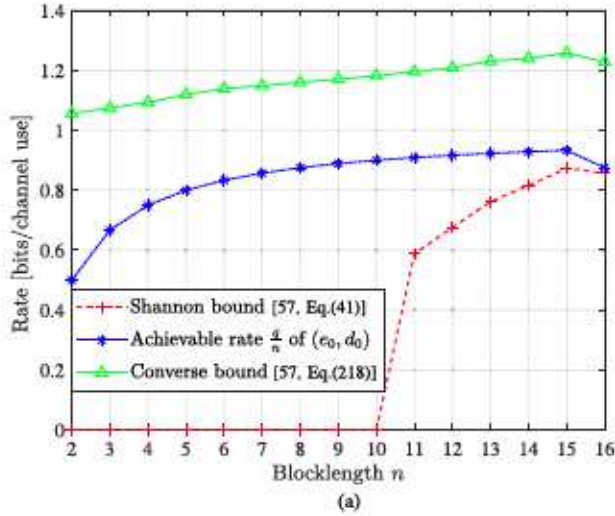
n	$\mathbf{P}_e(e_0, d_0)$
2	$3.26 \cdot 10^{-5}$
3	$1.040 \cdot 10^{-4}$
4	$2.042 \cdot 10^{-4}$
5	$3.620 \cdot 10^{-4}$
6	$5.280 \cdot 10^{-4}$
7	$6.442 \cdot 10^{-4}$
8	$7.710 \cdot 10^{-4}$
9	$8.982 \cdot 10^{-4}$
10	$1.0438 \cdot 10^{-3}$
11	$1.2410 \cdot 10^{-3}$
12	$1.4864 \cdot 10^{-3}$
13	$2.0256 \cdot 10^{-3}$
14	$2.2706 \cdot 10^{-3}$
15	$2.8636 \cdot 10^{-3}$
16	$1.7192 \cdot 10^{-3}$

(b)

Fig. 4. Figure 4a shows the rate versus the blocklength n obtained with $\epsilon \triangleq \mathbf{P}_e(e_0, d_0)$ listed in Figure 4b when $\text{SNR}_B = 9\text{dB}$.

The leakage is evaluated using MINE as follows. We used a fully connected feed-forward neural network with 4 hidden layers, each having 400 neurons, and a ReLU as activation function. The input layer has $k + n$ neurons, and the Adam optimizer with a learning rate of 0.001 is used for the training. The samples of the joint distribution p_{MZ^n} are produced via uniform generation of messages $M \in \{0,1\}^k$ that are fed to the encoder $e = e_0 \circ \varphi_s$, whose output X^n produces the channel output Z^n at Eve. The samples of the marginal distributions are generated by dropping either m or z^n from the joint samples (m, z^n). We have trained the neural network over 10000 epochs of 20,000 messages with a batch size of 2500. Figure 5 shows the leakage versus the blocklength n for different values of k and $\text{SNR}_E = -5\text{dB}$. We observe that the leakage increases as k increases for fixed n and SNR_E, which is also supported by the following inequality on the leakage. When $k = 2$, if we write $M = (M_1, M_2)$, where $M_1, M_2 \in \{0,1\}$, then by the chain rule and nonnegativity of the mutual information, we have

$$I(M; Z^n) = I(M_1; Z^n) + I(M_2; Z^n | M_1) \geq I(M_1; Z^n),$$

where $I(M_1; Z^n)$ is interpreted as the leakage of a code with secrecy rate $\frac{1}{n}$ by considering that M_2 is a random bit part of B in (3).

TABLE I

SELECTED SEEDS FOR THE SECURITY LAYER

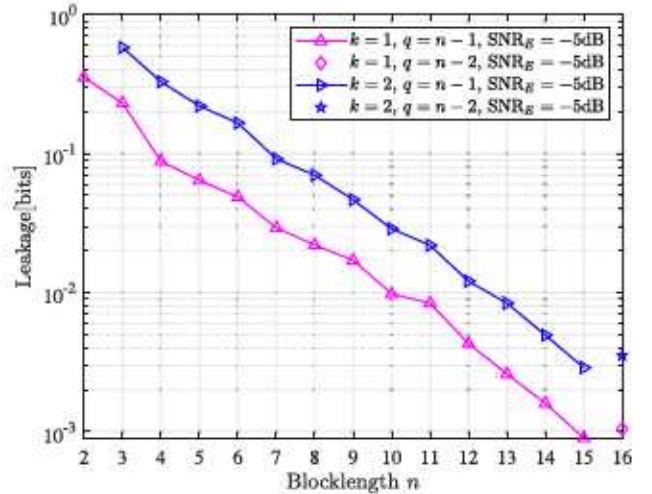| n	seed s ($k = 1$)	seed s ($k = 2$)
2	1	-
3	11	11
4	010	010
5	1100	1100
6	00010	00011
7	001001	001001
8	0001101	0001101
9	10000000	10000000
10	100000000	100000000
11	1000000000	1000000000
12	10000000000	10000000000
13	100000000000	100000000000
14	1000000000000	1000000000000
15	10000000000000	10000000000000
16	100000000000000	100000000000000

Fig. 5. Leakage $\widehat{I}(M; Z^n)$ versus blocklength. When $n \in \{2, 3, 4 \ldots, 15\}$, $q = n - 1$, and when $n = 16$, $q = n - 2$.

Remark 1: We observe a significant improvement in the leakage for a channel code coupled with our secrecy layer compared to the same channel code without any secrecy layer. For instance, for the blocklength $n = 8$ when $q = n - 1$ and $\text{SNR}_E = -5\text{dB}$, the estimated mutual information between the input message of length q to the encoder e_0 and the eavesdropper's channel observations is $\widehat{I}(V; Z^n) = 1.55$ bits. Therefore, for a one-bit input, the leakage is 0.2214 bits on average. Also, for $n = 8$, $q = n - 1$, $k = 1$, $s = 0001101$, and $\text{SNR}_E = -5\text{dB}$, the estimated mutual information between the one-bit confidential message and the eavesdropper's channel observations is $\widehat{I}(M; Z^n) = 0.022$ bits. Hence, in this example, without our secrecy layer, a leakage as low as 22 percent is obtained per information bit on average, while with our secrecy layer, a leakage as low as 2.2 percent is obtained per information bit.

3) Average Probability of Error Analysis: To evaluate $\mathbf{P}_e(e, d)$, the trained encoder e_0 encodes the message $M \in \{0,1\}^k$ as $e_0(\varphi_s(M, B))$, as described in Section IV-C, where $B \in \{0,1\}^{q-k}$ is a sequence of $q - k$ bits generated uniformly at random. The trained decoder d_0 forms $\widehat{M} \triangleq \psi_s(d_0(Y^n))$,
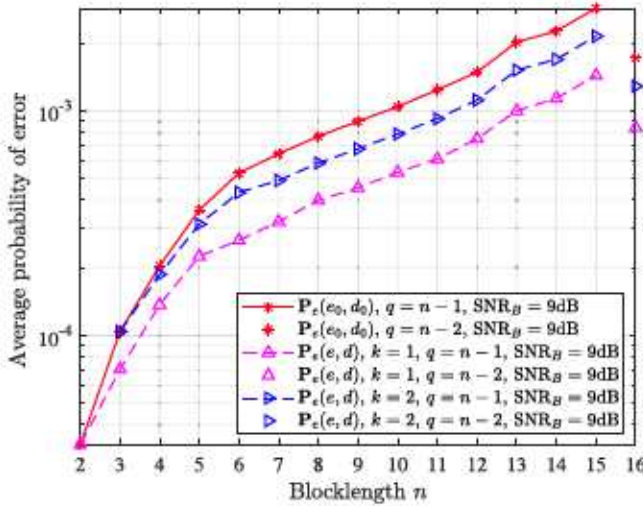
Fig. 6. Average probability of error versus blocklength. When $n \in \{2, 3, 4 \ldots, 15\}$, $q = n - 1$, and when $n = 16$, $q = n - 2$. During the training, the signal-to-noise ratio is $\mathrm{SNR}_B = 9\mathrm{dB}$.

