EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Pre-coded LDPC coding for physical layer security

Kyunghoon Kwon, Taehyun Kim and Jun Heo*

Abstract

This paper examines a simple and practical security preprocessing scheme for the Gaussian wiretap channel. A security gap based error rate is used as a measure of security over the wire-tap channel. In previous works, information puncturing and scrambling schemes based on low-density parity-check (LDPC) codes were employed to reduce the security gap. Unlike the previous works, our goal is to improve security performance by using the precode of the feed-forward (FF) structure. We demonstrate that the FF code has an advantage for the security gap compared to the perfect scrambling scheme. Furthermore, we propose the joint iterative decoding method between LDPC and FF codes to improve the reliability/security performances. The proposed joint iterative method is able to achieve outstanding performance by using the proposed scaling and correction factors based on signal-to-noise ratio (SNR) evolution. The improved performances by these factors are demonstrated through the extrinsic information transfer (EXIT) chart and simulation results. Finally, the simulation results suggest that the proposed coding scheme is more effective than the conventional scrambling scheme.

Keywords: Feed-forward, Pre-code, LDPC code, BCJR algorithm, Physical layer security, Wiretap channel, Scrambling, Security gap, Joint iterative decoding, EXIT chart

1 Introduction

For several decades, wireless communication technologies have been available that exchange information rapidly and reliably between a sender and a receiver. Owing to the continued development of communication technologies, we can today access communication networks conveniently and with transportability, whenever and wherever we wish. In conjunction with this development, a growing interest has developed in secure information transmission over wireless networks related to the specific security vulnerabilities caused by the inherent openness of wireless media. It is difficult to detect eavesdropping because anybody can acquire transmitted information over a wireless communication channel.

Shannon established communication theory in 1949 and defined the basic concept of secure communication from the information-theoretic perspective [1]. Using Shannon's approaches, a sender, Alice, securely transmits an information message M to a legitimate receiver, Bob, across a public channel. To be "perfectly secure", the requirement of the mutual information $I(M;X) = 0$ must be satisfied between Alice's information message M and the transmitted word X. From this definition, Shannon proved that Alice and Bob must share a key string to achieve perfect security. This theory was the introduction of the key distribution problem and is the basis of symmetric key cryptography defense systems for the upper layer implemented today. Present systems based on cryptography prevent the extraction of information without a secure key string when information is exposed to the eavesdropper Eve. This public key algorithm depends on the computational limit of the eavesdropper to ensure computational security. In spite of the improvements in public key algorithms, there remains a problem for security based on the assumption of Eve's limited computational resources considering the advancement of available computing power.

An alternative technology that is not based on computational complexity, is physical layer security. Unlike the key distribution problem, physical layer security utilizes the characteristics of a communication channel and allows

*Correspondence: junheo@korea.ac.kr
The School of Electrical Engineering, Korea University, 5-1 Anam-dong, Sungbuk-gu, 136-713, Seoul, Republic of Korea

a legitimate receiver to decode correctly. The important difference compared to Shannon's theory is that the eavesdropper can observe information transmitted by the sender through another channel. Physical layer security guarantees security analytically, based on information theory, regardless of the eavesdropper's computational power. Therefore, there is no elevation of risk due to the advancement of high speed computing.

A security system based on the physical layer was introduced by Wyner in 1975 [2] and information-theoretically secure communication was studied in [3, 4]. According to the wiretap channel model defined by Wyner, the main channel was defined between the sender, Alice, and the legitimate receiver, Bob; the wiretap channel was defined as a degraded version of the main channel. The main and wiretap channels were assumed to be discrete memoryless channels. Suppose that Alice sends Bob an s-bit message M across the main channel. Alice encodes M into an n-bit transmitted word X. Bob and Eve receive message X across the main and wiretap channel, respectively. Bob and Eve's channel observations are denoted by Y and Z, respectively. Alice encodes the information for two objectives [2] as follows: (i) the error probability between the message M and Bob's decoded message $\hat{M}_B$ of the received message Y must converge to zero (with negligibly small probability of error) [reliability]. ii) no information is shared between information message M and Eve's received message Z. For a precise expression, the formulation is articulated as the rate of mutual information $\frac{1}{n}I(M;Z) \rightarrow 0$ when $n \rightarrow \infty$ [security]. Wyner defined that physical layer security is achieved without key distribution using forward error correction (FEC) when it corresponds to the considerations of reliability and security. Moreover, the secrecy rate is defined by the rate s/n, where s and n are the number of secret message bits and the number of bits transmitted over the channel, respectively. A detailed explanation of Wyner code could be found in [5].

