Randomized Convolutional Codes for the Wiretap Channel

Alireza Nooraiepour and Tolga M. Duman, *Fellow, IEEE*

***Abstract*— We study application of convolutional codes to the randomized encoding scheme introduced by Wyner as a way of confusing the eavesdropper over a wiretap channel. We describe optimal and practical sub-optimal decoders for the main and the eavesdropper's channels, and estimate the security gap, which is used as the main metric. The sub-optimal decoder works based on the trellis of the code generated by a convolutional code and its dual, where one encodes the data bits and the other encodes the random bits. By developing a code design metric, we describe how these two generators should be selected for optimal performance over a Gaussian wiretap channel. We also propose application of serially concatenated convolutional codes to this setup so as to reduce the resulting security gaps. Furthermore, we provide an analytical characterization of the system performance by extending existing lower and upper bounds for coded systems to the current randomized convolutional coding scenario. We illustrate our findings via extensive simulations and numerical examples, which show that the newly proposed coding scheme can outperform the other existing methods in the literature in terms of security gap.**

***Index Terms*— Convolutional codes, code concatenation, turbo codes, randomized encoding, wiretap channel, security gap.**

I. Introduction

THE wiretap channel introduced by Wyner [1] is a basic model for studying secure communications. The system consists of one transmitter (Alice) and two receivers: a legitimate receiver (Bob) and an eavesdropper (Eve) connected to the transmitter through the main and the eavesdropper's channels, respectively. In his original work, Wyner introduces a metric called equivocation indicating how much information is leaked to the eavesdropper about the original message as a measure of its confusion and points out that a system designer wants to make the probability of decoding error over the main channel arbitrarily small (reliability constraint) while the normalized mutual information $\frac{1}{n}I(M;Z^n)$ goes to zero (security constraint) where M is the transmitted message by Alice and Z^n denotes the eavesdropper's observation. Wyner defines the notion of secrecy capacity C_s as the maximum achievable transmission rate that satisfies both the security and the reliability constraints simultaneously. He also proves that one can achieve the secrecy capacity using a randomized encoding scheme at the transmitter which is the main source of confusion for the eavesdropper [1]. This encoding method is often referred to as *coset-coding* and is studied further in the subsequent literature, e.g., in [2].

Inspired by this method, application of low density parity check (LDPC) codes to the wiretap channel is studied in [3]. The authors prove that using capacity approaching codes for each secret message over the eavesdropper's channel can achieve the secrecy capacity, asymptotically. More practically, when the main channel is noiseless and the eavesdropper's channel is a binary erasure channel, they point out that using dual of an LDPC code and its cosets can satisfy the security constraint without the need for capacity approaching codes. Application of lattice codes in the context of physical layer security is studied in [4] where the authors define a secrecy gain metric which was related to the theta series of lattices and show the amount of confusion at the eavesdropper. Without introducing a decoding method, they evaluate the performance of different lattices based on the secrecy gain. The confusion at the eavesdropper in [4] is the result of using a random lattice in addition to the lattice which is responsible for transmitting the original message. Furthermore, the application of polar codes to the wiretap channel is studied in [5] where the channel polarization phenomenon of polar codes enables the proposal of a practical coding scheme based on coset-coding which achieves secrecy capacity when both main and eavesdropper's channels are binary symmetric. We emphasize that the coding schemes proposed in [3] and [5] use an information-theoretic metric to measure physical layer security while the authors in [4] use an alternative one.

For the case of additive white Gaussian noise (AWGN) channels, secrecy capacity equals to the difference between the capacities of the main and the eavesdropper's channels, and for it to be greater than zero the signal to noise ratio (SNR) of the main channel must be larger than that of the eavesdropper's channel [6]. In this context, an important parameter is the difference between the qualities of the main and eavesdropper's channels (dubbed as *security gap*) needed for achieving physical layer security [7]. Small security gaps are desirable because they make physical layer security achievable even with a small degradation of the eavesdropper's channel with respect to the main one. By denoting the bit error rates (BERs) calculated through the main and eavesdropper's channels by P_{main} and P_{eve}, respectively, one can

Manuscript received August 5, 2016; revised December 10, 2016, February 26, 2017, and May 4, 2017; accepted May 4, 2017. Date of publication May 16, 2017; date of current version August 14, 2017. This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under the grant 113E223. Part of this paper is submitted for presentation at the 2017 IEEE GLOBECOM. The associate editor coordinating the review of this paper and approving it for publication was Y.-W. P. Hong. *(Corresponding author: Tolga M. Duman.)*

The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, TR-06800 Ankara, Turkey (e-mail: nooraiepour@ee.bilkent.edu.tr; duman@ee.bilkent.edu.tr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2017.2704586

use alternative reliability and security constraints as follows: $P_{main} \leq P_{main}^{max}$ (≈ 0) and $P_{eve} \geq P_{eve}^{min}$ (≈ 0.5) where P_{main}^{max} and P_{eve}^{min} represent the maximum and minimum desired BERs for Bob and Eve, respectively. Denoting the lowest SNR which satisfies the reliability constraint by SNR_{main} and the largest SNR which satisfies the security constraint by SNR_{eve}, the security gap measured in dBs is defined as $\text{SNR}_{\text{main}} - \text{SNR}_{\text{eve}}$.

Several practical coding schemes aiming at reducing the security gap have been proposed in the literature. Specifically, punctured LDPC codes are exploited for physical layer security in [7]. Furthermore, [8] demonstrates that using non-systematic codes obtained from scrambling information bits of a systematic code are quite effective to reduce the security gap, while [9] applies different techniques including scrambling, concatenation, and hybrid automatic repeat-request to LDPC and BCH codes in order to further reduce the security gap.

In this paper, we describe how convolutional codes can be applied to Wyner's randomized encoding method, evaluate the performance of finite length (terminated) randomized convolutional codes over Gaussian and binary symmetric wiretap channels, and provide optimal and practical sub-optimal decoders for use at the receivers. We argue that the concept of dual of a convolutional code plays a crucial role in this set-up. Furthermore, we construct randomized serially concatenated convolutional codes (RSCCCs) based on the proposed randomized convolutional codes. Finally, using existing algorithms for computing the distance spectra of convolutional codes [10], we provide lower and upper bounds on the performance of the randomized convolutional codes in terms of codeword error probabilities, by utilizing Seguin's lower bound and tangential sphere bound [11], [12].

The rest of the paper is organized as follows: the channel model is introduced in Section II. The encoding scheme and convolutional code design for the randomized coding scheme are given in Section III. The optimal and several sub-optimal decoders are presented in Section IV. Development of randomized serially concatenated convolutional codes is studied in Section V. Lower and upper bounds on the error rate performance of the proposed system are developed in Section VI. Extensive numerical examples are provided in Section VII, and finally, the paper is concluded in Section VIII.

II. Channel Model

For the Gaussian case, we assume that both the main and eavesdropper's channels are additive white Gaussian noise (AWGN) channels and express the input-output relationship for a single use of the channel as $y = x + N$ where $x = (-1)^c$ is the binary phase-shift keying (BPSK) modulated version of the transmitted bit c. N represents the Gaussian noise with zero mean and variance $N_0/2$. We also assume that different noise components are independently and identically distributed (i.i.d.). In this set-up SNR (E_s/N_0) equals $E_b R/N_0$ where E_b denotes the energy per bit and R is the transmission rate. We emphasize that this model is used for both the main and eavesdropper's channels (with different noise power levels).

For the binary symmetric channel (BSC) case, both channels are BSC with different cross-over probabilities.

III. Randomized Convolutional Codes–Encoding

A. Randomized Encoding Method

To construct a randomized encoding scheme which aims to confuse the eavesdropper, we assign a coset to each message being transmitted. To transmit a k-bit message we need 2^k cosets. Suppose that there are 2^r codewords in each coset. Then, we need a linear code of length n and dimension at least $k + r$ (assuming $k + r \leq n$) which we call the *big code* to cover all the codewords. In this manner, each coset consists of a unique set of codewords and no n-tuple can be found which belongs to more than one coset. We choose a terminated convolutional code $\mathcal{C}(n, r)$ (with length n and dimension r) as the first coset which we call the *small code* with generators $\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_r$ where the $\mathbf{g}_i$'s are $1 \times n$ vectors. To generate the remaining $2^k - 1$ cosets with unique codewords, we identify linearly independent n-tuples outside $\mathcal{C}$ which we denote by $\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_k$.

A message denoted by *data bits* $\mathbf{s} = [s_1, s_2, ..., s_k]$ is mapped to the coset obtained by $s_1\mathbf{h}_1 + s_2\mathbf{h}_2 + ... + s_k\mathbf{h}_k + \mathcal{C}$ which makes the transmission rate $R = k/n$. Finally, the transmitted codeword $\mathbf{c}$ of length n is determined by choosing a random codeword in $\mathcal{C}$ which is done using a random vector denoted by $\mathbf{v} = [v_1, v_2, ..., v_r]$ (where v_i's are i.i.d. 0's and 1's each with probability 1/2) as follows [3]

$$\mathbf{c} = s_1\mathbf{h}_1 + s_2\mathbf{h}_2 + ... + s_k\mathbf{h}_k + v_1\mathbf{g}_1 + v_2\mathbf{g}_2 + ... + v_r\mathbf{g}_r. \quad (1)$$

This method requires two sets of generators to encode the message: one for random bits (v_i's) and one for data bits (s_i's). It is desirable to select the $\mathbf{h}_i$'s and the $\mathbf{g}_i$'s such that the reliability and security constraints are satisfied.