as described in Section IV-C. Figure 6 shows $\mathbf{P}_e(e, d)$ versus the blocklength n. Note that we only plotted $\mathbf{P}_e(e, d)$ when $k = 1$ and $k = 2$ as an example, as one will always have $\mathbf{P}_e(e, d) \leq \mathbf{P}_e(e_0, d_0)$ for any value of k by construction. From Figure 6, we also observe that, for fixed n, q, and $\mathrm{SNR}_B = 9\mathrm{dB}$, the probability of error decreases as k decreases, which is also supported by the following inequality

$$\mathbb{P}[(\widehat{M}_1, \widehat{M}_2) \neq (M_1, M_2)] \geq \mathbb{P}[\widehat{M}_1 \neq M_1],$$

where we have used the union bound and $\mathbb{P}[\widehat{M}_1 \neq M_1]$ is interpreted as the probability of error of a code with secrecy rate $\frac{1}{n}$ by considering that M_2 is a random bit part of B in (3).

4) Discussion: From Figures 5 and 6, we, for instance, see that for $\mathrm{SNR}_B = 9\mathrm{dB}$ and $\mathrm{SNR}_E = -5\mathrm{dB}$, we have designed codes that show that the secrecy rate $\frac{1}{10}$ is ($\epsilon = 5.330 \cdot 10^{-4}, \delta = 9.80 \cdot 10^{-3}$)-achievable with blocklength $n = 10$, and the secrecy rate $\frac{2}{13}$ is ($\epsilon = 1.5194 \cdot 10^{-3}, \delta = 8.40 \cdot 10^{-3}$)-achievable with blocklength $n = 13$.

Figure 7a compares the achievable secrecy rate $\frac{k}{n}$ of our code design (e, d) with the best known achievability [24, Theorem 7] and converse bounds [24, Theorem 12] for the Gaussian wiretap channel, which are reviewed in the appendix for convenience. We observe that the rate of our code outperforms the best known achievability bounds for blocklengths smaller than or equal to 16 when $k = 1$, $\mathrm{SNR}_B = 9\mathrm{dB}$, and $\mathrm{SNR}_E = -5\mathrm{dB}$. Note that the best known upper bounds from [24] may not be optimal for small blocklengths, and improving them is an open problem. Note also that for each value of n, the comparison is made for a given average probability of error $\mathbf{P}_e(e, d)$ and leakage $\widehat{I}(M; Z^n)$ as specified in Figure 7.

In Figure 8, we also plotted $\epsilon \triangleq \mathbf{P}_e(e, d)$ versus $\delta \triangleq \widehat{I}(M; Z^n)$ obtained from Figure 7b.

F. Simulations and Examples of Code Designs for $n \leq 128$

We consider the channel model (1) with $\sigma_Y^2 \triangleq 10^{-\mathrm{SNR}_B/10}$ and $\mathrm{SNR}_B = 0\mathrm{dB}$. For the design of the reliability layer

(a)
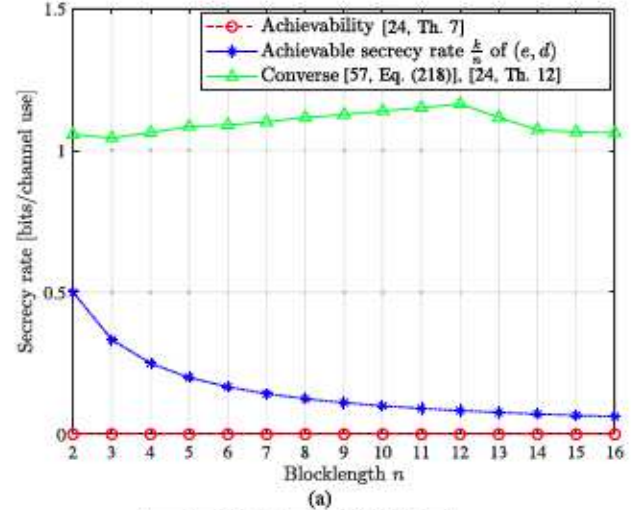
n	$\mathbf{P}_e(e, d)$ $(k = 1)$	$\widehat{I}(M; Z^n)$ $(k = 1)$
2	$3.26 \cdot 10^{-5}$	$3.5300 \cdot 10^{-1}$
3	$7.06 \cdot 10^{-5}$	$2.3100 \cdot 10^{-1}$
4	$1.370 \cdot 10^{-4}$	$8.800 \cdot 10^{-2}$
5	$2.252 \cdot 10^{-4}$	$6.440 \cdot 10^{-2}$
6	$2.654 \cdot 10^{-4}$	$4.870 \cdot 10^{-2}$
7	$3.208 \cdot 10^{-4}$	$2.930 \cdot 10^{-2}$
8	$4.000 \cdot 10^{-4}$	$2.200 \cdot 10^{-2}$
9	$4.542 \cdot 10^{-4}$	$1.710 \cdot 10^{-2}$
10	$5.330 \cdot 10^{-4}$	$9.80 \cdot 10^{-3}$
11	$6.104 \cdot 10^{-4}$	$8.40 \cdot 10^{-3}$
12	$7.510 \cdot 10^{-4}$	$4.30 \cdot 10^{-3}$
13	$1.0014 \cdot 10^{-3}$	$2.60 \cdot 10^{-3}$
14	$1.1376 \cdot 10^{-3}$	$1.16 \cdot 10^{-3}$
15	$1.4368 \cdot 10^{-3}$	$9.0 \cdot 10^{-4}$
16	$8.426 \cdot 10^{-4}$	$1.05 \cdot 10^{-3}$

(b)

Fig. 7. Figure 7a shows the secrecy rate versus the blocklength n obtained from $\epsilon \triangleq \mathbf{P}_e(e, d)$ and $\delta \triangleq \widehat{I}(M; Z^n)$ listed in Figure 7b when $\mathrm{SNR}_B = 9\mathrm{dB}$ and $\mathrm{SNR}_E = -5\mathrm{dB}$. The converse bound is obtained as the minimum between [57, Eq. (218)] and [24, Th. 12].
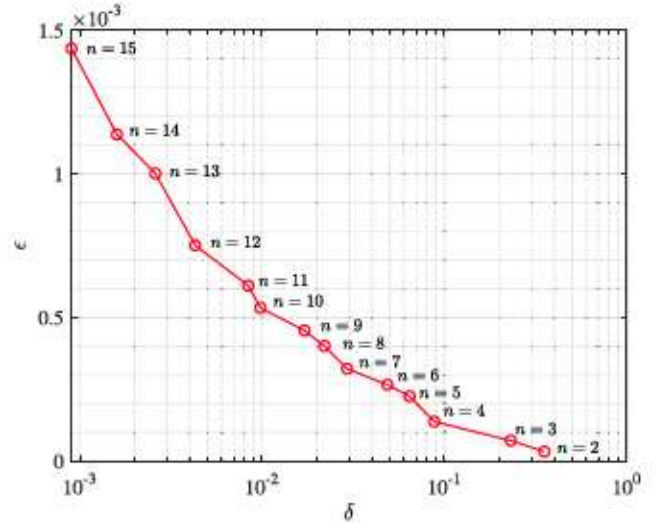

Fig. 8. $\epsilon \triangleq \mathbf{P}_e(e, d)$ versus $\delta \triangleq \widehat{I}(M; Z^n)$ obtained from Figure 7. When $n \in \{2, 3, 4 \ldots, 15\}$, $q = n - 1$, and the secrecy rate is $\frac{1}{n}$.