Cheong generalized the Gaussian wiretap channel [6] based on Wyner's wiretap channel model as illustrated in Fig. 1. Wyner showed that if the wiretap channel is a degraded version of the main channel then secrecy capacity is positive. In [4], the authors showed that the secrecy capacity is positive when the main channel is "less noisy" than the wiretap channel such as $\sigma_B^2 \leq \sigma_E^2$ (corollary 3 in [4]). Then, Bob's received signal-to-noise ratio (SNR) $\left(P/\sigma_B^2\right)$ is greater than Eve's SNR $\left(P/\sigma_E^2\right)$.

Several security measurement metrics for physical layer security are used for evaluating transmissions over the wiretap channel. These security metrics depend on the characteristic of the coding scheme used for transmissions. Among the metrics, bit error rate (BER) can be a practical metric as a security measure when modulation and coding schemes (MCS) are considered in a practical system [7, 8]. Therefore, since the BER metric allows for easy measurement and straightforward assessment, in this paper, we focus on the BER security metric. Another useful metric to measure the security is the equivocation rate analysis by information-theoretic security on the secret message [9–11]. The information theoretic approach could be developed, since BER metric could not provide the same amount of information for the information theoretic approach and guarantee perfect secrecy. However, it is out of scope of this paper. The BER of approximately 0.5 of Eve's decoded message $\hat{M}_E$ with random noise does not guarantee that she will not be able to obtain sufficient information on the transmitted message. Security measurement using BER was introduced by Klinc et al. and is called "security gap". Security gap is defined as the difference between Bob and Eve's received SNR and can be used to achieve physical layer security. It is assumed that Bob's received SNR is greater than Eve's. To achieve physical layer security for the same received messages, an average BER over Eve's channel, P_e^E must approach 0.5 and an average BER over Bob's, P_e^B must approach zero. Thus, the reliability and security conditions are as follows:

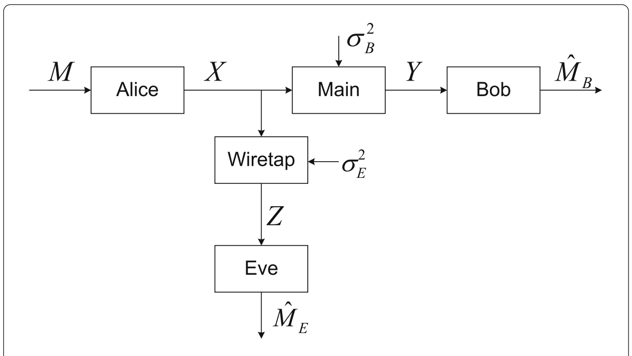

Fig. 1 Block diagram of a Gaussian wiretap channel

$$\text{(a) Reliability} : P_e^B \leq P_{e,max}^B;$$
$$\text{(b) Security} : P_e^E \geq P_{e,min}^E,$$

where $P_{e,max}^B$ and $P_{e,min}^E$ are the BER thresholds for reliability and security, respectively. Bob's near-zero BER implies a negligibly small probability of error in a practical system and Eve's BER around 0.5 implies that half of the information is corrupted by channel noise. Therefore, $P_{e,max}^B$ and $P_{e,min}^E$ as BER thresholds are defined by BER 10^{-5} and 0.4 in this paper. Thus, the security gap can be expressed in terms of the SNR as follows [7]:

$$S_G(security\ gap) = \frac{SNR_{B,min}}{SNR_{E,max}}, \tag{1}$$

where $SNR_{B,min}$ is the lowest SNR for which (a) is satisfied and $SNR_{E,max}$ is the highest SNR for which (b) holds.