Given the generators of $\mathcal{C}$ ($\mathbf{g}_i$'s), obtaining $\mathbf{h}_i$'s requires an exhaustive search which is not practical for medium to large length codes. Here, we introduce a practical way to attack this problem by first defining what we refer to as *pseudo-self-dual* codes.

Definition 1: A linear code $\mathcal{C}(n, r)$ with generator matrix $\mathbf{G}$ is called *pseudo-self-dual* if $\mathbf{G}\mathbf{G}^T$ is rank-deficient.

Theorem 1: Suppose $\mathcal{C}^{\perp}(n, n - r)$ is the dual of linear code $\mathcal{C}(n, r)$. The non-zero codewords of $\mathcal{C}^{\perp}$ and $\mathcal{C}$ are different if $\mathcal{C}^{\perp}$ is *not* pseudo-self-dual.

Proof: Let us denote the generator matrices of $\mathcal{C}$ and $\mathcal{C}^{\perp}$ with $\mathbf{G}$ and $\mathbf{G}^{\perp}$, respectively. Assume that there is a non-zero codeword that belongs to both of these codes, so there should be non-zero vectors $\mathbf{u}$ and $\mathbf{v}$ such that $\mathbf{u}\mathbf{G} = \mathbf{v}\mathbf{G}^{\perp}$. Right multiplying both sides with $(\mathbf{G}^{\perp})^T$, we obtain $\mathbf{u}\mathbf{G}(\mathbf{G}^{\perp})^T = \mathbf{v}\mathbf{G}^{\perp}(\mathbf{G}^{\perp})^T$ which results in $\mathbf{v}\mathbf{G}^{\perp}(\mathbf{G}^{\perp})^T = \mathbf{0}$ since $\mathcal{C}$ and $\mathcal{C}^{\perp}$ are duals of each other. But the last equality is in contradiction with the assumption that $\mathcal{C}^{\perp}$ is not pseudo-self-dual concluding the proof. ∎

We recall that two conditions need to be satisfied for $\mathbf{h}_i$'s: 1) they should not be codewords in the small code $\mathcal{C}$; 2) they should be linearly independent. Using Theorem 1, by choosing generators of $\mathcal{C}^{\perp}$ as the $\mathbf{h}_i$'s, the first condition is satisfied if $\mathcal{C}^{\perp}$ is not pseudo-self-dual, and the second condition is satisfied since they are generators of a linear code ($\mathcal{C}^{\perp}$).

Theorem 1 implies that it is not always possible to use generators of $\mathcal{C}^{\perp}$ to construct the cosets of $\mathcal{C}$. As an example, let

us consider the small code $\mathcal{C}$ to be a single parity check (SPC) code ($n = 8$, $k = 7$, $d_{min} = 2$). $\mathcal{C}^{\perp}$ is then the repetition code ($n = 8$, $k = 1$, $d_{min} = 8$) which has only one generator: $G^{\perp} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$. But this generator is a codeword in $\mathcal{C}$ which means using it in (1) only reproduces the small code $\mathcal{C}$ and will not result in a new coset. In this example, we note that $G^{\perp}(G^{\perp})^T$ is rank-deficient, i.e., $\mathcal{C}^{\perp}$ is pseudo-self-dual.

We note that one of the main motivations for using convolutional codes is that the big code formed by two convolutional codes ($\mathcal{C}$ and $\mathcal{C}^{\perp}$) is another convolutional code as will be discussed in Section IV-B.2. Hence, their trellis structures enable us to propose efficient sub-optimal decoders which are necessary in practice. Furthermore, one can extend this idea to develop randomized concatenated convolutional codes for use in physical layer security (see Section V). Finally, by utilizing the distance spectra of convolutional codes [10], we can obtain lower and upper bounds on the codeword error rates in the randomized encoding setup (see Section VI) which are important for a theoretical characterization of the performance at the eavesdropper and the main user, respectively.

B. Dual of a Convolutional Code

Based on Theorem 1, we can use the dual of a convolutional code for the randomized encoding scheme if it is not pseudo-self-dual. In this subsection, we describe how the dual of a convolutional code can be obtained in a systematic way.

For a binary convolutional encoder of rate a/b and memory m, the information sequence $\mathbf{u} = \mathbf{u}_0\mathbf{u}_1\mathbf{u}_2...$ ($\mathbf{u}_i$'s are $1 \times a$) and the encoded sequence $\mathbf{v} = \mathbf{v}_0\mathbf{v}_1\mathbf{v}_2...$ ($\mathbf{v}_i$'s are $1 \times b$) satisfy

$$\mathbf{v}_t = \mathbf{u}_t\mathbf{G}_0 + \mathbf{u}_{t-1}\mathbf{G}_1 + ... + \mathbf{u}_{t-m}\mathbf{G}_m \tag{2}$$

where $\mathbf{G}_i$ is an $a \times b$ binary matrix. That is, one can write $\mathbf{v} = \mathbf{uG}$ with

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_m & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_m & \\ & & \ddots & \ddots & & \ddots \end{bmatrix}. \tag{3}$$

The generator matrix of the dual code which is of rate $(b-a)/a$ can be written as

$$\mathbf{G}^{\perp} = \begin{bmatrix} \mathbf{G}_0^{\perp} & \mathbf{G}_1^{\perp} & \dots & \mathbf{G}_{m^{\perp}}^{\perp} & & \\ & \mathbf{G}_0^{\perp} & \mathbf{G}_1^{\perp} & \dots & \mathbf{G}_{m^{\perp}}^{\perp} & \\ & & \ddots & \ddots & & \ddots \end{bmatrix} \tag{4}$$

with $\mathbf{G}(\mathbf{G}^{\perp})^T = \mathbf{0}$ where $m^{\perp}$ denotes the memory of the dual code. We now restate a result from [13].

Definition 2: The *reverse* of a convolutional code $\mathcal{C}$ with polynomial generator $\mathbf{G}(D) = \mathbf{G}_0 + \mathbf{G}_1 D + ... + \mathbf{G}_m D^m$ is defined as the convolutional code $\widetilde{\mathcal{C}}$ with polynomial generator $\widetilde{\mathbf{G}}(D) = \mathbf{G}_m + \mathbf{G}_{m-1} D + ... + \mathbf{G}_0 D^m$.

Theorem 2: (Taken from [13]) The dual of a convolutional code $\mathcal{C}$ with polynomial generator $\mathbf{G}(D)$ has a polynomial generator of the form $\widetilde{\mathbf{H}}(D)$ where $\mathbf{G}(D)(\mathbf{H}(D))^T = \mathbf{0}$.

Proof: For completeness, we provide a brief proof of this result. Let $\mathbf{G}(D) = \mathbf{G}_0 + \mathbf{G}_1 D + ... + \mathbf{G}_m D^m$ and denote the polynomial generator of its dual $\mathcal{C}^{\perp}$ by $\mathbf{G}^{\perp}(D) = \mathbf{G}_0^{\perp} + \mathbf{G}_1^{\perp} D + ... + \mathbf{G}_{m^{\perp}}^{\perp} D^{m^{\perp}}$. The polynomial generator of the reverse of $\mathcal{C}^{\perp}$ is determined as $\widetilde{\mathbf{G}}^{\perp}(D) = \mathbf{G}_{m^{\perp}}^{\perp} + \mathbf{G}_{m^{\perp}-1}^{\perp} D + ... + \mathbf{G}_0^{\perp} D^{m^{\perp}}$. Consider

$$\begin{aligned} &\mathbf{G}(D)(\widetilde{\mathbf{G}}^{\perp}(D))^T \\ &\quad = \mathbf{G}_0(\mathbf{G}_{m^{\perp}}^{\perp})^T + (\mathbf{G}_0(\mathbf{G}_{m^{\perp}-1}^{\perp})^T \\ &\qquad + \mathbf{G}_1(\mathbf{G}_{m^{\perp}}^{\perp})^T) D + \cdots + \mathbf{G}_m(\mathbf{G}_0^{\perp})^T D^{m+m^{\perp}} \end{aligned} \tag{5}$$

One can see that the coefficients of D^i (for all i's) in (5) are elements of the matrix $\mathbf{G}(\mathbf{G}^{\perp})^T$ which are zero since $\mathbf{G}$ and $\mathbf{G}^{\perp}$ are duals of each other, i.e., $\mathbf{G}(D)(\widetilde{\mathbf{G}}^{\perp}(D))^T = \mathbf{0}$ which results in $\mathbf{H}(D) = \widetilde{\mathbf{G}}^{\perp}(D)$ or equivalently, $\mathbf{G}^{\perp}(D) = \widetilde{\mathbf{H}}(D)$ concluding the proof. ∎

To use Theorem 2, we need to compute $\mathbf{H}(D)$ based on $\mathbf{G}(D)$ such that $\mathbf{G}(D)(\mathbf{H}(D))^T = \mathbf{0}$. A straightforward way is to convert $\mathbf{G}(D)$ to systematic form by row operations. Having $\mathbf{G}_{sys}(D) = [\mathbf{I}_k|\mathbf{P}(D)]$ one can write $\mathbf{H}_{sys}(D) = [\mathbf{P}^T(D)|\mathbf{I}_{n-k}]$ where $\mathbf{I}$ is the identity matrix and some elements of $\mathbf{H}_{sys}(D)$ are rational functions of D. Multiplying $\mathbf{H}_{sys}(D)$ by a suitable polynomial will remove the denominators and will result in $\mathbf{H}(D)$.

