Design of Short Blocklength Wiretap Channel Codes: Deep Learning and Cryptography Working Hand in Hand

Vidhi Rana and Rémi A. Chou
Department of Electrical Engineering and Computer Science
Wichita State University, Wichita, KS
vxrana@shockers.wichita.edu and remi.chou@wichita.edu

2021 IEEE Information Theory Workshop (ITW) | 978-1-6654-0312-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITW48936.2021.9611401

***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 via hash functions. For blocklengths smaller than 16, we evaluate through simulations the probability of error at the legitimate receiver and the leakage at the eavesdropper of 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 recent mutual information neural estimator. Simulation results provide examples of codes with positive rates that achieve a leakage inferior to one percent of the message length.**

I. Introduction

The wiretap channel [1] 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 [1], 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 [2], 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 [1], then for arbitrary discrete memoryless channels in [3], and then for Gaussian channels in [4].

Coding schemes based on low-density parity-check (LDPC) codes [5]–[7], polar codes [8]–[11], and invertible extractors [12], [13] have been constructed for degraded and symmetric wiretap channel models. Moreover, the method in [12], [13] has been extended to the Gaussian wiretap channel [14]. Coding schemes based on random lattice codes have also been proposed for the Gaussian wiretap channel [15]. Subsequently, constructive [16]–[18] and random [19] polar coding schemes have been proposed to achieve the secrecy capacity of non-degraded discrete wiretap channels. More recently, a coding scheme that combines polar codes and invertible extractors has also been proposed to avoid the need for a pre-shared secret under strong secrecy [20].

All the references above consider the asymptotic regime, i.e., that n approaches infinity. However, many practical applications require short packets or low latency. 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 [21]–[25].

In this paper, we propose to design short blocklength codes (smaller than 16) for the Gaussian wiretap channel under information-theoretic security guarantees. 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. Our goal is to design codes that can achieve a leakage inferior to one percent of the message length. The main idea of our approach is to the decouple the reliability and secrecy constraints. Specifically, we use a deep learning approach based on a feed-forward neural network autoencoder to handle the reliability code constraint and cryptographic tools, namely, hash functions, to handle the secrecy code constraint. 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 [26], k-nearest neighbor statistics [27], maximum likelihood estimation [28], or variational lower bounds [29]. Instead, to estimate the leakage, we will use a recently proposed estimator called mutual information neural estimator (MINE) [30], which is provably consistent and offer better performances than other known estimators in high dimension.

We summarize the main features offered by our proposed code design as follows. i) A modular approach that separates the code design in 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 modular approach allows a simplified code design, for instance, if only one of the two layers needs to be (re)designed. ii) A precise control of the leakage through the independent (from the reliability layer) design of the

This work is supported in part by NSF grant CCF-2047913.

secrecy layer. Indeed, as discussed next in Section II, code designs based on deep learning that seek to achieve reliability and secrecy jointly do not seem to offer a good control on how small the leakage is. iii) A universal way of dealing with the secrecy constraint through the use of hash functions. This is beneficial, for instance, for compound settings [31] as it then becomes sufficient to design our code with respect to the worst case eavesdropper's channel. iv) 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.

The remaining of the paper is organized as follows. Section II reviews related works. Section III introduces the channel model. Section IV describes our proposed code design and our simulation results. Finally, Section V provides concluding remarks.

II. Related Works

A challenging task for Gaussian wiretap channel coding is code design in the finite blocklength regime. Next, we review known finite-length code constructions based on coding theoretic tools and deep learning tools.

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 [32]–[35]. Specifically, randomized convolutional codes for Gaussian and binary symmetric wiretap channels are studied in [32], and randomized turbo codes for the Gaussian wiretap channel are investigated in [33]. Coding schemes for the Gaussian wiretap channel based on LDPC codes are proposed in [34], [35]. Additionally, another non-information-theoretic security approach called practical secrecy is investigated in [36], where a leakage between Alice's message and an estimate of the message at Eve is estimated.