According to (1), the security gap should be kept as small as possible, so that the desired security is achieved with small degradation of Eve's channel. Therefore, it is important to construct an error-correcting code (ECC) to reduce the security gap. As mentioned above, the main target of this paper is to keep the security gap as small as possible.

Studies on the error-correcting code for physical layer security have focused on low-density parity-check (LDPC) codes. LDPC codes [12] have a remarkable error-correcting capability and a powerful analysis tool for a belief propagation (BP) decoder, [13] called density evolution (DE) [14] or the extrinsic information transfer (EXIT) chart [15]. Klinc et al. [7] proposed a security-achieving algorithm using LDPC codes with a puncturing scheme. Only parity bits are transmitted to eliminate the exposure of secret messages and the decoders recover the punctured bits using the received parity bits. Baldi proposed non-systematic codes [16, 17] for physical layer security using a scrambling matrix inspired by the McEliece Cryptosystem [18]. This scheme causes intentional bit error propagation where transmitted bits consist of scrambled information bits. This achieves secrecy maintaining the error correction capability of FEC and the advantage of a decrease in the signal power compared with the puncturing scheme [19]. However, since the scrambling scheme produced leads to an error propagation phenomenon, an improved reliability in terms of frame error rate cannot be expected.

In this paper, we propose a feed-forward (FF) pre-code that resolves the disadvantage of the puncturing scheme for linear block codes and addresses the advantage of a decrease in the signal power with respect to the conventional scrambling scheme. Unlike the previous scrambling scheme that uses a hard decision value for error propagation only, the proposed code has an improved reliability at a high SNR region compared to the scrambling scheme. We demonstrate that the proposed code has improved reliability performance at high SNR with a reduced security gap. The proposed system consists of an LDPC code as an inner code and an FF code as a pre-code (outer code). The outer code has a code rate approaching one to minimize the loss of transmitted information against the conventional scrambling scheme. By concatenating LDPC and FF codes, reliability is achieved using LDPC and security is realized using the FF code. Unlike the scrambling scheme, the FF code employs soft decision decoding to recover the secret message and has superior reliability performance compared to the scrambling scheme. The reliability performance can be improved by applying joint iterative decoding to the proposed system. The improved performance is demonstrated through the EXIT chart curves [20–22].

The outline of this paper is as follows. In Section 2, we introduce the wiretap channel model and review previous works, information puncturing, and scrambling schemes. In Section 3, the encoding and decoding procedures of the FF code are discussed and the performance is evaluated. In Section 4, the joint iterative decoding procedure is explained and the security and reliability performances of the proposed system are evaluated. Also, we approximate the factors used in this paper and analyze the performance of the proposed system using the EXIT chart curve. The conclusion is presented in Section 5.

2 Preliminaries and related works

This section discusses some background concepts and the previous works that will be used throughout the paper.

2.1 System model

Alice sends an n-bit transmitted sequence $X^n \in \{x_1, x_2, \cdots, x_n\}$ after encoding a k-bit pre-coded message $M^k \in \{m_1, m_2, \cdots, m_k\}$ (M^k is the pre-coded message of the s-bit secret message $U^s \in \{u_1, \cdots, u_s\}$). The received sequences of Bob and Eve are denoted as Y^n and Z^n, respectively. Alice sends message X using binary phase-shift keying (BPSK) modulation. The Gaussian wiretap channel model can then be generalized [9, 10] as follows:

$$\begin{aligned} Y_i &= X_i + N_i^{Bob} \\ Z_i &= \kappa X_i + N_i^{Eve} \end{aligned} \tag{2}$$

where N_i^{Bob} and N_i^{Eve} are independent and identically distributed (i.i.d) zero-mean Gaussian random variables of variance σ_B^2 and σ_E^2, respectively, and κ is a positive constant that models the gain advantage of the eavesdropper over the destination.