As a simple example, if $\mathbf{G}(D) = [1 + D + D^2 \quad 1 + D^2]$ then $\mathbf{H}(D) = [1 + D^2 \quad 1 + D + D^2]$. Using Theorem 2, we get $\mathbf{G}^{\perp}(D) = \widetilde{\mathbf{H}}(D) = [1 + D^2 \quad 1 + D + D^2]$. Hence, the dual of a [7 5][1] convolutional code with memory 2 is the [5 7] code. Similarly, the dual of [117 155] with memory 6 is [133 171]. For these two cases, one can also verify that $\mathbf{G}^{\perp}$ is not *pseudo-self-dual* which makes them suitable for the proposed encoding scheme.

C. Obtaining a Subset of Convolutional Codes

As discussed in Section III-A, the codewords in each coset represent a single message and are aimed at confusing the eavesdropper. If the main channel is noiseless, we are not concerned with the decoding process at the legitimate receiver, and we only want to confuse the eavesdropper. In this case, it is desirable to use as many codewords as possible in each coset. If the main channel is also noisy, then one should consider reducing the number of codewords in each coset in order to increase the error correction capabilities at the legitimate receiver. As discussed in Section III-A, the number of codewords in each coset is governed by the small code $\mathcal{C}(n, r)$ introduced in Section III-A and equals 2^r assuming that the random bits are being encoded by generators of the small code.

Let $\mathcal{C}$ be a convolutional code of rate a/b with the generator matrix $\mathbf{G}(D)$ with a rows. After finding the equivalent generator matrix $\mathbf{G}^{[k]}(D)$ to $\mathbf{G}(D)$ with rate ka/kb for $k = 2, 3, \dots,$ one can obtain a subset of $\mathcal{C}$ by choosing different rows from the ka available rows of $\mathbf{G}^{[k]}(D)$. Clearly, the resulting convolutional code has a smaller rate than $\mathcal{C}$ and it offers improved error correction capabilities.

We now explain how one can obtain an equivalent generator matrix $\mathbf{G}^{[k]}(D)$ with rate k/bk, $k = 2, 3, \dots$ for a convolutional code with generator matrix $\mathbf{G}(D)$ of rate $1/b$. The extension of the method to the general case (for a rate a/b code) is quite straightforward. $\mathbf{G}^{[k]}(D)$ accepts k input bits

[1]Throughout this paper, we denote convolutional codes with octal notation.

in each time slot. The input bits u_i's are fed to the encoder in the following manner

$$\begin{array}{ccccccc} \dots & u_{i+3k-1} & u_{i+2k-1} & u_{i+k-1} & \rightarrow & \mathbf{g}_1 \\ \dots & u_{i+3k-2} & u_{i+2k-2} & u_{i+k-2} & \rightarrow & \mathbf{g}_2 \\ & \vdots & \vdots & \vdots & & \vdots \\ \dots & u_{i+2k+1} & u_{i+k+1} & u_{i+1} & \rightarrow & \mathbf{g}_{k-1} \\ \dots & u_{i+2k} & u_{i+k} & u_i & \rightarrow & \mathbf{g}_k \\ \dots & D^2 & D & 1 & & \end{array} \tag{6}$$

where "$\rightarrow\ \mathbf{g}_i$" means that the bits are being fed to a specific generator $\mathbf{g}_i$ (a row of $\mathbf{G}^{[k]}(D)$), and the last row denotes the delay associated with the input bits in each column. We denote the output of the encoder corresponding to $\mathbf{G}(D)$ to the input u_{i+f} by $\mathbf{v}_f$ whose elements are $v_{f,j}$ where $0 \leq f \leq k-1$ and $1 \leq j \leq b$. Furthermore, we consider the corresponding output of $\mathbf{G}^{[k]}(D)$ to the input vector $[u_i\ u_{i+1}\ \dots u_{i+k-1}]$ as $\left[\mathbf{o}_0\ \mathbf{o}_1\ \dots\ \mathbf{o}_{k-1}\right]$ where each $\mathbf{o}_f$ is a vector consisting of b sequences, and each sequence is the sum of the delayed u_i's produced through the k generators within the structure in (6). $\mathbf{G}^{[k]}(D)$ and $\mathbf{G}(D)$ are equivalent if

$$\mathbf{v}_f = \mathbf{o}_{k-f-1},\ 0 \leq f \leq k-1 \tag{7}$$

where $\mathbf{v}_f = u_{i+f}\mathbf{G}(D)$ which is known since $\mathbf{G}(D)$ is given. We note that each element of $\mathbf{o}_i$ is produced by a column of $\mathbf{G}^{[k]}(D)$. Hence, each of the bk equations in (7) determines the suitable k generators, $\mathbf{g}_i$'s, $1 \leq i \leq k$ needed for the corresponding column of $\mathbf{G}^{[k]}(D)$.

Example 1: Consider the [561 753] convolutional code of memory $m = 8$ and rate $1/2$, i.e., $\mathbf{G}(D) = [1 + D^2 + D^3 + D^4 + D^8, 1 + D + D^2 + D^3 + D^5 + D^7 + D^8]$. Following the same steps described above, we can obtain the equivalent generator matrix of $\mathbf{G}(D)$ with rate $4/8$:

$$\mathbf{G}^{[4]}(D) = \begin{bmatrix} p(D) & 1+D^2 & 0 & 1+D & 1 & 1 & 1 & 1+D \\ D & D+D^2 & p(D) & 1+D^2 & 0 & 1+D & 1 & 1 \\ D & D & D & D+D^2 & p(D) & 1+D^2 & 0 & 1+D \\ 0 & D+D^2 & D & D & D & D+D^2 & p(D) & 1+D^2 \end{bmatrix} \tag{8}$$

where $p(D)$=$1 + D + D^2$. One can use any subset of the rows of $\mathbf{G}^{[4]}(D)$ as the generator matrix. We note that the resulting subset will have a smaller rate than the original code $\mathcal{C}$. For example, if we choose only one of the rows of $\mathbf{G}^{[4]}(D)$ as the generator matrix, the resulting code will have a rate of $1/8$. ■

D. Convolutional Code Design for the Randomized Scheme

Earlier in this section, we discussed how a small code and its dual can be used to form the big code. Since both the small code and its dual are assumed to be convolutional codes, the big code is also a convolutional code. Clearly, the minimum pairwise distance among the codewords in each coset with respect to a specific codeword is larger than (or equal to) the minimum distance of the big code with respect to the same codeword. So, the codewords at minimum distance in the big code belong to different cosets and assuming that a minimum distance decoder is being used, they are important sources of decoding errors. Hence, a design metric becomes the minimum pairwise distance among the codewords of the big code which controls the error correction capability of the minimum distance decoder. In practice, one should choose this distance in a way that results in the smallest security gap.

If one uses a convolutional code $\mathcal{C}(n, r)$ (small code) to encode the random bits and its dual $\mathcal{C}^{\perp}(n, n-r)$ to encode the data bits, the big code will consist of all the 2^n n-tuples (ignoring trellis termination to zero state for the time being); a fact that results in the lowest possible minimum distance (one) for the big code. In this case, performance of the minimum distance decoder is poor from the legitimate receiver's point of view. Alternatively, one can use the approach described in the previous subsection to obtain a subset of $\mathcal{C}(n, r)$ denoted by $\mathcal{C}'(n, r')$ where $r' < r$, i.e., using $\mathcal{C}'$ and $\mathcal{C}^{\perp}$ to encode random and data bits, respectively, the big code will have $r' + n - r$ many generators which is less than n; hence, the resulting big code can achieve a larger minimum distance. We note that in either case the data transmission rate is $(n-r)/n$ since the data bits' encoder is the same.