2) Information-theoretic secrecy metric: Next, we review works that consider the leakage $I(M;Z^n)$ as information-theoretic secrecy metric. In [37], 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 [38], 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 [39], randomized Reed-Muller codes are developed for the Gaussian wiretap channel, and a leakage as low as one percent of the message length has been obtained for a blocklength $n = 16$.

B. Works based on deep learning

Artificial neural networks (NNs) have gained attention in communication system design because their performance approaches the one of the state-of-the-art channel coding solutions. In [40], [41], 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 [42], [43] with reinforcement learning, in [44] with mutual information estimators, and in [45] with generative adversarial networks.

Recently, the autoencoder approach for channel coding has been extended to wiretap channel coding. In [46], [47], 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 [48], a coding scheme for the Gaussian wiretap channel is develop under the information-theoretic leakage $I(M;Z^n)$ with an autoencoder approach that seeks to simultaneously optimized the reliability and secrecy constraints. A leakage as low as 15 percent of the message length is obtained in [48] 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 in a part that only deals with the reliability constraint (by means of an autoencoder) and in another part that only deals with the secrecy constraint (by means of hash functions). As discussed at the end of the introduction and supported by our simulation results, one of the advantages of our approach is a better control of how small the leakage is.

III. Model

Notation: 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. $GF(2^q)$ denotes a finite field of order 2^q, $q \in \mathbb{N}^*$.

Consider a memoryless Gaussian wiretap channel 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 $e : \{0,1\}^k \to \mathbb{B}_0^n(\sqrt{nP})$, and a decoder $d : \mathbb{R}^n \to \{0,1\}^k$.*

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

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

IV. Coding scheme

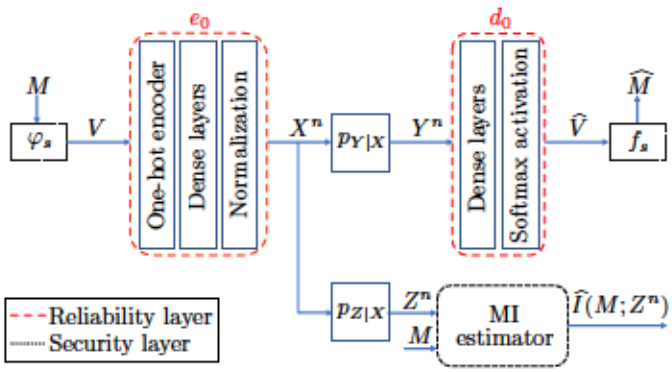

Figure 1. Our proposed coding scheme consists of a reliability layer and a security layer. The reliability layer is implemented using an autoencoder (e_0,d_0) described in Section IV-B, and the security layer is implemented using the functions φ_s and f_s described in Section IV-C1. The security performance is evaluated in terms of the leakage $I(M;Z^n)$ via a mutual information neural estimator described in Section IV-C2.

A. High-level description of 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 autoencoder as described in Section IV-B. We will design two functions φ_s and f_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 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 $e_0(\varphi_s(M,B))$. The overall encoding map e that describes the encoding at the secrecy and reliability layers is described by the map $e : \{0,1\}^k \times \{0,1\}^{q-k} \to \mathbb{B}_0^n(\sqrt{nP}), (M,B) \mapsto e_0(\varphi_s(M,B))$.

Decoding: Given Y^n and s, Bob decodes the message as $f_s(d_0(Y^n))$. The overall decoding map d that describes the decoding at the reliability and secrecy layers is described by the map $d : \mathbb{R}^n \to \{0,1\}^k, Y^n \mapsto f_s(d_0(Y^n))$. 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 mutual information neural estimator as described in Section IV-C2. Our proposed code design is summarized in Fig. 1.