Let n_{ch} be the number of transmitted bits over the channel, and n_{code} denote the codeword block length of the LDPC code. Define the design rate $R_d = \frac{k}{n_{ch}}$, the secret rate $R_s = \frac{s}{n_{ch}}$, and the code rate $R_c = \frac{k}{n_{code}}$. In general, if the number of the secret message bits s is equal to the dimension of the LDPC code k, then $R_s = R_d$. If $R_s < R_d$ in [7], it may help to achieve the reduced security gap but higher power should be needed to achieve the reliability condition. Since the power saving is important in many applications, $R_s \approx R_d$ is preferred.

2.2 Punctured and scrambled code for Gaussian wiretap channel

In [7], D.Klinc et al. proposed punctured LDPC codes to achieve security over the Gaussian wiretap channel. The punctured LDPC codes are employed to remove the exposure of the secret message to Eve. The puncturing fraction is denoted by p, which implies the fraction of the punctured secret message. To construct the $R_s = R_d$ code, the mother code with rate $R_c = p < 0.5$ must be used, since the secret rate $R_s = p/(1-p)$. The authors of [7, 8] demonstrated that the punctured code can remarkably reduce the security gap compared with the non-punctured code. However, the punctured code has less reliable performance than the non-punctured code and requires higher power to achieve good performance over the main channel. To overcome these vulnerabilities, non-systematic codes using scrambling schemes were proposed by Baldi et al. [16, 17]. In the scrambling scheme, Alice generates the pre-coded message m by multiplying the secret message vector u and scrambling matrix S. Alice then sends the encoded message x by a product of the pre-coded message $m = u \cdot S$ and the generator matrix G to Bob. The scrambling procedure transforms the systematic code to the non-systematic code. Unlike the previous puncturing scheme, the scrambling scheme maintains that the secret and code rates are equal, that is $R_s = R_c$, and the scheme requires the same signal power to achieve reliability. The expression of scrambling can be written as

$$x = u \cdot S \cdot G = m \cdot G.$$

A $1 \times n$ pre-coded codeword x is generated by multiplying a $k \times n$ generator matrix G and $1 \times k$ pre-coded message m constructed by multiplying a $1 \times k$ secret message u and a $k \times k$ scrambling matrix S. Figure 2 illustrates a simple example of the puncturing and scrambling schemes. The received signal is first decoded using the channel decoder. The decoded message $\hat{u}$ is solved through multiplication by the inverse scrambling (descrambling) matrix S^{-1} and the decoded message m, and the expression of descrambling can be written as

$$\hat{u} = (m + e) \cdot S^{-1} = u \cdot S \cdot S^{-1} + e \cdot S^{-1} = u + e \cdot S^{-1}$$

It is possible to recover the secret message with correct decoding. However, if decoding fails, an error propagation phenomenon is observed due to the density of the descrambling matrix S^{-1} in the right-side term of the above equation. In [17], perfect scrambling is denoted by a descrambling matrix with row and column weight > 1 and a density close to 0.5. Thus, perfect scrambling with one (or more) error(s) causes an error rate around 0.5 in the final decoded message. Since the BER of Eve is very close to 0.5 (if errors are randomly distributed), it would be difficult to extract much information about the message. In terms of the gain of signal power, Baldi et al. showed that the puncturing scheme has worse error correcting performance than the scrambling scheme with respect to systematic LDPC coding. This is because the puncturing scheme increases the code rate and has a negative impact on the code minimum distance which is reduced [23, 24]. However, the scrambling scheme can only provide an error propagation effect, not error correction. The use of the scrambling scheme without FEC (as unitary rate coding, section 3-A in [17]) guarantees security performance on average, though it does not provide improved reliability.