(e_0, d_0), the autoencoder is trained for $(n, q) = (32, 8)$, $(n, q) = (64, 8)$, $(n, q) = (96, 12)$, and $(n, q) = (128, 12)$ at a learning rate of 0.0001 over 600 epochs of 4×10^5 random
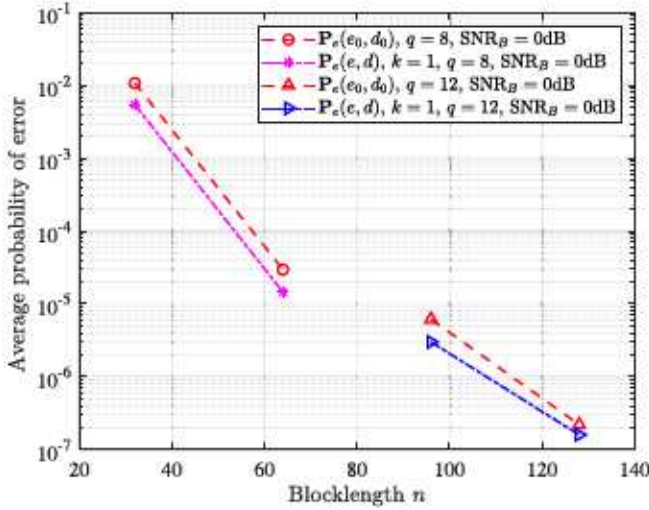
Fig. 9. Average probability of error versus blocklength. When $n \in \{32, 64\}$, $q = 8$, and when $n \in \{96, 128\}$, $q = 12$.
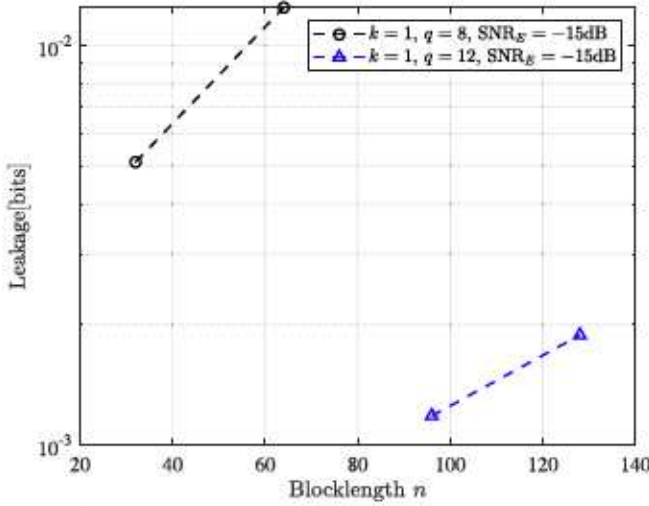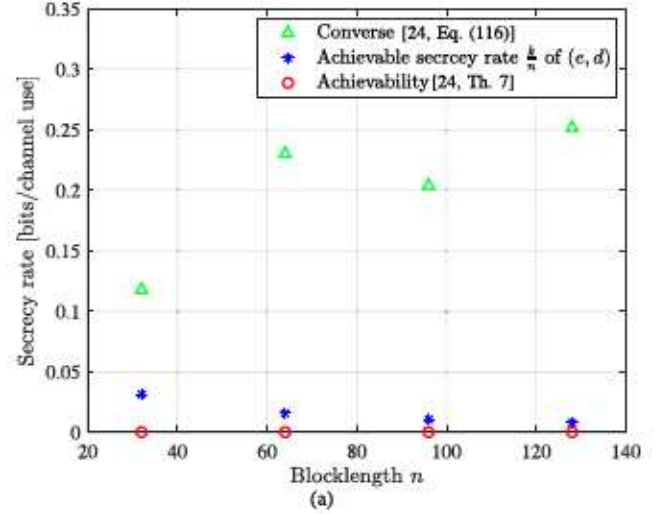

Fig. 10. Leakage $\widehat{I}(M; Z^n)$ versus blocklength. When $n \in \{32, 64\}$, $q = 8$, and when $n \in \{96, 128\}$, $q = 12$.

(a)

n	$\mathbf{P}_e(e,d)$ $(k=1)$	$\widehat{I}(M;Z^n)$ $(k=1)$
32	$5.430199 \cdot 10^{-3}$	$5.10 \cdot 10^{-3}$
64	$1.4549 \cdot 10^{-5}$	$1.244 \cdot 10^{-2}$
96	$2.999 \cdot 10^{-6}$	$1.18 \cdot 10^{-3}$
128	$1.59 \cdot 10^{-7}$	$1.88 \cdot 10^{-3}$

(b)

Fig. 11. Figure 11a shows the secrecy rate versus the blocklength n obtained from $\epsilon \triangleq \mathbf{P}_e(e,d)$ and $\delta \triangleq \widehat{I}(M;Z^n)$ listed in Figure 11b when $\text{SNR}_B = 0$dB and $\text{SNR}_E = -15$dB.

encoder input messages with a batch size of 5000. Then, 50×10^6 random messages are used to evaluate the average probability of errors $\mathbf{P}_e(e_0, d_0)$ and $\mathbf{P}_e(e, d)$ as described in Sections IV-E.1 and IV-E.3. Figure 9 shows $\mathbf{P}_e(e, d)$ versus the blocklength n. As expected, we observe that, for fixed k and q, the probability of error decreases as n increases.

The secrecy layer is implemented similarly to Section IV-C with $k = 1$, and we compute the leakage $I(M; Z^n)$ as in Section IV-E.2. We consider the model in (2) with $\sigma_Z^2 \triangleq 10^{-\text{SNR}_E/10}$ and $\text{SNR}_E = -15$dB. Additionally, in our simulations, for blocklengths $n = 32$ and $n = 64$, the seed is chosen as $s = 00000001$, and for blocklengths $n = 96$ and $n = 128$, the seed is chosen as $s = 000010000000$. Figure 10 shows $\widehat{I}(M; Z^n)$ versus the blocklength n. As expected, we observe that, for fixed k and q, the leakage increases as n increases. Finally, we observe in Figure 11a that the rate of our code outperforms the best known achievability

bounds [24, Theorem 7] for blocklengths smaller than or equal to 128 when $k = 1$, $\text{SNR}_B = 0$dB, and $\text{SNR}_E = -15$dB.

V. COMPOUND AND ARBITRARILY VARYING WIRETAP CHANNEL MODELS

We first motivate the compound and arbitrarily varying wiretap channel models in Section V-A. We then formally describe these two models in Section V-B. We present our coding scheme design in Section V-C. Finally, we evaluate the performances of our code design through simulations for the compound and arbitrarily varying Gaussian wiretap channels in Sections V-D and V-E, respectively.