Remark. *Note that a coding strategy that separately handle the reliability and secrecy constraints is also used for the discrete wiretap channel in [12], [13], and for the Gaussian wiretap channel in [14]. In these works, an asymptotic regime is considered, i.e., the blocklength n tends to infinity, and the security layer relies on the random choice of a hash function in a family of universal hash functions. 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 part of the coding scheme design as elaborated on in our simulation results. We also highlight that it is not possible to use the analysis in [14] to guarantee secrecy in our setting at finite blocklength because the reliability layer in our proposed coding scheme does not employ Gaussian codebooks. Instead, we will verify through simulations that our proposed coding scheme satisfies the secrecy constraint in Definition 1.*

B. Design of the reliability layer (e_0,d_0)

The design of the reliability layer consists in designing an ϵ-reliable (q,n,P) code described by the encoder/decoder pair (e_0,d_0) for the channel described by (1). Let $\mathcal{V} \triangleq \{1,2,\ldots,Q\}$ be the message set of this code where $Q \triangleq 2^q$.

(e_0,d_0) is implemented by an autoencoder as in [40]. 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. Intuitively, as any error-correcting code, the autoencoder adds redundancy to the message to ensure recoverability by Bob in the presence of noise. More specifically, an autoencoder as in [40] is a deep neural network that models a communication system with Alice's message as input and Bob's message estimate as output. As depicted in Fig. 1, the encoder consists of (i) an embedding layer 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 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 $P=1$ because 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 Fig. 1, 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 $p^{|\mathcal{V}|} \in (0,1)^{|\mathcal{V}|}$, where the entries p_v, $v \in \mathcal{V}$, are $p_v \triangleq \exp(\mu_v)\left(\sum_{t=1}^{|\mathcal{V}|}\exp(\mu_t)\right)^{-1}$. Finally, the decoded message $\hat{v}$ corresponds to the index of $p^{|\mathcal{V}|}$ associated with the highest probability, i.e., $\hat{v} \triangleq \arg\max_v p_v$. Similar to [40], the autoencoder is trained over of all possible messages $v \in \mathcal{V}$ using a stochastic gradient descent (SGD) [49] and the categorical cross-entropy loss function $L_{\text{loss}} \triangleq -\log(p_v)$.

C. Design of the security layer (φ_s, f_s)

The objective is here to design (φ_s, f_s) such that the total amount of leaked information about the original message is small, i.e., $I(M;Z^n) \leq \delta$, for some $\delta > 0$. For a given choice of (φ_s, f_s), the performance of our code construction will be evaluated using a mutual information neural estimator (MINE) [30]. Before we describe the construction of (φ_s, f_s), we review the definition of 2-universal hash functions.

Definition 3. *[50] 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, f_s): Let $\mathcal{S} \triangleq \{0,1\}^q \backslash \{\mathbf{0}\}$. For $k \leq q$, consider the 2-universal hash family of functions $\mathcal{F} \triangleq \{f_s\}_{s\in\mathcal{S}}$, where for $s \in \mathcal{S}$, $f_s : \{0,1\}^q \to \{0,1\}^k, v \mapsto (s \odot v)_k$,where $\odot$ is the multiplication in $GF(2^q)$ and $(\cdot)_k$ selects the k most significant bits. In our proposed code design, the security layer is handled via a specific function $f_s \in \mathcal{F}$ indexed by the seed $s \in \mathcal{S}$. Then, we define $\varphi_s : \{0,1\}^k \times \{0,1\}^{q-k} \to \{0,1\}^q, (m,b) \mapsto s^{-1} \odot (m\|b)$, 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 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 of $\widehat{M}$ of M, i.e., $\widehat{M} \triangleq f_s(\widehat{V})$.

2) Leakage evaluation via Mutual Information Neural Estimator (MINE) [30]: Let $\mathcal{F} \triangleq \{T_\theta\}_{\theta\in\Theta}$ be a set of functions paramaterized by a deep neural network with parameters $\theta \in \Theta$. Define the neural information measure