3 Feed-forward pre-code for physical layer security

To achieve physical layer security with minimum loss of code rate, the difference in the dimension between secret and pre-coded messages must be minimized. This also enables low complexity of the security processing. The block diagram of the entire proposed system with the pre-coded LDPC concatenation is illustrated in Fig. 3. The sender (Alice) encodes the s-bit secret message U using security preprocessing (FF encoder) and then encodes the FF-coded message M into an n-bit codeword X. Bob and Eve receive the message X across the main and wiretap channel, respectively; then, using the received sequence

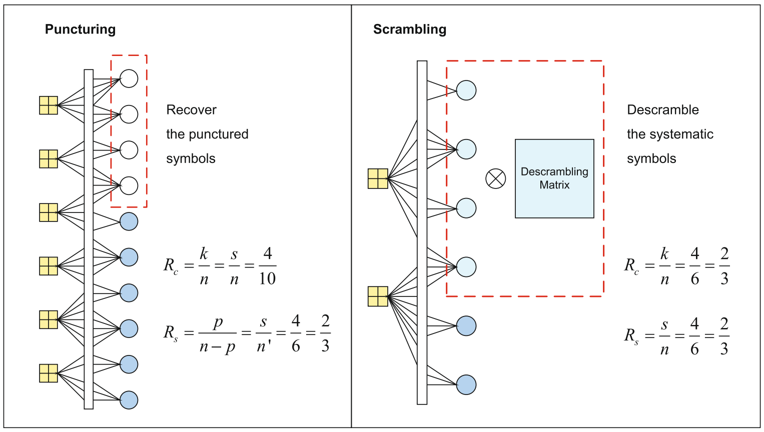
Puncturing
Recover
the punctured
symbols
$R_c = \frac{k}{n} = \frac{s}{n} = \frac{4}{10}$
$R_s = \frac{p}{n-p} = \frac{s}{n'} = \frac{4}{6} = \frac{2}{3}$
Scrambling
Descrambling Matrix
Descramble
the systematic
symbols
$R_c = \frac{k}{n} = \frac{4}{6} = \frac{2}{3}$
$R_s = \frac{s}{n} = \frac{4}{6} = \frac{2}{3}$

Fig. 2 Examples of an information puncturing and scrambling schemes

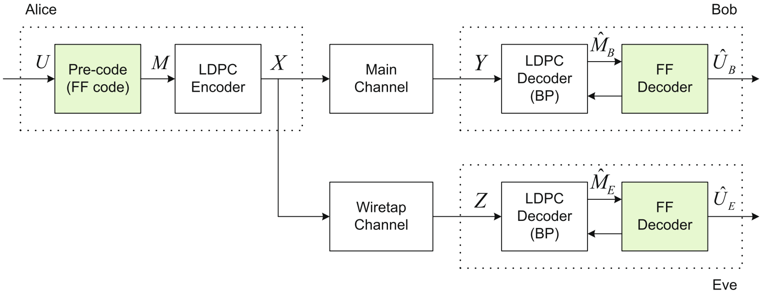

Fig. 3 Block diagram of the proposed system with the pre-coded LDPC concatenation over Gaussian wiretap channel

of Bob "Y" and Eve "Z", the decoded messages $\hat{M}_B$ and $\hat{M}_E$ are achieved by performing their own LDPC decoding procedure, respectively. The secret messages $\hat{U}_B$ and $\hat{U}_E$ can be recovered via the FF decoder into the decoded messages for Bob and Eve, respectively. In our simulations, BPSK modulation $\{+1, -1\}$ is employed and the code rate of LDPC is 1/2. The number of transmitted bits is 960. The FF decoder employs the Bahl-Cocke-Jelinek-Raviv (BCJR) decoding algorithm for soft decision decoding. We employ an LDPC code, as specified in the IEEE 802.16e standard, in the proposed system for the following analysis [25]. For LDPC decoding, the message-passing algorithm in [13] is used. However, in this section, we only provide the encoding and decoding procedures of the FF code as a pre-code and evaluate its reliability and security performances.