A. Background

In the setting of Section III, the channel statistics are assumed to be perfectly known to Alice and Bob and fixed during the entire transmission. However, in practice, the channel statistics may not be perfectly known due to the nature of the wireless channel and inaccuracy in estimating channel statistics. Further, in some scenarios, eavesdroppers could be active and influence their own channel statistics by changing their location, or the statistics of Bob's channel through jamming. To model such scenarios, two types of models have been introduced: compound wiretap channels and arbitrarily varying wiretap channels. For compound models, e.g., [33], [34], [58], and [59], the channel statistics are fixed for all channel uses. Whereas for arbitrarily varying models, e.g., [35], [36], [58], [59], [60], and [61], the channel statistics may change in an unknown and arbitrary manner from channel

use to channel use. Constructive coding schemes have also been proposed in [21] and [22] for discrete compound and arbitrarily varying wiretap channels. While all the works above consider the asymptotic regime, in this section, we design short blocklength codes for the compound and arbitrarily varying wiretap channel models.

B. Models

For $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \mathbb{R}$, a compound or arbitrarily varying memoryless Gaussian wiretap channel $(\mathcal{X}, (p_{Y_i Z_j | X})_{i \in \mathcal{I}, j \in \mathcal{J}}, \mathcal{Y} \times \mathcal{Z})$ is defined for $i \in \mathcal{I}$, $j \in \mathcal{J}$, by

$$Y_i \triangleq X + N_{Y_i}, \quad Z_j \triangleq X + N_{Z_j},$$

where N_{Y_i} and N_{Z_j} are zero mean Gaussian random variables with variances $\sigma_{Y_i}^2$ and $\sigma_{Z_j}^2$, respectively.

For the compound wiretap channel model, the channel statistics are constant throughout the transmission and are known to belong to given uncertainty sets $\mathcal{I}$, $\mathcal{J}$. The confidential message M is encoded into a transmitted sequence X^n, and Y_i^n and Z_j^n represent the corresponding received sequence at the legitimate receiver and eavesdropper, respectively, for some $i \in \mathcal{I}$ and $j \in \mathcal{J}$.

Definition 4: A secrecy rate $\frac{k}{n}$ is (ϵ, δ)-achievable with power constraint P for the compound wiretap channel if there exists a (n, k, P) code such that

$$\max_{i \in \mathcal{I}} \mathbf{P}_e^i(e, d) \leq \epsilon, \tag{4}$$

$$\max_{j \in \mathcal{J}} I(M; Z_j^n) \leq \delta, \tag{5}$$

where $\mathbf{P}_e^i(e, d) \triangleq \frac{1}{2^k} \sum_{m=1}^{2^k} \mathbb{P}[d(Y_i^n) \neq m | m \text{ is sent}]$.

In contrast to the compound wiretap channel, the channel statistics in arbitrarily varying wiretap channel models may vary in an unknown and arbitrary manner from channel use to channel use. Specifically, for the arbitrarily varying wiretap channel model, let $Y_{\mathbf{i}}^n$ and $Z_{\mathbf{j}}^n$ represent the corresponding received sequence at the legitimate receiver and eavesdropper, respectively, for some $\mathbf{i} \in \mathcal{I}^n$ and $\mathbf{j} \in \mathcal{J}^n$.

Definition 5: A secrecy rate $\frac{k}{n}$ is (ϵ, δ)-achievable with power constraint P for the arbitrarily varying wiretap channel if there exists a (n, k, P) code such that

$$\max_{\mathbf{i} \in \mathcal{I}^n} \mathbf{P}_e^{\mathbf{i}}(e, d) \leq \epsilon,$$

$$\max_{\mathbf{j} \in \mathcal{J}^n} I(M; Z_{\mathbf{j}}^n) \leq \delta,$$

where $\mathbf{P}_e^{\mathbf{i}}(e, d) \triangleq \frac{1}{2^k} \sum_{m=1}^{2^k} \mathbb{P}[d(Y_{\mathbf{i}}^n) \neq m | m \text{ is sent}]$.

C. Coding Scheme Design

For the compound and the arbitrarily varying wiretap channels, the design of (e_0, d_0) for the reliability layer and (φ_s, ψ_s) for the secrecy layer is similar to Sections IV-B and IV-C, respectively. Specifically, we train the encoder/decoder pair for Bob's channel with noise variance $\sigma_{Y_{i^*}}^2 \triangleq \max_{i \in \mathcal{I}} \sigma_{Y_i}^2$, where $\sigma_{Y_i}^2 \triangleq 10^{-\text{SNR}_B(i)/10}$, $i \in \mathcal{I}$. In other words, the reliability layer is designed for the worse, in terms of signal-to-noise ratio, Bob's channel. Note also that, during the training
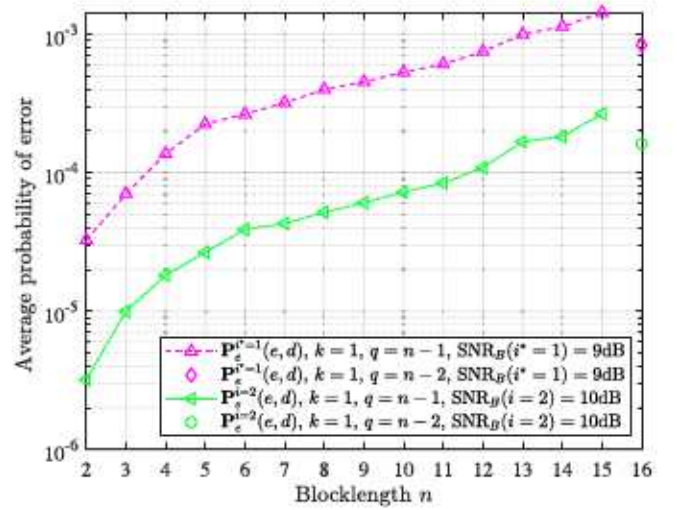

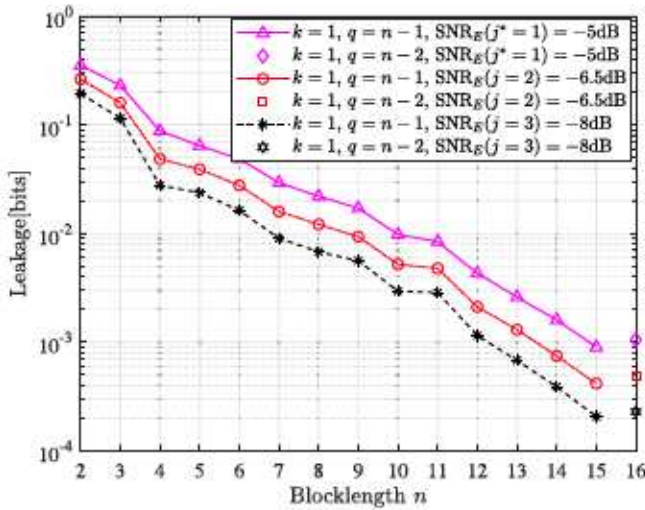Fig. 12. Average probability of error versus blocklength n.

phase, the noise variance is fixed for all the channel uses. Then, we optimize the seed s by minimizing the leakage for Eve's channel with noise variance $\sigma_{Z_{j^*}}^2 \triangleq \min_{j \in \mathcal{J}} \sigma_{Z_j}^2$, where $\sigma_{Z_j}^2 \triangleq 10^{-\text{SNR}_E(j)/10}$, $j \in \mathcal{J}$. In other words, the secrecy layer is designed for the best, in terms of signal-to-noise ratio, Eve's channel. This optimized seed s is then used by the encoder/decoder pair $(e, d) = (e_0 \circ \varphi_s, \psi_s \circ d_0)$, from which we evaluate $(\mathbf{P}_e^i(e, d), I(M; Z_j^n))$, $i \in \mathcal{I}$, $j \in \mathcal{J}$, and $(\mathbf{P}_e^{\mathbf{i}}(e, d), I(M; Z_{\mathbf{j}}^n))$, $\mathbf{i} \in \mathcal{I}^n$, $\mathbf{j} \in \mathcal{J}^n$ in Sections V-D, V-E.