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

where p_{MZ^n} is the joint probability distribution of (M,Z^n). By [30], the neural information measure $I_\Theta(p_{MZ^n})$ 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

$$\widehat{I}(M;Z^n) \triangleq \sup_{\theta\in\Theta} \frac{1}{k}\sum_{t=1}^{k}[T_\theta(m_t, z_t^n)] - \log \frac{1}{k}\sum_{t=1}^{k}[e^{T_\theta(\bar{m}_i, z_i^n)}],$$

where the term $\frac{1}{k}\sum_{t=1}^{k}[T_\theta(m_t, z_t^n)]$ represents a sample mean using k samples $(m_t, z_t^n)_{t\in\{1,\ldots,k\}}$ from p_{MZ^n}, and the term $\frac{1}{k}\sum_{t=1}^{k}[e^{T_\theta(m_i, z_i^n)}]$ represents a sample mean using k samples $(\bar{m}_t, \bar{z}_t^n)_{t\in\{1,\ldots,k\}}$ from $p_M p_{Z^n}$.

The goal of MINE is to design T_θ such that $\widehat{I}(M;Z^n)$ approaches the mutual information $I(M;Z^n)$. By [30], the estimator $\widehat{I}(M;Z^n)$ converges to $I(M;Z^n)$ when the number of samples is sufficiently large [30]. Guidelines to implement the estimator $\widehat{I}(M;Z^n)$ are provided in [30].

D. Simulations and examples of code designs

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 the Keras 2.3.1 library and Tensorflow 2.2.0.

1) Autoencoder training for the design of the reliability layer (e_0, d_0): We consider the channel model (1) with $\sigma_Y^2 = 10^{-\text{SNR}_B/10}$ and $\text{SNR}_B = 10dB$, where, without loss of generality (see Section IV-B), we chose $P = 1$. The autoencoder is trained for $q = n-1$ using SGD with the Adam optimizer [49] at a learning rate of 0.001 over 200 epochs of $40,000$ random messages with a batch size of 500. 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. Fig. 2 shows $\mathbf{P}_e(e_0, d_0)$ versus the blocklength n.

2) Design of the secrecy layer and leakage evaluation: The seeds selected for the simulations are given in Table I. The seeds are picked at random for the values of n greater than eight, and multiple seeds have been tested for the values of n smaller than eight to minimize the leakage. The leakage is evaluated using MINE as follows. We have used a fully connected feed-forward neural network with 4 hidden layers, each having 400 neurons and using rectified linear unit (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 $e = \varphi_s \circ e_0$, 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. Fig. 3 shows an example of leakage estimation

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	0111	0111
6	01000	01000
7	001001	010010
8	1011100	0000001
9	00000001	00001001
10	000000001	000000001
11	0000000011	0000000011
12	10000000000	10000000000
13	1000000000000	0000000000001
14	10000000000000	00000000000001
15	100000000000000	000000000000001

for $k = 1$ and $n = 7$ at $\text{SNR}_E = -5$dB, and Fig. 4 shows the leakage versus the blocklength n for different values of k and SNR_E.

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 $M \in \{0,1\}^k$ is the message, and $B \in \{0,1\}^{q-k}$ is a sequence of $q-k$ random bits generated uniformly at random. The trained decoder d_0 forms $\widehat{M} \triangleq f_s(d_0(Y^n))$, as described in Section IV-C. Fig. 2 shows $\mathbf{P}_e(e,d)$ versus the blocklength n. Note that we only plotted $\mathbf{P}_e(e,d)$ when $k = 2$ 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.

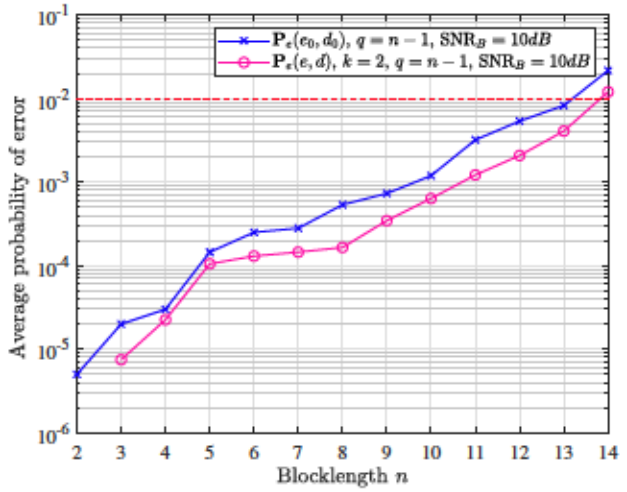

Figure 2. Average probability of error versus blocklength n when $q = n$ 1 and $\text{SNR}_B = 10$dB.