The proposed coding scheme employs the simplest convolutional encoding with one tail bit to protect the secret message for an improved reliability performance, and the decoding complexity of the proposed scheme is higher due to soft decision decoding (BCJR algorithm).

3.1 Encoding

Security processing with error propagation must be provided to achieve security. Thus, in this paper, we propose the FF code as a pre-code, which is the inverse form of a differential coding (DC) scheme. The proposed code has low complexity and a feed-forward structure, not a recursive form. Its generator polynomial is $g_{FF}(D) = 1 + D$ with a memory order of 1. Figure 4 presents the block diagram of the FF encoder.

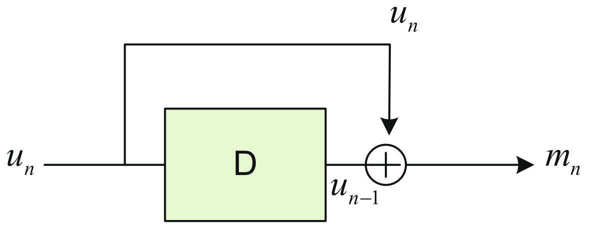

Fig. 4 Block diagram of feed-forward encoder

The FF encoder is a reversed form of the differential encoder, i.e., the FF encoder and differential decoder constructions are the same structure. The matrix equation of the proposed encoder is expressed as follows:

$$G_{FF} = \begin{bmatrix} 1 & 1 & & 0 \\ & 1 & \ddots & \\ & & \ddots & 1 \\ 0 & & & 1 \end{bmatrix}, G_{FF}^{-1} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ & 1 & \cdots & 1 \\ & & \ddots & \vdots \\ 0 & & & 1 \end{bmatrix}, \tag{3}$$

and the pre-coded sequence m_n can be directly expressed as

$$m_n = u_{n-1} \oplus u_n. \tag{4}$$

Unlike the differential encoder, the output message of the FF encoder consists of the modulo-2 addition between the previous input symbol and the present input symbol. The density of the descrambling matrix G_{FF}^{-1} is close to 0.5 due to the full upper triangular matrix. For arbitrary n, the density of G_{FF}^{-1}, D_{FF}, can be written as:

$$D_{FF} = \frac{\sum_{i=1}^{n} i}{n^2} = \frac{n+1}{2n} \tag{5}$$

where n is the length of the secret message. If n approaches infinity,

$$\lim_{n \to \infty} D_{FF} = \lim_{n \to \infty} \frac{n+1}{2n} = 0.5. \tag{6}$$

On the case of binary phase shift keying (BPSK), the bit and frame error probability are given as

$$\begin{cases} P_e = \dfrac{1}{2}\text{erfc}\left(\sqrt{\dfrac{E_b}{N_0}}\right), \\ P_f = 1 - (1 - P_e)^n = 1 - \left(1 - \dfrac{1}{2}\text{erfc}\left(\sqrt{\dfrac{E_b}{N_0}}\right)\right)^n. \end{cases}$$

Therefore, an upper bound (UB) of FF hard decision decoding is guaranteed as

$$P_{e,UB}^{FF} = \left(\frac{n+1}{2n}\right)\left\{1-\left[1-\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)\right]^n\right\} \tag{7}$$

$$\geq \frac{1}{2}\left\{1-\left[1-\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)\right]^n\right\}. \tag{8}$$

The proposed code with density 0.5 guarantees the requirement of perfect scrambling, and achieves the limit of security performance when n goes to infinity. In contrast to the conventional scrambling scheme based on a non-singular random matrix, the FF code consists of the straightforward structures of the encoder and decoder.