Consider the small code $\mathcal{C}$ to be a convolutional code of rate $R = b/c$ with *minimal-basic* generator matrix $\mathbf{G}(D)$ [13]. Equivalent generator matrices to $\mathbf{G}(D)$ which reproduce $\mathcal{C}$ are obtained by $\mathbf{G}_{2nd}(D) = \mathbf{T}(D)\mathbf{G}(D)$ where $\mathbf{T}(D)$ is a $b \times b$ full rank matrix. Then, instead of working with $\mathbf{G}(D)$, one may use $\mathbf{G}_{2nd}(D)$ in Section III-C to obtain new subsets of $\mathcal{C}$ and consequently new generators for random bits. Different choices for $\mathbf{T}(D)$ result in different generators for random bits. It is clear that different generators for random bits, result in different sets of codewords in each coset and consequently possibly different minimum distances for the big code. In the next example, given the encoder for data bits, we search for an encoder for random bits which results in a big code with a large minimum distance.

Example 2: Let us choose the small code $\mathcal{C}$ as the convolutional code [561 753] which is the same code given earlier in Example 1. Its dual $\mathcal{C}^{\perp}$ is the optimal convolutional code (in terms of minimum distance) of memory 8 and rate $1/2$ with the generator [657 435]. If one uses generators of $\mathcal{C}^{\perp}$ and $\mathcal{C}$ to encode data and random bits, respectively, the resulting big code will have a minimum distance of 2 (they do not cover all the n-tuples because of the trellis termination to zero state). However, if one uses generators of $\mathcal{C}^{\perp}$ for data bits and $[D\ D\ D\ D + D^2\ p(D)\ 1 + D^2\ 0\ 1 + D]$ for random bits which is a subset of $\mathcal{C}$ as derived in Example 1, the big code will attain a minimum distance of 6.

We can improve the minimum distance even more by using $\mathbf{G}^{[4]}_{2nd}(D) = \mathbf{T}(D)\mathbf{G}^{[4]}(D)$ where $\mathbf{G}^{[4]}(D)$ is the same as (8) and the 4×4 matrix $\mathbf{T}(D)$ is given by its polynomial inverse

$$\mathbf{T}^{-1}(D) = \begin{bmatrix} 1+D & D & D & 1+D \\ D & D^2+1 & 1 & D \\ D & D & 1+D & D \\ 1+D & 1 & D & D \end{bmatrix}. \tag{9}$$

After some straightforward algebra, one can calculate $\mathbf{G}^{[4]}_{2nd}(D)$ (which is 4×8) and obtain one of its rows as

$$\begin{aligned} &[D^5 + D^4 + D^3 \quad D^5 + D^3 + D^2 \quad D^4 + D^3 \quad D^5 + D \\ &\quad D^5 + D^4 + D^3 + D^2 + D + 1 \quad D^5 + D^3 + D^2 + D + 1 \\ &\quad D^3 + D^2 \quad D^3 + D^2 + 1]. \end{aligned} \tag{10}$$

Using $\mathcal{C}^{\perp}$ and (10), we obtain a big code with minimum distance 10. Here, it is clear that data bits are encoded with rate $1/2$ while the random bits' encoding rate is $1/8$. We note that the code $\mathcal{C}^{\perp}$ has a minimum distance of 12 which is an upper bound on the minimum distance of the big code. ■

IV. Decoding Methods

A. Optimal Decoder

Given a received noisy vector $\mathbf{y}$, the optimal maximum a posteriori probability (MAP) decoder picks a coset index which maximizes the probability $p(C^i|\mathbf{y})$ where C^i denotes the ith coset. Assuming that there are M cosets which represent M messages and in each of them there are N codewords, the output of the MAP decoder is

$$\hat{i} = \underset{i=1,2,...,M}{\operatorname{argmax}} \; p(C^i|\mathbf{y}) \tag{11}$$

Using Bayes' rule and the total probability theorem (assuming that the codewords in each coset have equal probabilities to be transmitted), we can write

$$p(C^i|\mathbf{y}) = \frac{p(\mathbf{y}|C^i)p(C^i)}{p(\mathbf{y})}, \;\; p(\mathbf{y}|C^i) = \frac{1}{N}\sum_{j=1}^{N} p(\mathbf{y}|\mathbf{c}_{ji}), \tag{12}$$

where $\mathbf{c}_{ji}$ denotes the jth codeword in the ith coset. Finally, for an AWGN channel and equiprobable cosets, the MAP decoder has the form

$$\hat{i} = \underset{i=1,2,...,M}{\operatorname{argmax}} \sum_{j=1}^{N} e^{\frac{-\|\mathbf{y}-\mathbf{c}_{ji}\|^2}{2\sigma^2}}, \tag{13}$$

where $\sigma^2 = N_0/2$. Note that for the main and eavesdropper's channels the noise variances are different, hence the resulting optimal decoding rules are different.

For the case of a binary symmetric channel with cross over probability p

$$p(\mathbf{y}|\mathbf{c}_{ji}) = (1-p)^n \Big(\frac{p}{1-p}\Big)^{d_H(\mathbf{y},\mathbf{c}_{ji})} \tag{14}$$

where $d_H(\mathbf{y},\mathbf{c}_{ji})$ is the Hamming distance between the received vector $\mathbf{y}$ and the codeword $\mathbf{c}_{ji}$. In this case, the optimal decoding rule is obtained from (12) as

$$\hat{i} = \underset{i=1,2,...,M}{\operatorname{argmax}} \sum_{j=1}^{N} \Big(\frac{p}{1-p}\Big)^{d_H(\mathbf{y},\mathbf{c}_{ji})}. \tag{15}$$

We note that for MAP decoding, one goes through all the codewords in all the cosets making the algorithm prohibitively complex to be implemented in practice. However, this process can be used for toy examples with small code lengths. For instance, the performance of the optimal decoder is shown for a Reed-Muller code of length 16 in Fig. 1 for the AWGN channel (along with the performance bounds which will be introduced in Section VI). We emphasize that this is introduced as a toy example only. We will provide examples of good codes with low security gaps in Section VII.

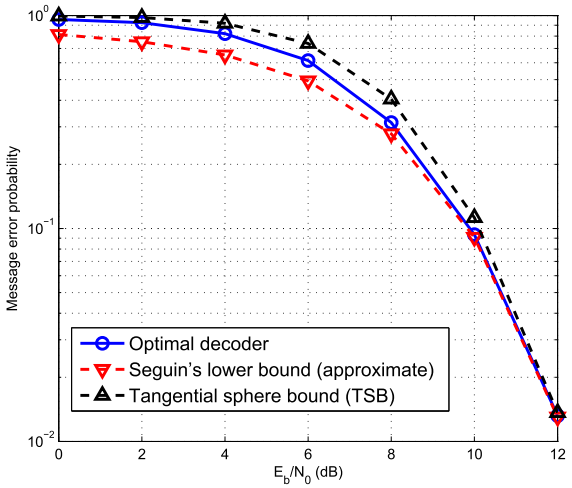

Fig. 1. Performance of the MAP decoder in (13) over an AWGN channel using a Reed-Muller code of length 16 to encode the messages. The number of cosets or messages is 2^5 each of which contains 2^{11} codewords $(n = 16, r = 11, k = 5)$.

B. Sub-Optimal Decoders

Implementation of the optimal decoder for the randomized encoding scheme is formidable in practice, hence here we consider several sub-optimal alternatives.

1) Binary Gaussian Elimination: The encoding scheme in Section III-A can be written in matrix form. Suppose $\mathbf{G}$ is the generator matrix of the small code $\mathcal{C}(n, r)$. We form a matrix $\mathbf{H}$ whose rows are k linearly independent n-tuples $\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_k$ outside $\mathcal{C}$. Therefore, as in [3] one can write the transmitted codeword as follows

$$\mathbf{x} = [\mathbf{s} \;\; \mathbf{v}]\mathbf{G}_B, \qquad \mathbf{G}_B = \begin{bmatrix} \mathbf{H} \\ \mathbf{G} \end{bmatrix}. \tag{16}$$

Motivated by this, a decoding approach becomes performing hard decisions on the received vector to obtain a binary vector which will be denoted by $\hat{\mathbf{x}}$, forming $[\mathbf{G}_B|\hat{\mathbf{x}}^T]$, and through binary Gaussian elimination obtaining $[\mathbf{I}|\mathbf{x}_d^T]$ where $\mathbf{I}$ is the identity matrix. The first k bits of $\mathbf{x}_d$ are the decoded versions of the message $\mathbf{s}$.

This decoding method ignores the available soft information and may not result in a good performance, however it is a general method, i.e., given the generator matrices for the random and data bits ($\mathbf{G}$ and $\mathbf{H}$), it can be applied to any kind of codes. Specifically, low density generator matrix (LDGM) codes introduced in [14] are systematic codes with generator matrices of the form $\mathbf{G} = [\mathbf{I}_{k\times k}|\mathbf{P}_{k\times(n-k)}]$ where $\mathbf{P}$ is a sparse matrix. Hence, given one of $\mathbf{G}$ or $\mathbf{H}$, the other can be obtained, and the binary Gaussian elimination can be used for the present setup with ease.