4) Discussion: As expected, Fig. 4 shows that the leakage decreases as SNR_E decreases (for a fixed n, q, and k) and increases as the length of the message k increases (for a fixed n, q, and SNR_E). From Fig. 2 and 4, we, for instance, see that

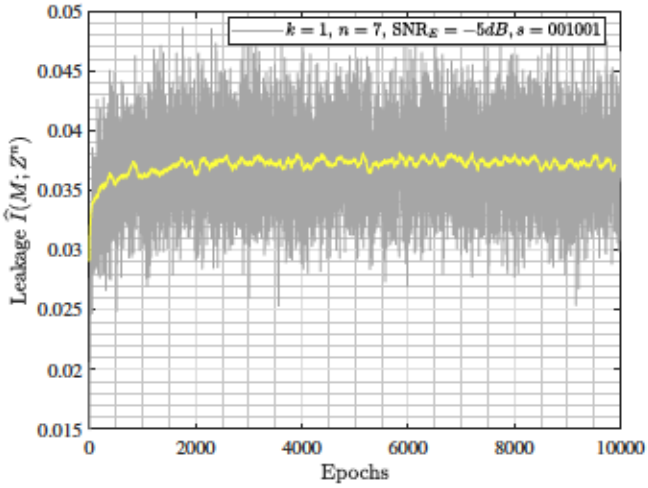

Figure 3. Example of leakage $\widehat{I}(M;Z^n)$ versus epochs when $q = n$ 1. The yellow curve represents the 100-sample moving average of $\widehat{I}(M;Z^n)$.

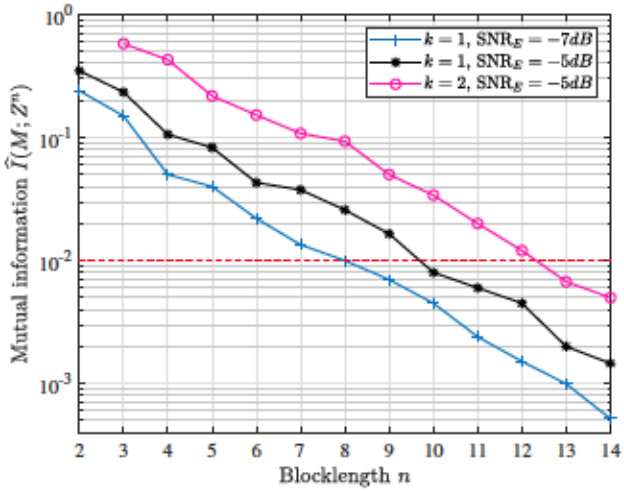

Figure 4. Leakage $\widehat{I}(M;Z^n)$ versus blocklength n when $q = n$ 1.

for $\text{SNR}_B = 10$dB and $\text{SNR}_E = -5$dB, we have designed codes that show that the rate $R = \frac{1}{10}$ is $(\epsilon = 1.19 \cdot 10^{-3}, \delta = 8 \cdot 10^{-3})$-achievable with blocklength $n = 10$, and the rate $R = \frac{2}{13}$ is $(\epsilon = 4.07 \cdot 10^{-3}, \delta = 6.73 \cdot 10^{-3})$-achievable with blocklength $n = 13$. As another example, from Fig. 2 and 4, we also see that for $\text{SNR}_B = 10$dB and $\text{SNR}_E = -7$dB, we have designed codes that show that the rate $R = \frac{1}{8}$ is $(\epsilon = 5.35 \cdot 10^{-4}, \delta = 10^{-2})$-achievable with blocklength $n = 8$.

V. CONCLUDING REMARKS

We highlight that our code design method can be applied to any channel model and does not require the knowledge of the channel model but only the knowledge of input and output channel samples. Unreported results also show that our code design is applicable to settings where uncertainty holds on the channel statistics, e.g., compound wiretap channels [31], and arbitrarily varying wiretap channels [51].