From [6], it is easily proved that the bit error probability after FF hard decision decoding approaches half the frame error probability, as in [16, 17]. Let j be the number of errors, P_j be the probability that a received n-bit vector contains j errors before FF hard decision decoding, m_i be the ith error position in an n-bit string which contains j errors, and ξ_j be the number of all possible cases after FF hard decision decoding in the n-bit string which contains j errors. Ω_e denotes the expectation value of the number of errors after FF hard decision decoding. Under such assumptions, the bit error probability after FF hard decision decoding can be expressed as follows:

$$P_e^{FF} = \frac{\Omega_e}{n}, \tag{9}$$

with

$$\begin{cases} P_j = \binom{n}{j} P_e^j (1-P_e)^{n-j} \\ \Omega_e = \sum_{j=1}^{n} \frac{P_j}{\xi_j}\left[\sum_{m_1=1}^{n-j+1}\sum_{m_2=m_1+1}^{n-j+2}\cdots\sum_{m_j=m_{j-1}+1}^{n}\left\{\sum_{l=1}^{j}(n+1-m_l)(-1)^{l-1}\right\}\right]. \end{cases} \tag{10}$$

In Fig. 5, the BER performance of the FF hard decision decoding with the number of transmitted bits $n = 10$ is evaluated by the upper bound, error probability of perfect scrambling, error probability of FF hard decision decoding, and simulation. The upper bound and error probabilities are computed from [7–9]. The simulation results show that the performances of equations [7–9] are very close to the simulation result. From the figure, the performance and the descrambling density of the proposed FF code are close to the conventional scrambling scheme.

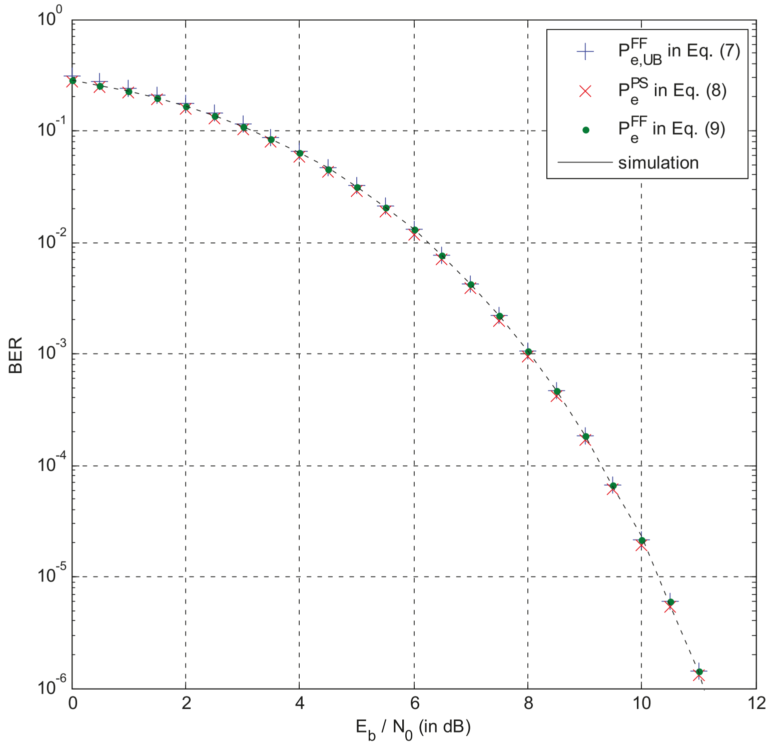

Fig. 5 Upper bound (7), perfect scrambling (8) and the analysis of FF hard decision decoding (9) with $n = 10$ bits