D. Simulations and Examples of Code Designs for the Compound Wiretap Channel

1) Average Probability of Error Analysis: In our simulations, we consider $\mathcal{I} = \{1, 2\}$ and $\text{SNR}_B(i) \in \{9, 10\}$dB, $i \in \mathcal{I}$. We evaluate the average probability of error $\mathbf{P}_e^i(e, d)$ for $i \in \mathcal{I}$ as follows. The autoencoder is trained at $\text{SNR}_B(i^*) = 9$dB, where $i^* = 1$. The message $M \in \{0, 1\}^k$ generated uniformly at random is passed through the trained encoder e_0, which generates the codewords X^n and the channel output $Y_i^n \triangleq X^n + N_{Y_i}^n$, $i \in \mathcal{I}$, where $N_{Y_i} \sim \mathcal{N}(0, \sigma_{Y_i}^2)$. Then, the trained decoder d_0 forms an estimate $\widehat{M}_i, i \in \mathcal{I}$. Here, $\sigma_{Y_i}^2$ is fixed for the entire duration of the transmission. We use 5×10^6 random messages to evaluate the average probability of error. Figure 12 shows the average probability of error $\mathbf{P}_e^i(e, d)$ when $k = 1$ for $\text{SNR}_B(i^* = 1) = 9$dB and $\text{SNR}_B(i = 2) = 10$dB. We observe from Figure 12 that it is sufficient to design our code for the worst signal-to-noise ratio for Bob, i.e., $\text{SNR}_B(i^*) = 9$dB. In particular, we observe that, irrespective of what the actual channel is, Bob is able to decode the message with a probability of error smaller than or equal to $\mathbf{P}_e^{i^*=1}(e, d)$.

2) Leakage Evaluation: For the simulations, we consider $\mathcal{J} = \{1, 2, 3\}$ and $\text{SNR}_E(j) \in \{-8, -6.5, -5\}$dB, $j \in \mathcal{J}$. We compute the leakage $I(M; Z_j^n)$ for $j \in \mathcal{J}$ as in Section IV-E.2. The message $M \in \{0, 1\}^k$ is passed through the trained encoder e_0, which generates the codewords X^n, and the channel output at Eve is $Z_j^n \triangleq X^n + N_{Z_j}^n, j \in \mathcal{J}$, where $N_{Z_j} \sim \mathcal{N}(0, \sigma_{Z_j}^2)$. The noise variance $\sigma_{Z_j}^2$ is fixed

Fig. 13. Leakage $\widehat{I}(M; Z_j^n)$ versus blocklength n.

Fig. 14. Average probability of error versus blocklength n when $k = 1$ and training $\mathrm{SNR}_B(\mathbf{i}^*) = 9\mathrm{dB}$.

for the entire duration of the transmission. Figure 13 shows the estimated leakage $\widehat{I}(M; Z_j^n)$, at $\mathrm{SNR}_E(j = 1) = -8\mathrm{dB}$, $\mathrm{SNR}_E(j = 2) = -6.5\mathrm{dB}$, and $\mathrm{SNR}_E(j^* = 3) = -5\mathrm{dB}$. From Figure 13, we observe that it is sufficient to design our code for the best signal-to-noise ratio, i.e., $\mathrm{SNR}_E(j^*) = -5\mathrm{dB}$. In particular, we see that, irrespective of what the actual eavesdropper's channel is, we always achieve a leakage smaller than or equal to $\widehat{I}(M; Z_{j^*}^n)$, which is also supported by the following inequality on the leakage. For $j \in \mathcal{J}$ and $j^* \in \arg\min_{j \in \mathcal{J}} \sigma_{Z_j}^2$, we have
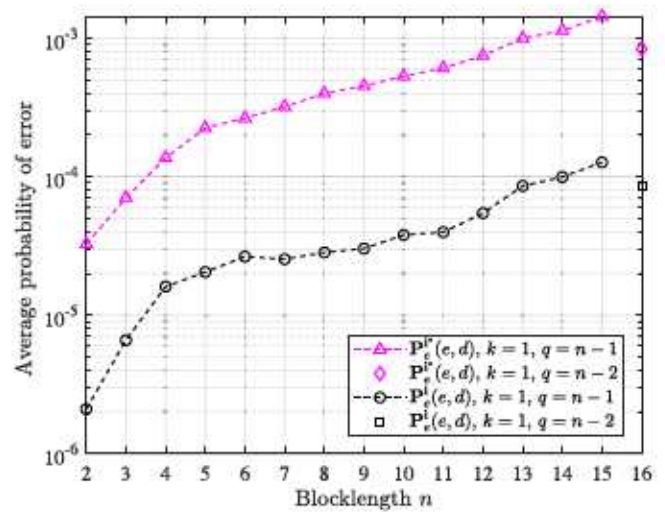
$$
\begin{aligned}
I(M; Z_j^n) &\leq I(M; Z_j^n Z_{j^*}^n) \\
&= I(M; Z_{j^*}^n) + I(M; Z_j^n | Z_{j^*}^n) \\
&= I(M; Z_{j^*}^n),
\end{aligned}
$$

where the first inequality holds by the chain rule and nonnegativity of the mutual information, the first equality holds by the chain rule, and the last equality holds because, without loss of generality, one can redefine Z_j such that $Z_j = Z_{j^*} + N$, where $N \sim \mathcal{N}(0, \sigma_{Z_j}^2 - \sigma_{Z_{j^*}}^2)$, since the distributions $p_{Z_j|X}$ and $p_{Z_{j^*}|X}$ are preserved and the constraints (4) and (5) of the problem do not depend on the joint distributions $p_{Z_j Z_{j^*}}$.

As an example, from Figures 12 and 13, we see that for $\mathrm{SNR}_B(i) \in \{9, 10\}\mathrm{dB}$, $i \in \mathcal{I}$, and $\mathrm{SNR}_E(j) \in \{-8, -6.5, -5\}\mathrm{dB}$, $j \in \mathcal{J}$, we have designed codes that show that the secrecy rate $\frac{1}{8}$ is $(\epsilon = 4.0 \cdot 10^{-4}, \delta = 2.2 \cdot 10^{-2})$-achievable with blocklength $n = 8$.