REFERENCES

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

[2] U. Maurer and S. Wolf, "Information-theoretic key agreement: From weak to strong secrecy for free," in *Lecture Notes in Computer Science*, 2000, pp. 351–368.

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

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

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

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

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

[8] 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, 2011.

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

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

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

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

[13] M. Bellare, S. Tessaro, and A. Vardy, "Semantic security for the wiretap channel," *Advances of cryptology*, pp. 294–311, 2012.

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

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

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

[17] R. Chou and M. 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, 2016.

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

[19] T. Gulcu and A. Barg, "Achieving secrecy capacity of the wiretap channel and broadcast channel with a confidential component," in *IEEE Inf. Theory Workshop*, 2015, pp. 1–5.

[20] R. Chou, "Unified framework for polynomial-time wiretap channel codes," 2020. [Online]. Available: https://arxiv.org/pdf/2002.01924.pdf

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

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

[23] V. Tan, "Achievable second-order coding rates for the wiretap channel," in *IEEE Int. Conf. Commun. Systems*, 2012, pp. 65–69.

[24] M. Hayashi, H. Tyagi, and S. Watanabe, "Strong converse for a degraded wiretap channel via active hypothesis testing," in *Annual Allerton Conf. Commun., Control, and Computing*, 2014, pp. 148–151.

[25] V. Tan and M. Bloch, "Information spectrum approach to strong converse theorems for degraded wiretap channels," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 9, pp. 1891–1904, 2015.

[26] G. 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, 1999.

[27] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E*, vol. 69, no. 6, p. 066138, 2004.

[28] T. Suzuki, M. Sugiyama, J. Sese, and T. Kanamori, "Approximating mutual information by maximum likelihood density ratio estimation," in *New challenges for feature selection in data mining and knowledge discovery*, 2008, pp. 5–20.

[29] D. Barber and F. Agakov, "The im algorithm: A variational approach to information maximization," in *Int. Conf. Neural Inf. Processing Systems*, 2003, pp. 201–208.

[30] M. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. Hjelm, "Mine: Mutual information neural estimation," 2018, https://arxiv.org/abs/1801.04062.

[31] Y. Liang, G. Kramer, H. Poor, and S. Shamai, "Compound wiretap channels," *EURASIP J. Wirel. Commun. Netw.*, no. 142374, 2009.

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

[33] ——, "Randomized turbo codes for the wiretap channel," in *IEEE Global Commun. Conf.*, 2017, pp. 1–6.

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

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

[36] W. Harrison, E. Beard, S. Dye, E. Holmes, K. Nelson, M. Gomes, and J. Vilela, "Implications of coding layers on physical-layer security: A secrecy benefit approach," *Entropy*, vol. 21, no. 8, p. 755, 2019.

[37] 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.

[38] 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.

[39] 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.

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

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

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

[43] M. Goutay, F. Aoudia, and J. Hoydis, "Deep reinforcement learning autoencoder with noisy feedback," 2018, arXiv preprint arXiv:1810.05419.

[44] R. Fritschek, R. Schaefer, and G. Wunder, "Deep learning for channel coding via neural mutual information estimation," in *IEEE Int. Workshop Signal Processing Advances Wirel. Commun.*, 2019, pp. 1–5.

[45] H. Ye, G. Li, F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional GAN," in *IEEE Globecom Workshops*, 2018, pp. 1–5.

[46] R. Fritschek, R. Schaefer, and G. Wunder, "Deep learning for the Gaussian wiretap channel," in *IEEE Int. Conf. Commun.*, 2019, pp. 1–6.

[47] ——, "Deep learning based wiretap coding via mutual information estimation," in *ACM Workshop on Wireless Security and Machine Learning*, 2020, pp. 74–79.

[48] 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.

[49] D. Kingma and J. Ba, "A method for stochastic optimization," 2014, arXiv preprint arXiv:1412.6980.

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

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