2) Trellis Based Decoding: When the Euclidean distances among the codewords in each coset are relatively large or when the SNR is sufficiently high, the summations (13) and (15) are dominated by terms which correspond to codewords at the minimum Euclidean distance to the received vector $\mathbf{y}$. Therefore, as an approximate decoding approach, one can find the codeword at the minimum Euclidean (or, Hamming) distance to the given received noisy vector (referred to as the minimum distance decoder). Since at high SNRs, most

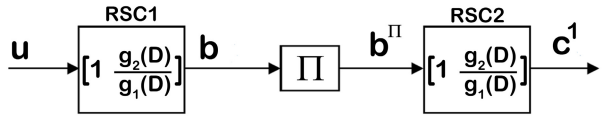

Fig. 2. The encoder for the SCCC.

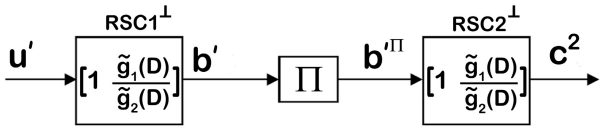

Fig. 3. The encoder for dual of the SCCC in Fig. 2.

errors will be due to closeby codewords, we expect that the performance of this decoder will be close to that of the optimal decoder in this regime.

Following with the development in Section III, we recall that the encoding process needs two convolutional codes whose trellises can be combined to form a trellis for the big code governing codewords obtained by (1), i.e., the codewords that are being sent through the channel. This "big" trellis enables us to find the minimum distance codeword to the output of the channel $\mathbf{y}$ by applying the Viterbi algorithm.

V. Randomized Serially Concatenated Convolutional Codes (RSCCCs)

In this section, we construct a new class of codes, namely, randomized serially concatenated convolutional codes (RSCCCs), by utilizing the results of Sections III and IV-B.2. We note that SCCCs exhibit very sharp slopes in their BER performances [15], and therefore, they are potential candidates to achieve small security gaps.

A. Encoding

Figure 2 depicts an SCCC which consists of two recursive systematic convolutional (RSC) codes. Here, the outer RSC code (abbreviated as RSC1) encodes the information sequence, i.e., u_k's where $1 \leq k \leq K/2$. The resulting codeword is permuted, and then it is fed to the inner RSC code (RSC2) to generate the final codeword.

One needs to obtain the dual of the code in Fig. 2 in order to adapt it for the randomized scheme. This can be accomplished by replacing each RSC code with its corresponding dual as illustrated in Fig. 3. We note that using Theorem 2, if $G(D) = [1 \ \ \frac{g_2(D)}{g_1(D)}]$, then $G^{\perp}(D) = [1 \ \ \frac{\tilde{g}_1(D)}{\tilde{g}_2(D)}]$. Therefore, we are able to use one of the encoders in Figs. 2 and 3 to encode the random bits and the other for the data bits. Assuming $\mathbf{c}^1 = [v_1, q_1, v_2, q_2, \ldots, v_K, q_K]$ and $\mathbf{c}^2 = [v'_1, q'_1, v'_2, q'_2, \ldots, v'_K, q'_K]$, the transmitted codeword is the modulo-2 sum of these two codewords, i.e., $\mathbf{c} = \mathbf{c}^1 + \mathbf{c}^2 = [v_1 + v'_1, q_1 + q'_1, v_2 + v'_2, q_2 + q'_2, \ldots, v_K + v'_K, q_K + q'_K]$.

B. Decoding

The optimal MAP decoding rule for RSCCCs is the same as (13) which is not practical. In this section, we propose a sub-optimal decoder which jointly decodes the random and data bits (i.e. u_i's and u'_i's) by generalizing the decoder introduced in [15] for SCCCs. For this purpose, for the component convolutional codes, the MAP decoding rule is given by

$$(\hat{u}_l, \hat{u}'_l) = \underset{(u_l, u'_l)}{\operatorname{argmax}} \ P\big((u_l, u'_l)|\mathbf{y}\big) \tag{17}$$

where $\mathbf{y}$ is the received signal and $(u_l, u'_l) \in \{00, 01, 10, 11\}$. Joint probabilities $P\big((u_l, u'_l)|\mathbf{y}\big)$ are computed using

$$P\big((u_l = k, u'_l = j)|\mathbf{y}\big) = \sum_{\mathcal{U}_{kj}} p(s_{l-1} = s', s_l = s, \mathbf{y}) \tag{18}$$

where $(kj) \in \{00, 01, 10, 11\}$ and $\mathcal{U}_{kj}$ is set of pairs (s', s) for the state transitions $(s_{l-1} = s') \rightarrow (s_l = s)$ whose corresponding input labels are kj. Using the BCJR algorithm [15], such probabilities are computed efficiently.

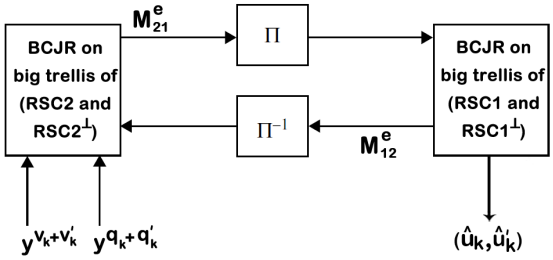

Fig. 4. Iterative decoder for the randomized encoding scheme where one of the encoders in Figs. 2 and 3 encodes the random bits and the other encodes the data bits.

Fig. 4 illustrates the iterative decoder for RSCCCs. The constituent decoders utilize the big trellis introduced in Section IV-B.2, and they exchange information on the pair of bits (b_i, b'_i) introduced in Figs. 2 and 3. Specifically, M_{12} and M_{21} are of the form $\big[\log(p^e_{00}) \ \log(p^e_{01}) \ \log(p^e_{10}) \ \log(p^e_{11})\big]$ where p^e_{kj} denotes the extrinsic probability that $(b_i, b'_i) = (k, j)$. We refer the readers to [16] for details.

VI. Performance Bounds

In order to provide a theoretical assessment of the decoder performance in the randomized encoding scheme, we establish bounds on the resulting error rates. Specifically, we obtain lower and upper bounds on the error rates which indicate the best performance of the eavesdropper and the worst performance of the legitimate receiver, respectively, which are important from a design and analysis point of view.

A. Assumptions

As mentioned in Section III-A, the adopted randomized scheme maps each message to a coset of codewords. Hence, in contrast to conventional encoding, the decision region for each message is not just a simple Voronoi region around the transmitted codeword. This fact results in further complications in calculating the corresponding ML decoding bounds. To proceed, we define the notion of *favorable* codewords.

Definition 3: Suppose $\mathbf{c}_{ij}$ which is the ith codeword in the jth coset is sent through the channel. We call all the other codewords in the jth coset *favorable* to $\mathbf{c}_{ij}$.

Known bounds on the ML decoding performance of linear codes can be applied to the randomized encoding scheme by

making the following assumption: considering transmission of $\mathbf{c}_{ij}$, we ignore all the favorable codewords to $\mathbf{c}_{ij}$, i.e., neglect part of the correct decision region, and compute lower and upper bounds on the performance of decoders in the randomized encoding scheme accordingly.

Theorem 3: Let $\mathbf{c}_{ij}$ be the ith codeword in the jth coset and denote the distance spectrum of the code $\mathcal{C}$ with respect to $\mathbf{c}_{ij}$ by $DS^{C}\{\mathbf{c}_{ij}\}$, and the distance spectrum of the big code (bc) after ignoring the favorable codewords with respect to $\mathbf{c}_{ij}$ by $DS\{\mathbf{c}_{ij}\}$. Then $DS\{\mathbf{c}_{ij}\} = DS\{\mathbf{c}_{lk}\}$, $i \neq l$ and $j \neq k$, if the big code (bc) and the small code (sc) are both linear.

Proof: The distance spectrum of the big code with respect to $\mathbf{c}_{ij}$ after ignoring the favorable codewords can be written as $DS\{\mathbf{c}_{ij}\} = DS^{bc}\{\mathbf{c}_{ij}\} - DS^{coset\ j}\{\mathbf{c}_{ij}\}$, i.e., for each distance $d \geq 1$ subtract the numbers of codewords with distance d in $DS^{bc}\{\mathbf{c}_{ij}\}$ and $DS^{coset\ j}\{\mathbf{c}_{ij}\}$ from each other. Since the big code is linear $DS^{bc}\{\mathbf{c}_{ij}\} = DS^{bc}\{\mathbf{c}_{lk}\}$. The linearity of the small code results in $DS^{sc}\{\mathbf{c}_{11}\} = DS^{sc}\{\mathbf{c}_{i1}\}$. Furthermore, coset j obtained by adding a unique codeword to the small code which does not have any effect on the distance spectrum, namely, $DS^{coset\ j}\{\mathbf{c}_{ij}\} = DS^{coset\ k}\{\mathbf{c}_{lk}\} = DS^{sc}\{\mathbf{c}_{11}\}$, hence $DS\{\mathbf{c}_{ij}\} = DS\{\mathbf{c}_{lk}\}$ concludes the proof. ∎

By using Theorem 3, it is possible to compute the distance spectrum of the big code after ignoring the favorable codewords by considering only the all-zero codeword as the transmitted codeword, via the distance spectra of the small and big codes. Once the distance spectrum is computed, we utilize it with the existing bounds on ML decoding performance to obtain performance bounds for the randomized encoding scheme. If both the small and big codes are convolutional, their distance spectra can be obtained through efficient algorithms (e.g., [10]) based on their state transition matrices computed using their trellis representations.