E. Simulations and Examples of Code Designs for the Arbitrarily Varying Wiretap Channel

1) Average Probability of Error Analysis: For the arbitrarily varying channel, we evaluate the probability of error $\mathbf{P}_e^{\mathbf{i}}(e, d)$, $\mathbf{i} \in \mathcal{I}^n$, for $k = 1$ in Figure 14 as follows. We consider $\mathcal{I} = \{1, 2, 3, \ldots, 31\}$ and $\mathrm{SNR}_B(i) \in \{9, 9.1, 9.2, \ldots, 12\}$, $i \in \mathcal{I}$. The autoencoder is trained at $\mathrm{SNR}_B(i^* = 1) = 9\mathrm{dB}$, where the noise variance is fixed for the entire duration of the transmission. The message $M \in \{0, 1\}^k$ generated uniformly at random is passed through the

trained encoder e_0, which generates the codewords X^n and the channel output at Bob $Y_i^n \triangleq X^n + N_{Y_i^n}$, $\mathbf{i} \in \mathcal{I}^n$. Then, the trained decoder d_0 forms an estimate $\widehat{M_i}$, $\mathbf{i} \in \mathcal{I}^n$. Here, $N_{Y_i^n}$ is a length n vector whose variance of each component is picked uniformly at random from the known uncertainty set $\{10^{-12\mathrm{dB}/10}, 10^{-11.9\mathrm{dB}/10}, 10^{-11.8\mathrm{dB}/10}, \ldots, 10^{-9\mathrm{dB}/10}\}$. For our simulations, the variance of the noise vector $N_{Y_i^n}$ is fixed for every $50,000$ codewords. The autoencoder is tested with 5×10^6 random messages for $n > 10$ and with 10^7 random messages for $n \leq 10$. Figure 14 shows that even though there is a mismatch between the training signal-to-noise ratio of the encoder/decoder pair and the actual channel, Bob is still able to decode the message with a probability of error smaller than or equal to $\mathbf{P}_e^{\mathbf{i}^*}(e, d)$, where $\mathbf{i}^*$ is a vector made of n repetitions of i^*.

2) Leakage Evaluation: For the arbitrarily varying channel, we evaluate the leakage $I(M; Z_{\mathbf{j}}^n)$, for $\mathbf{j} \in \mathcal{J}^n$, as in Section IV-E.2. The channel output at Eve is $Z_{\mathbf{j}}^n \triangleq X^n + N_{Z_{\mathbf{j}}^n}, \mathbf{j} \in \mathcal{J}^n$. In Figure 15, we consider $\mathcal{J} = \{1, 2, 3, \ldots, 301\}$ and $\mathrm{SNR}_E(j) \in \{-8, -7.99, -7.98, \ldots, -5\}\mathrm{dB}$, $j \in \mathcal{J}$. Figure 15 shows the estimated mutual information $\widehat{I}(M; Z_{\mathbf{j}}^n)$ when $k = 1$ and $n = 8$, where the variance of the noise vector $N_{Z_{\mathbf{j}}^n}$ is fixed for $20,000$ codewords per epoch. Here, $N_{Z_{\mathbf{j}}^n}$ is a length n vector whose variance of each component is picked uniformly at random from the known uncertainty set $\{10^{5\mathrm{dB}/10}, 10^{5.01\mathrm{dB}/10}, 10^{5.02\mathrm{dB}/10}, \ldots, 10^{8\mathrm{dB}/10}\}$. We can see from Figure 15 that it is sufficient to design our code for the best signal-to-noise ratio, i.e., $\mathrm{SNR}_E(j^* = 301) = -5\mathrm{dB}$. In particular, we observe that, regardless of what the actual channel is, we always achieve a leakage smaller than or equal to $\widehat{I}(M; Z_{\mathbf{j}^*}^n)$, where $\mathbf{j}^*$ is a vector made of n repetitions of j^*, which is also supported by the following inequality on the leakage.

For any $\mathbf{j} \in \mathcal{J}^n$, we have $I(M; Z_{\mathbf{j}}^n) \leq I(M; Z_{\mathbf{j}^*}^n)$, because, similar to the compound setting, without loss of generality, one can redefine $Z_{\mathbf{j}}^n$ such that $M - Z_{\mathbf{j}}^n - Z_{\mathbf{j}^*}^n$ forms a Markov chain.
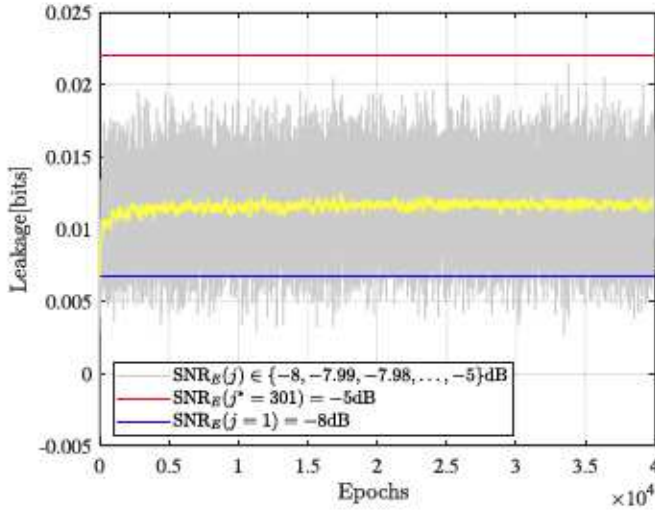
Fig. 15.   Example of leakage $\hat{I}(M;Z_j^n)$ versus epochs when $k = 1$, $n = 8$, $q = n - 1$, $s = 001101$, and $j \in \{1, 2, 3, \ldots, 301\}$. The grey curve represents the estimated leakage by a mutual information neural estimator and the yellow curve represents the 100-sample moving average of the estimated leakage. The red and blue curves represent the estimated leakage for $\text{SNR}_E(j^* = 301)$ and $\text{SNR}_E(j = 1)$, respectively, after convergence.
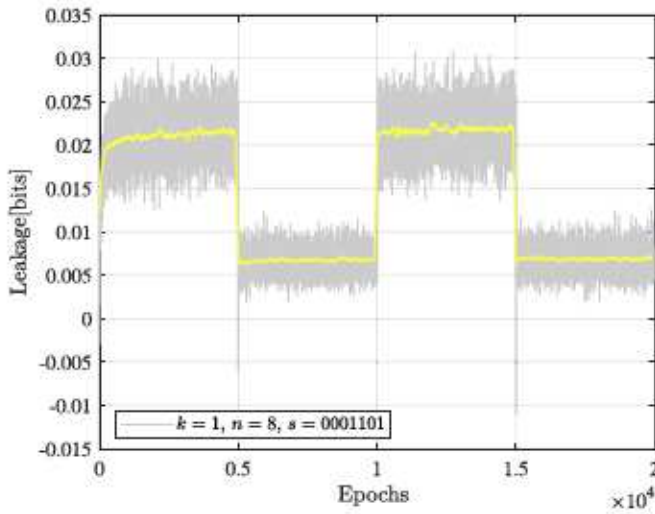

Fig. 16.   Example of leakage $\hat{I}(M;Z_j^n)$ versus epochs when $q = n - 1$ and $\text{SNR}_E(j) \in \{-8, -5\}$, $j \in \{1, 2\}$. The yellow curve represents the 100-sample moving average of the estimated leakage.

Figure 16 shows the estimated mutual informa-tion $\hat{I}(M;Z_j^n)$ for $k = 1$ and $n = 8$, when $\text{SNR}_E \in \{-8, -5\}$dB. Here, each component of the noise vector $N_{Z_j^n}$ has fixed variance, which alternatively takes the values $10^{5\text{dB}/10}$ and $10^{8\text{dB}/10}$, for every 5000 epochs with $20,000$ codewords per epoch. In other words, each component of the noise vector has variance $10^{5\text{dB}/10}$ for the first 5000 epochs, then $10^{8\text{dB}/10}$ for the next 5000 epochs, and so on. Again, we observe that it is sufficient to consider the worst case for the code design, since the leakage is always upper bounded by the leakage obtained for the best eavesdropper's signal-to-noise ratio, i.e., when $\text{SNR}_E = -5$dB.

VI. CONCLUDING REMARKS

We designed short blocklength codes for the Gaussian wire-tap channel under an information-theoretic secrecy metric. Our approach consisted in decoupling the reliability and secrecy constraints to offer a simple and modular code design, and to precisely control how small the leakage is. Specifically, we handled the reliability constraint via an autoencoder and the secrecy constraint via hash functions. We evaluated the performance of our code design through simulations in terms of probability of error at the legitimate receiver and leakage at the eavesdropper for blocklengths smaller than or equal to 128. Our results provide examples of code designs that outperform the best known achievable secrecy rates obtained non-constructively for the Gaussian wiretap channel. We high-light that our code design method can be applied to any channels since it does not require the knowledge of the channel model but only the knowledge of input and output channel samples. We also showed that our code design is applicable to settings where uncertainty holds on the channel statistics, e.g., compound wiretap channels, and arbitrarily varying wiretap channels.

APPENDIX

A. Achievability Bound for the Gaussian Wiretap Channel

The maximal secrecy rate $R(n, \epsilon, \delta)$ achievable by an ϵ-reliable and δ-secure (n, k, P) code is lower bounded as [24, Theorem 7 and Section IV.C-1]

$$R(n, \epsilon, \delta) \geq \frac{1}{n} \log_2 \frac{M(\epsilon, n)}{L(n, \delta)},$$

with $M(\epsilon, n)$ the number of codewords for a probability of error ϵ and blocklength n inferred by Shannon's channel coding achievability bound [57, Section III.J-4], and $L(n, \delta)$ such that

$$\sqrt{L(n, \delta)} \triangleq \min_\gamma \frac{\sqrt{\gamma \mathbb{E}[\exp(-|B_n - \log \gamma|)]}}{2(\delta + \mathbb{E}[\exp(-|B_n - \log \gamma|^+)] - 1)}, \quad (6)$$

where the minimization is over all $\gamma > 0$ such that the denominator is positive, and

$$B_n \triangleq \frac{n}{2} \log_2 \left(1 + \frac{P}{\sigma_Z^2}\right)$$
$$+ \frac{\log_2 e}{2} \sum_{t=1}^n \left(1 - \frac{(\sqrt{P}Z_t - \sqrt{\sigma_Z^2})^2}{P + \sigma_Z^2}\right),$$

where Z_t, $t \in \{1, \ldots, n\}$, are i.i.d. according to the standard normal distribution.

B. Converse Bound for the Gaussian Wiretap Channel

An ϵ-reliable and δ-secure (n, k, P) code for the wiretap channel $(\mathcal{X}, P_{YZ|X}, \mathcal{Y} \times \mathcal{Z})$ satisfies [24, Theorem 12 and Section IV.C-3]

$$2^k \leq \inf_{\tau \in (0, 1-\epsilon-\delta)} \frac{\tau + \delta}{\tau \beta_{1-\epsilon-\delta-\tau}(P_{X^nY^nZ^n}, P_{X^nZ^n}Q_{Y^n|Z^n})}, \quad (7)$$

where $P_{X^n Y^n Z^n}$ denotes the joint probability distribution induced by the code and for $Q_{Y^n|Z^n}$ as in [24, Eq. (129)]

$$\beta_{1-\epsilon-\delta-\tau}(P_{X^n Y^n Z^n}, P_{X^n Z^n} Q_{Y^n|Z^n}) \geq \mathbb{P}[D_{n+1} \geq \bar{\gamma}],$$

where

$$
\begin{aligned}
\bar{D}_{n+1} &\triangleq (n+1)C_s \\
&+ \frac{\log_2 e}{2} \sum_{t=1}^{n+1} \left(\frac{N_{Z_t}^2}{\sigma_Z^2} - \frac{(\bar{N}_{Z_t} - c_0(N_{Z_t} + \sqrt{P}))^2}{P + \sigma_Z^2} \right. \\
&\left. + \frac{\bar{N}_{Z_t}^2}{P + \sigma_Y^2} - \frac{(c_1 N_{Z_t} + c_0 \bar{N}_{Z_t} - c_0^2 \sqrt{P})^2}{\sigma_Y^2} \right),
\end{aligned}
$$

with $N_{Z_t} \sim \mathcal{N}(0, \sigma_Z^2)$, $\bar{N}_{Z_t} \sim \mathcal{N}(0, P + \sigma_Y^2)$, $C_s \triangleq \frac{1}{2}\log_2 \frac{1+P/\sigma_Y^2}{1+P/\sigma_Z^2}$, and $c_0 \triangleq \sqrt{\frac{\sigma_Z^2 - \sigma_Y^2}{P + \sigma_Z^2}}$, $c_1 \triangleq \frac{P + \sigma_Y^2}{P + \sigma_Z^2}$, and the threshold $\bar{\gamma}$ satisfies $\mathbb{P}[\bar{B}_{n+1} \geq \bar{\gamma}] = 1 - \epsilon - \delta - \tau$ with

$$
\begin{aligned}
B_{n+1} &\triangleq (n+1)C_s + \frac{\log_2 e}{2} \sum_{t=1}^{n+1} \left(\frac{(N_{Y_t} + \bar{N}_{Y_t})^2}{\sigma_Z^2} - \frac{N_{Y_t}^2}{\sigma_Y^2} \right. \\
&\left. + \frac{(\sqrt{P} + N_{Y_t})^2}{P + \sigma_Y^2} - \frac{(\sqrt{P} + N_{Y_t} + \bar{N}_{Y_t})^2}{P + \sigma_Z^2} \right),
\end{aligned}
$$

where $N_{Y_t} \sim \mathcal{N}(0, \sigma_Y^2)$ and $\bar{N}_{Y_t} \sim \mathcal{N}(0, \sigma_Z^2 - \sigma_Y^2)$, $t \in \{1, \ldots, n+1\}$, are independent and identically distributed.

REFERENCES

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

[2] A. D. Wyner, "The wire-tap channel," *Bell Syst. Tech. J.*, vol. 54, no. 8, pp. 1355–1387, Jan. 1975.

[3] U. Maurer and S. Wolf, "Information-theoretic key agreement: From weak to strong secrecy for free," in *Advances in Cryptology* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2000, pp. 351–368.

[4] I. Csiszár and J. Korner, "Broadcast channels with confidential messages," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 3, pp. 339–348, May 1978.

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

[6] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. W. McLaughlin, and J.-M. Merolla, "Applications of LDPC codes to the wiretap channel," *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2933–2945, Aug. 2007.

[7] A. Subramanian, A. Thangaraj, M. Bloch, and S. W. McLaughlin, "Strong secrecy on the binary erasure wiretap channel using large-girth LDPC codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 585–594, Sep. 2011.

[8] V. Rathi, R. Urbanke, M. Andersson, and M. Skoglund, "Rate-equivocation optimal spatially coupled LDPC codes for the BEC wiretap channel," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2011, pp. 2393–2397.

[9] H. Mahdavifar and A. Vardy, "Achieving the secrecy capacity of wiretap channels using polar codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6428–6443, Oct. 2011.

[10] E. ÅžaÅŸoÄŸlu and A. Vardy, "A new polar coding scheme for strong security on wiretap channels," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 1117–1121.

[11] M. Andersson, R. F. Schaefer, T. J. Oechtering, and M. Skoglund, "Polar coding for bidirectional broadcast channels with common and confidential messages," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 1901–1908, Sep. 2013.

[12] M. Andersson, V. Rathi, R. Thobaben, J. Kliewer, and M. Skoglund, "Nested polar codes for wiretap and relay channels," *IEEE Commun. Lett.*, vol. 14, no. 8, pp. 752–754, Aug. 2010.