We note that the derived bounds are applicable to other randomized coding setups as well (once the appropriate weight distributions are known). For instance, one can easily obtain lower bounds on the error rates of LDPC coded systems (e.g., as in [3]) in a straightforward manner as only a subset of codewords with small weights are needed in the computation.

B. Performance Lower Bounds

We first note that the assumption made in Section VI-A, namely, ignoring part of the correct decision region, results in approximate lower bounds. However, since the distance of a codeword to its favorable codewords is typically much larger than its distance to the other codewords (see Section III-D), the ignored correct decision region would have a negligible impact on the final result.

We use Seguin's bound [11] to provide a lower bound on the decoder performance which states that the probability of error given that the signal $\mathbf{s}_u$ is transmitted through an AWGN channel with variance $N_0/2$, denoted by $P(\varepsilon|\mathbf{s}_u)$, is lower bounded as

$$P(\varepsilon|\mathbf{s}_u) \geq \sum_{i \neq u} \frac{Q^2(\sqrt{2D_{ui}E_s/N_0})}{\sum_{j \neq u} \Psi(\rho_{ij}, \sqrt{2D_{ui}E_s/N_0}, \sqrt{2D_{uj}E_s/N_0})} \tag{19}$$

where D_{ui} is the Hamming distance between codewords u and i, E_s/N_0 is the SNR, $Q(\cdot)$ is the right tail probability of standard Gaussian distribution, and

$$\Psi(\rho, p_1, p_2) = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{p_1}^{\infty}\int_{p_2}^{\infty} exp(-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)})dxdy \tag{20}$$

with ρ_{ij} defined as

$$\rho_{ij} = \frac{w((\mathbf{c}_i + \mathbf{c}_u)(\mathbf{c}_j + \mathbf{c}_u))}{\sqrt{w(\mathbf{c}_i + \mathbf{c}_u)w(\mathbf{c}_j + \mathbf{c}_u)}} \tag{21}$$

being the correlation between two codewords $\mathbf{c}_i$ and $\mathbf{c}_j$ given that $\mathbf{c}_u$ is transmitted. w denotes the Hamming weight of a sequence.

It is clear from (19) that one can obtain a lower bound by taking only a subset of codewords into account; in other words, one does not need the entire distance spectrum to obtain a lower bound. Besides, as noted in [18], the codewords at the minimum distance and the corresponding ρ_{ij}'s play an important role on the tightness of this bound. Finally, for the case of a BSC, we use the lower bound proposed in [19].

C. Performance Upper Bounds

Similar to the lower bound, we ignore the favorable codewords for obtaining an upper bound on the error rates of the randomized encoding scheme. The resulting bound in this case is a true bound (not an approximation) on the performance of the maximum likelihood decoder since we ignore part of the correct decision region.

There are many upper bounds on the ML decoding performance of coded systems in the literature; to name two important ones, we cite the Duman-Salehi bound [20] and the tangential sphere bound (TSB) [12]. For a detailed review of the ML performance bounds, see [21]. Here, we adapt a version of the bound in [12] given by

$$P(\varepsilon) \leq \int_{-\infty}^{\infty} \frac{e^{-\frac{z_1^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \Bigg\{ \sum_{k \leq \frac{nr_0^2}{n+r_0^2}} \Bigg\{ S_k \int_{\beta_k(z_1)}^{r_{z_1}} \frac{e^{-\frac{z_2^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \int_0^{r_{z_1}^2 - z_2^2} f_V(v)dvdz_2 \Bigg\} + 1 - \gamma\left(\frac{n-1}{2}, \frac{r_{z_1}^2}{2\sigma^2}\right) \Bigg\} dz_1 \tag{22}$$

where S_k is the number of codewords with Hamming weight k, $\beta_k(z_1) = (\sqrt{n} - z_1)/(\sqrt{n/k - 1})$, $r_{z_1} = r_0(\sqrt{n} - z_1)$, r_0 is the optimal value of r_{z_1} computed in [12] and

$$f_V(v) = \frac{v^{\frac{n-4}{2}} e^{-\frac{v}{2\sigma^2}}}{2^{\frac{n-2}{2}} \sigma^{n-2}\ \Gamma(\frac{n-2}{2})}, \quad v \geq 0,$$
$$\gamma(a, x) = \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t} dt, \quad a > 0, x \geq 0. \tag{23}$$

For the case of a BSC, we use what is called the S bound (SB) given in [12]

$$P(\varepsilon) \leq \sum_{w=d}^{2(m_0-1)} S_w \sum_{\eta=t_w}^{m_0-1} \binom{w}{\eta} p^{\eta}(1-p)^{w-\eta} \sum_{k=0}^{m_0-\eta-1} \binom{n-w}{k} p^{k} (1-p)^{n-w-k} + \sum_{t=m_0}^{n} \binom{n}{l} p^{l}(1-p)^{n-l} \tag{24}$$

where $t_w = \lceil w/2 \rceil$ and m_0 is the smallest integer such that

$$\sum_{w=d}^{2m} S_w \sum_{\eta=t_w}^{m} \binom{w}{\eta}\binom{n-w}{m-\eta} \geq \binom{n}{m}. \tag{25}$$

D. A Simple Example

As an example the performance of lower and upper bounds introduced in Sections VI-B and VI-C for a Reed-Muller code is shown in Fig. 1 which indicates a good match between the bounds and simulated performance of the optimal decoders. We will provide further examples for more practical codes (considering both AWGN and binary symmetric channels) with competitive security gaps in the next Section.

VII. Numerical Examples

In this section, we provide numerical examples on the performance of the sub-optimal decoders introduced in Sections IV-B.1, IV-B.2 and V-B, the theoretical bounds introduced in Section VI and different code designs for cases with noiseless or noisy main channels. For all the examples, n denotes the length of the codewords in the big code, k is the number of data bits and r is the number of random bits.

A. Noiseless Main Channel

We first assume that the main channel is noiseless and the eavesdropper's channel is an AWGN or binary symmetric channel. As discussed in Section III-D, for this scenario, the only task is to confuse the eavesdropper without worrying about the decoding process at the legitimate receiver. This can be accomplished by constructing a big code which consists of all the n-tuples by using all the generators of the small code $\mathcal{C}(n, r)$ to encode the random bits and generators of its dual $\mathcal{C}^{\perp}(n, n-r)$ for the data bits (see section III-D).

Fig. 5 illustrates the message error rates for the randomized encoding scheme using terminated convolutional codes of memory 2. Comparing the two sub-optimal decoders introduced in Section IV-B, we observe that the performance of the trellis based decoder is always better than that of the binary Gaussian elimination decoder. Also TSB and Seguin's bounds are quite tight and all the curves meet at high SNRs as expected. We note that the Seguin's bound is obtained by considering the codewords at the minimum Hamming distance and calculating the correlations among them, while the TSB uses the entire distance spectrum. Furthermore, Fig. 6 shows the performance of the bounds and the minimum distance decoder over a BSC where the lengths of the data and random bit sequences are both 50.

As a second example, we construct the randomized encoding scheme using terminated convolutional codes of memory 6.

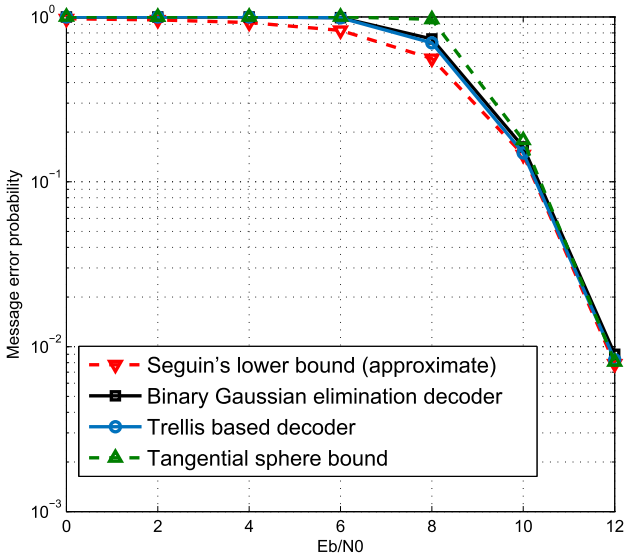

Fig. 5. Performance of the sub-optimal decoders introduced in Section IV-B and the bounds in Section VI when a [7 5] convolutional code with its dual [5 7] is used with $n = 204$ and $k = r = 100$.