[13] M. Hayashi and R. Matsumoto, "Construction of wiretap codes from ordinary channel codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 2538–2542.

[14] M. Bellare, S. Tessaro, and A. Vardy, "Semantic security for the wiretap channel," in *Advances of Cryptology*. Berlin, Germany: Springer, 2012, pp. 294–311.

[15] H. Tyagi and A. Vardy, "Explicit capacity-achieving coding scheme for the Gaussian wiretap channel," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 956–960.

[16] C. Ling, L. Luzzi, J.-C. Belfiore, and D. Stehle, "Semantically secure lattice codes for the Gaussian wiretap channel," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 6399–6416, Oct. 2014.

[17] Y.-P. Wei and S. Ulukus, "Polar coding for the general wiretap channel with extensions to multiuser scenarios," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 278–291, Feb. 2016.

[18] R. A. Chou and M. R. Bloch, "Polar coding for the broadcast channel with confidential messages: A random binning analogy," *IEEE Trans. Inf. Theory*, vol. 62, no. 5, pp. 2410–2429, May 2016.

[19] J. Renes, R. Renner, and D. Sutter, "Efficient one-way secret-key agreement and private channel coding via polarization," in *Advances in Cryptology*. Springer, 2013, pp. 194–213.

[20] T. C. Gulcu and A. Barg, "Achieving secrecy capacity of the wiretap channel and broadcast channel with a confidential component," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1311–1324, Feb. 2017.

[21] R. A. Chou, "Explicit wiretap channel codes via source coding, universal hashing, and distribution approximation, when the channels' statistics are uncertain," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 117–132, 2023.

[22] R. A. Chou, "Explicit codes for the wiretap channel with uncertainty on the Eavesdropper's channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 476–480.

[23] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 124–130, Jun. 2018.

[24] W. Yang, R. F. Schaefer, and H. V. Poor, "Wiretap channels: Nonasymptotic fundamental limits," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4069–4093, Jul. 2019.

[25] M. Hayashi, "Tight exponential analysis of universally composable privacy amplification and its applications," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7728–7746, Nov. 2013.

[26] V. Y. F. Tan, "Achievable second-order coding rates for the wiretap channel," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Nov. 2012, pp. 65–69.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[28] J. L. Carter and M. N. Wegman, "Universal classes of hash functions," *J. Comput. Syst. Sci.*, vol. 18, no. 2, pp. 143–154, Apr. 1979.

[29] G. A. Darbellay and I. Vajda, "Estimation of the information by an adaptive partitioning of the observation space," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, pp. 1315–1321, May 1999.

[30] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, p. 066, Jun. 2004.

[31] T. Suzuki, M. Sugiyama, J. Sese, and T. Kanamori, "Approximating mutual information by maximum likelihood density ratio estimation," in *Proc. Workshop New Challenges Feature Selection Data Mining Knowl. Discovery*, 2008, pp. 5–20.

[32] M. Belghazi et al., "MINE: Mutual information neural estimation," 2018, *arXiv:1801.04062*.

[33] Y. Liang, G. Kramer, H. V. Poor, and S. Shamai, "Compound wiretap channels," *EURASIP J. Wireless Commun. Netw.*, vol. 2009, no. 1, Dec. 2009, Art. no. 142374.

[34] I. Bjelakovic, H. Boche, and J. Sommerfeld, "Secrecy results for compound wiretap channels," *Problems Inf. Transmiss.*, vol. 49, no. 1, pp. 73–98, Jan. 2013.

[35] I. Bjelakovic, H. Boche, and J. Sommerfeld, "Capacity results for arbitrarily varying wiretap channels," in *Information Theory, Combinatorics, and Search Theory*. Berlin, Germany: Springer, 2013, pp. 123–144.

[36] E. MolavianJazi, M. Bloch, and J. Laneman, "Arbitrary jamming can preclude secure communication," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, 2009, pp. 1069–1075.

[37] A. Nooraiepour and T. M. Duman, "Randomized convolutional codes for the wiretap channel," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3442–3452, Aug. 2017.

[38] A. Nooraiepour and T. M. Duman, "Randomized turbo codes for the wiretap channel," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[39] D. Klinc, J. Ha, S. W. McLaughlin, J. Barros, and B.-J. Kwak, "LDPC codes for the Gaussian wiretap channel," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 532–540, Sep. 2011.

[40] M. Baldi, M. Bianchi, and F. Chiaraluce, "Non-systematic codes for physical layer security," in *Proc. IEEE Inf. Theory Workshop*, Aug. 2010, pp. 1–5.

[41] M. Baldi, F. Chiaraluce, N. Laurenti, S. Tomasin, and F. Renna, "Secrecy transmission on parallel channels: Theoretical limits and performance of practical codes," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 11, pp. 1765–1779, Nov. 2014.

[42] W. Harrison et al., "Implications of coding layers on physical-layer security: A secrecy benefit approach," *Entropy*, vol. 21, no. 8, p. 755, 2019.

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

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

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

[46] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[47] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[48] F. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, 2018, pp. 298–303.

[49] M. Goutay, F. Aoudia, and J. Hoydis, "Deep reinforcement learning autoencoder with noisy feedback," in *Proc. Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOPT)*, Jun. 2019, pp. 1–6.

[50] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for channel coding via neural mutual information estimation," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019, pp. 1–5.

[51] H. Ye, G. Y. Li, B.-H.-F. Juang, and K. Sivanesan, "Channel agnostic End-to-End learning based communication systems with conditional GAN," in *Proc. IEEE GLOBECOM Workshops (GC Wkshps)*, Dec. 2018, pp. 1–5.

[52] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for the Gaussian wiretap channel," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[53] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning based wiretap coding via mutual information estimation," in *Proc. 2nd ACM Workshop Wireless Secur. Mach. Learn.*, Jul. 2020, pp. 74–79.

[54] X. Zhang and M. Vaezi, "Deep learning based precoding for the MIMO Gaussian wiretap channel," in *Proc. IEEE GLOBECOM Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

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

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[57] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[58] R. Schaefer, H. Boche, and H. Poor, "Secure communication under channel uncertainty and adversarial attacks," *Proc. IEEE*, vol. 103, no. 10, pp. 1796–1813, Oct. 2015.

[59] H. Boche, R. F. Schaefer, and H. V. Poor, "On the continuity of the secrecy capacity of compound and arbitrarily varying wiretap channels," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2531–2546, Dec. 2015.

[60] R. Chou and A. Yener, "The Gaussian multiple access wiretap channel when the eavesdropper can arbitrarily jam," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2017, pp. 1958–1962.

[61] R. A. Chou and A. Yener, "Gaussian multiuser wiretap channels in the presence of a jammer-aided eavesdropper," *Entropy*, vol. 24, no. 11, p. 1595, Nov. 2022.

Vidhi Rana (Member, IEEE) received the B.Tech. degree in electronics and telecommunication from the College of Engineering, Roorkee, India, and the M.Tech. degree in digital signal processing from the Govind Ballabh Pant Engineering College, Pauri, India. She is currently pursuing the Ph.D. degree with Wichita State University, Wichita, KS, USA.

Rémi A. Chou (Member, IEEE) received the degree in engineering from Supélec, Gif-sur-Yvette, France, in 2011, and the Ph.D. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2015. From 2015 to 2017, he was a Post-Doctoral Scholar at The Pennsylvania State University, University Park, PA, USA. He is currently an Assistant Professor with the Electrical Engineering and Computer Science Department, Wichita State University, Wichita, KS, USA.