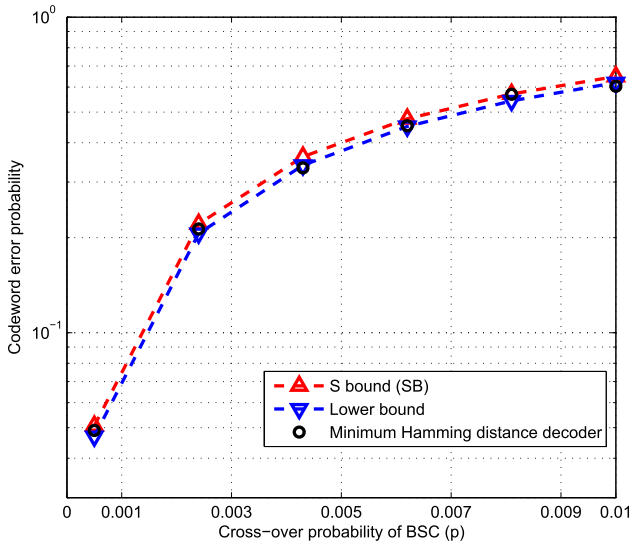

Fig. 6. Performance of the minimum distance decoder and the bounds introduced in Section VI over a BSC with a [7 5] convolutional code and its dual. $n = 104$ and $k = r = 50$.

As shown in Fig. 7, the performance of binary Gaussian elimination is almost the same as the one in Fig. 5, however, performance of the trellis based decoder is improved substantially due to the increase in the minimum distance of the big code.

Seguin's bound relies on low weight codewords to provide tight lower bounds on the performance of the ML decoders [18]. We note that the lower bound in Fig. 7 is not as tight as the one in Fig. 5 because the minimum distance of the big code in Fig. 7 is 2 while it is 1 in Fig. 5. Finally, TSB is not included in Fig. 7 because the state transition matrix of the big code is $(64 \times 64) \times (64 \times 64)$, and calculating the entire distance spectrum which is required for the TSB is not computationally feasible.

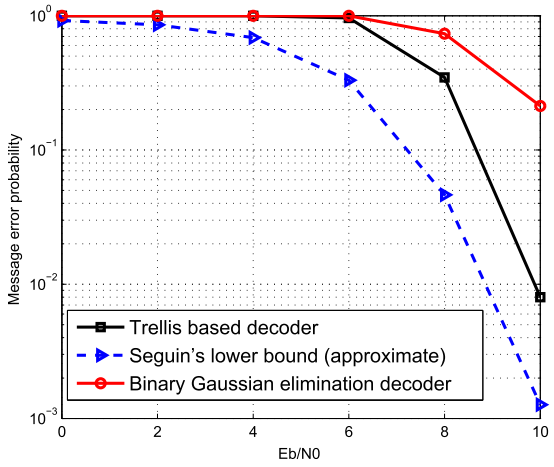

Fig. 7. Performance of the sub-optimal decoders using convolutional codes [117 155] and [133 171] with memory $m = 6$. $n = 212$ and $k = r = 100$.

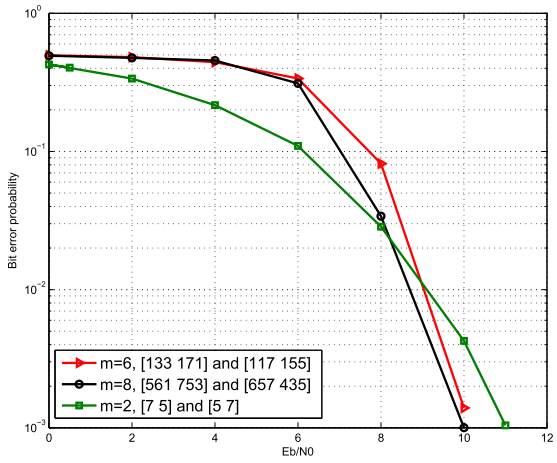

Fig. 8. Bit error probability for 3 convolutional codes with different memory sizes (m). $k = 100$, $r = 100$ and $n = 200 + 2m$. Trellis termination is used.

Fig. 8 shows the bit error rate results when the trellis based (minimum distance) decoder is used for randomized convolutional codes of different lengths. We emphasize that the performance of memory 6 and 8 codes is better than that of memory 2 because the minimum distance of the big code for these two cases is 2 while it is only 1 for the for the latter.

B. Noisy Main Channel

We now assume that both the main and eavesdropper's channels are noisy. Therefore, the generators for random and data bits should be selected in a way that results in low security gaps. We consider AWGN channels. In Section III-D, we have described how the number of codewords in each coset in the randomized encoding scheme can be reduced to obtain a big code with larger minimum distances to improve the decoding performance.

To evaluate the performance of the proposed randomized convolutional coding solution, we show the BER at the eavesdropper (P_{eve}^{min}) as a function of the security gap in Fig. 9 where the convolutional code [657 435] of rate $1/2$ is used for

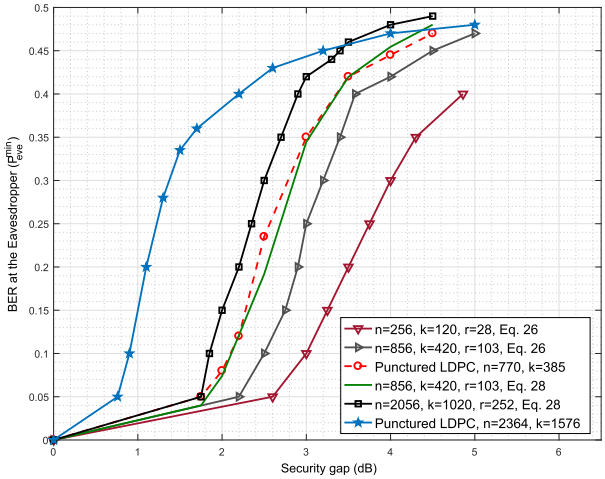

Fig. 9. Bit error probability of the eavesdropper versus the security gap (for $P_{main}^{max} = 10^{-5}$) when the convolutional code [657 435] encodes the data bits for 3 different codeword lengths and two different random bit encoders in (26) and (28). Results corresponding to the optimized punctured LDPC codes reported in [8] and [9] are also included for comparison.

data bits and a subset of its dual for random bits. Specifically, we use the following generator with rate $1/8$ and memory 4 to encode the random bits

$$\begin{bmatrix} D^3+1 & D^4+1 & D^4+D^3+D^2 & D^4+D^3+D+1 \\ D^3+D^2+D & D^3 & D^3+D^2 & D^3+D^2+1 \end{bmatrix} \quad (26)$$

which is obtained using the ideas in Section III-C with

$$\mathbf{T}^{-1}(D) = \begin{bmatrix} 1 & D^2 & 1 & 1 \\ D & 1 & 1 & 1 \\ 1 & 1 & 1 & D \\ 1 & 1 & D & 1 \end{bmatrix}. \quad (27)$$

We note that the resulting big code has a minimum distance of 8. One can increase this distance to 10 by using the following encoder of rate $1/8$ and memory 5 for the random bits (which is another subset of the dual of [657 435])

$$\begin{bmatrix} D^4+D^3+1 & D^3+1 & D^5+D^4+D^2+D+1 & D^4 \\ D^5+D^4+D^3+D^2 & D^4+D^2+1 \\ D^5+D^4+D^3+D^2+D+1 \\ D^5+D^4+D^2+D \end{bmatrix}. \quad (28)$$

Fig. 9 demonstrates that for high BER values at the eavesdropper ($P_{eve}^{min} > 0.45$), the proposed randomized convolutional codes result in lower security gaps compared to the punctured LDPC codes. On the other hand, punctured LDPC codes outperform the randomized convolutional codes for lower P_{eve}^{min} values.

Figs. 10 and 11 demonstrate the performance of RSCCCs and the scrambling approach reported in [8] for physical layer security. The component code for RSCCCs in all the cases is [1 7/5] whose dual is obtained as [1 5/7], and the number of iterations is set to 10 for the iterative decoder. We use the S-random interleaver introduced in [22]. Fig. 10 illustrates that increasing the code length is effective for reducing the security

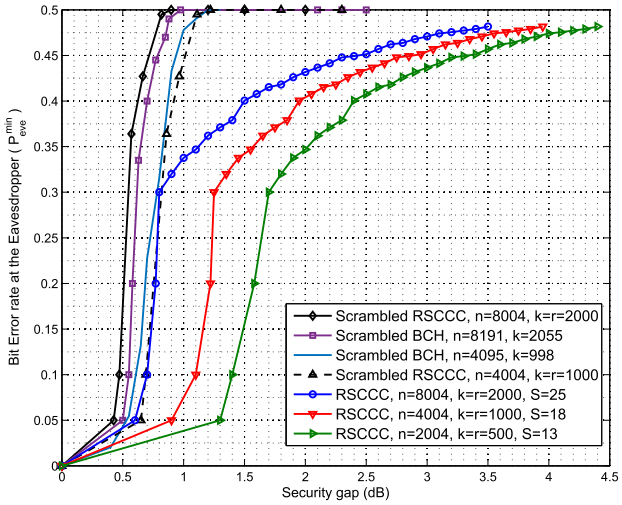

Fig. 10. P_{eve}^{min} versus the security gap (for $P_{main}^{max} = 10^{-5}$) for RSCCCs of different lengths. The results obtained using scrambling [8] are also included for the sake of comparison.

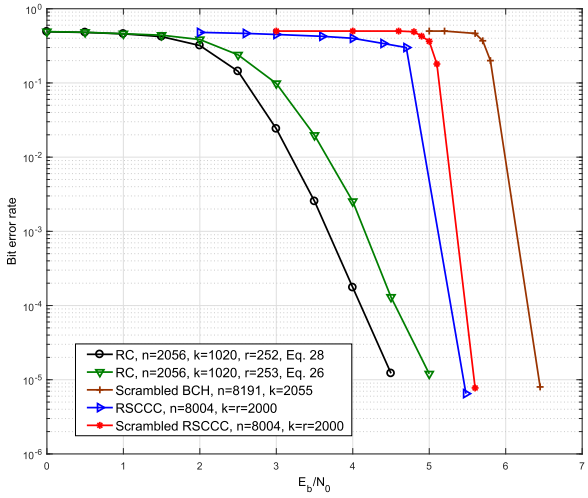

Fig. 11. BER curves corresponding to some of the codes used in Figs. 9 and 10 where RC stands for randomized convolutional coding scheme.

gaps offered by the randomized SCCC scheme. Furthermore, for similar (and large enough) code lengths (around 8000) and similar code rates (close to $1/4$), a scrambled RSCCC results in a 0.1 dB lower security gap compared to a scrambled BCH code (which, to the best of our knowledge, is the best existing scheme in the literature to date as far as the security gap is concerned). Fig. 11 demonstrates the bit error rate curves for some of the schemes presented in Figs. 9 and 10 with the lowest security gaps.

The main idea in scrambling is to multiply a non-singular $k \times k$ binary scrambling matrix $\mathbf{S}$ with the information vector $\mathbf{u}$ of length k before it gets encoded by a linear code. Both the legitimate receiver and the eavesdropper know $\mathbf{S}$ completely and multiply the decoded sequence with $\mathbf{S}^{-1}$ to obtain the message bits. An achievable theoretical security gap for the case where a scrambler is used along with a channel code is computed in [23]. As an example, for $P_{eve}^{min} \approx 0.499$ and for the parameters of the scrambled RSCCC in Fig. 10, the achievable security gap is about 1.24 dB which is close to the corresponding results obtained from the scrambled RSCCC (or scrambled BCH scheme in [8]).[2]

VIII. Conclusions

We propose a randomized coding scheme based on convolutional codes and their duals for the wiretap channel (where a code encodes data bits while its dual encodes a sequence of random bits). We describe the optimal MAP decoder and practically implementable sub-optimal alternatives. In particular, one of the decoders utilizes the trellis of the big code generated by the two terminated convolutional codes, and finds the codeword at the minimum (Euclidean or Hamming) distance to the received noisy vector. We also apply SCCCs to the randomized scheme and describe the corresponding iterative decoder. We devise lower and upper bounds on the error rate performance of the decoders in the proposed setup in terms of the message error probability to analytically characterize the decoder behavior at the eavesdropper and the legitimate receiver, respectively. We illustrate our findings via extensive numerical examples which demonstrate that using scrambling along with the RSCCCs can result in security gaps lower than 1 dB.

References

[1] A. D. Wyner, "The wire-tap channel," *Bell Syst. Tech. J.*, vol. 54, no. 8, pp. 1355–1387, 1975.
[2] L. H. Ozarow and A. D. Wyner, "Wire-tap channel II," *Bell Lab. Tech. J.*, vol. 63, no. 10, pp. 2135–2157, Dec. 1984.
[3] 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.
[4] F. Oggier, P. Solé, and J.-C. Belfiore, "Lattice codes for the wiretap Gaussian channel: Construction and analysis," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5690–5708, Oct. 2016.
[5] 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.
[6] S. Leung-Yan-Cheong and M. Hellman, "The Gaussian wire-tap channel," *IEEE Trans. Inf. Theory*, vol. 24, no. 4, pp. 451–456, Jul. 1978.
[7] 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.
[8] M. Baldi, M. Bianchi, and F. Chiaraluce, "Non-systematic codes for physical layer security," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Dublin, Ireland, Aug. 2010, pp. 1–5.
[9] M. Baldi, F. Bianchi, and F. Chiaraluce, "Coding with scrambling, concatenation, and HARQ for the AWGN wire-tap channel: A security gap analysis," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 883–894, Jun. 2012.
[10] R. McEliece, "How to compute weight enumerators for convolutional codes," in *Communications and Coding*, M. Darnell and B. Honary, Eds. New York, NY, USA: Wiley, 1998, pp. 121–141.
[11] G. E. Seguin, "A lower bound on the error probability for signals in white Gaussian noise," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 3168–3175, Nov. 1998.
[12] G. Poltyrev, "Bounds on the decoding error probability of binary linear codes via their spectra," *IEEE Trans. Inf. Theory*, vol. 40, no. 4, pp. 1284–1292, Jul. 1994.
[13] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. Piscataway, NJ, USA: IEEE Press, 1999.
[14] J. Garcia-Frias and W. Zhong, "Approaching Shannon performance by iterative decoding of linear codes with low-density generator matrix," *IEEE Commun. Lett.*, vol. 7, no. 6, pp. 266–268, Jun. 2003.

[2]Note that this comparison is only approximate since the codes utilized (RSCCC or BCH) do not achieve capacity and the transmission rate to the eavesdropper is not above the channel capacity.

[15] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 909–926, May 1998.
[16] A. Nooraiepour, "Randomized convolutional and concatenated codes for the Gaussian wiretap channel," M.S. thesis, Dept. Elect. Eng., Bilkent Univ., Ankara, Turkey, 2016.
[17] M. Baldi, F. Bambozzi, and F. Chiaraluce, "On a family of circulant matrices for quasi-cyclic low-density generator matrix codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 6052–6067, Sep. 2011.
[18] A. Ozcelikkale and T. M. Duman, "Lower bounds on the error probability of turbo codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, Jul. 2014, pp. 3170–3174.
[19] A. Cohen and N. Merhav, "Lower bounds on the error probability of block codes based on improvements on de Caen's inequality," *IEEE Trans. Inf. Theory*, vol. 50, no. 2, pp. 290–310, Feb. 2004.
[20] T. M. Duman and M. Salehi, "New performance bounds for turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 6, pp. 717–723, Jun. 1998.
[21] I. Sason and S. Shamai (Shitz), "Performance analysis of linear codes under maximum-likelihood decoding: A tutorial," *Found. Trends Commun. Inf. Theory*, vol. 3, pp. 1–222, Jul. 2006.
[22] D. Divsalar and F. Pollara, "Multiple turbo codes for deep-space communications," Telecommun. Data Acquisition Rep., Jet Propulsion Lab., Pasadena, CA, USA, May 1995, pp. 66–77.
[23] I.-M. Kim, B.-H. Kim, and J. K. Ahn, "BER-based physical layer security with finite codelength: Combining strong converse and error amplification," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3844–3857, Sep. 2016.

Alireza Nooraiepour received the B.S. degree in electrical engineering from the Amirkabir University of Technology in 2013 and the M.S. degree in electrical and electronics engineering from Bilkent University in 2016. He has been a Research Assistant with the Communication Theory and Applications Research Laboratory, Bilkent University, since 2014. His current research focuses on wireless communications and coding theory.

Tolga M. Duman (S'95–M'98–SM'03–F'11) received the B.S. degree from Bilkent University, Ankara, Turkey, in 1993, and the M.S. and Ph.D. degrees from Northeastern University, Boston, MA, USA, in 1995 and 1998, respectively, all in electrical engineering. He was with the Electrical Engineering Department, Arizona State University, as an Assistant Professor from 1998 to 2004, an Associate Professor from 2004 to 2008, and a Professor from 2008 to 2015. He is currently a Professor of Electrical and Electronics Engineering Department, Bilkent University, and an Adjunct Professor with the School of ECEE, Arizona State University. His current research interests are in systems, with particular focus on communication and signal processing, including wireless and mobile communications, coding/modulation, coding for wireless communications, data storage systems, and underwater acoustic communications.

Dr. Duman was a recipient of the National Science Foundation CAREER Award and the IEEE Third Millennium Medal. He served as an Editor of the IEEE TRANSACTION ON WIRELESS COMMUNICATIONS from 2003 to 2008, the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS from 2002 to 2007, the IEEE TRANSACTION ON COMMUNICATIONS from 2007 to 2012, and *Physical Communication* (Elsevier) from 2010 to 2016. He has been the Coding and Communication Theory Area Editor of the IEEE TRANSACTION ON COMMUNICATIONS since 2011, an Editor of the IEEE TRANSACTION ON WIRELESS COMMUNICATION since 2016, and the Editor-in-Chief of *Physical Communication* (Elsevier) since 2016.