3.2 Decoding

The inverse generator polynomial is $g_{FF}^{-1}(D) = \frac{1}{1+D}$ because the pre-coded message $\hat{M} = (\hat{m}_1, \hat{m}_2, \cdots, \hat{m}_n)$ consists of the generator polynomial $g_{FF}(D) = 1 + D$. The FF decoder is a recursive form of the encoder. Because of this construction, the FF-decoded message $\hat{U} = (\hat{u}_1, \hat{u}_2, \cdots, \hat{u}_n)$ has a regularity as follows:

$$\hat{u}_n = \hat{m}_n \oplus \hat{u}_{n-1}. \tag{11}$$

The recursive form of a decoder can continuously propagate a bit error when an error occurs in the received message. The construction of the FF code is based on the convolutional code. Thus, the FF code can be expressed using a trellis diagram. The FF code can be decoded using a soft-input soft-output (SISO) decoder or symbol-by-symbol maximum a posteriori (MAP) algorithm. The representative MAP decoding algorithm is the BCJR algorithm [26] used in classical turbo decoding. By applying the symbol detection of the BCJR algorithm using soft decision, the performance loss of the sequence detection from hard decision can be reduced. The trellis diagram of the FF code is presented in Fig. 6.

Figure 6 describes the nth FF-decoded message $\hat{u}_n$ value 0 (1) as a solid (dotted) line. When the decoding is performed, the FF-decoded bit is correlated with all of the incoming bits. It has a coding gain in the high SNR region owing to the correlation property. Figure 7 presents the BER and frame error rate (FER) of the proposed scheme compared to the conventional scrambling scheme.

While the scrambling scheme only has error propagation capability, the proposed FF code, with increased minimum Hamming distance ($d_{min} = 2$) using redundant bit (tail bit) and coding gain using the BCJR algorithm, has a noticeable performance gain in the high SNR region. In the low SNR region, this code demonstrates a BER of 0.5. Security as defined in this paper is achieved. Moreover, this code has an improved performance of about 0.4 dB compared to the uncoded system at the BER of 10^{-7}, owing to the BCJR decoding algorithm. Compared with the conventional scrambling scheme, the proposed code has a performance improvement of approximately 1.4 dB at the BER of 10^{-7}.

If information from other symbols with low reliability is incorrect, errors accumulate for the entire code sequence, which cause error propagation. Unlike channel errors, the error positions after FF decoding (or descrambling) are not exactly i.i.d. Moreover, the operation of the FF code employs the correlation effect between consecutive symbols and each symbol is dependent on other symbols. Therefore, we cannot state that this system has a perfect secrecy even though Eve's BER is equal to 0.5. This does not ensure the maximum entropy for Eve, since the error positions are not i.i.d.

The security performance using security gap is presented in Fig. 8 and Table 1, where the number of transmitted bits is 480, and Bob's maximum BER, $P_{e,max}^{B}$, is 10^{-5}. From the figure, we can observe that Eve's BER converges very slowly toward the ideal value of 0.5; hereafter, $P_{e,min}^{E} \geq 0.4$. Moreover, the security gap performances at $P_{e,min}^{E} \geq 0.48$ are almost the same. We will refer to "$P_{e,min}^{E} \geq 0.4$" as a sufficient amount of physical layer security in this paper, but our schemes still apply to stricter security thresholds ($P_{e,min}^{E} = 0.5$). Consider that when the Eve's minimum BER is $P_{e,min}^{E} = 0.4$, the uncoded scheme (only BPSK $\{+1, -1\}$) requires a large (>20 dB) security gap to achieve security performance. In the case of the scrambling scheme, to achieve $P_{e,min}^{E} = 0.4$, only a 6.29 dB security gap is required. However, the proposed FF code, unlike in the scrambling scheme, yields a security gap gain of approximately 0.74 dB at $P_{e,min}^{E} = 0.4$ compared to perfect scrambling. A security gap of only a 5.55 dB is required to achieve $P_{e,min}^{E} = 0.4$.

3.3 Complexity

One way to compare the complexity of the perfect scrambling and the pre-code (FF hard and soft decoding) is to compare the type of operations and count the number of times each operation is performed. The BCJR algorithm of the pre-code involves the following operations:

- Forward/backward recursion: let t be the number of states of the FF code, n be the number of the length of a trellis, respectively. From the Fig. 7, each state has two outgoing branches. For each state, $(2t)$ multiplication operations and t addition operation are needed. Therefore, for a trellis with length n, a

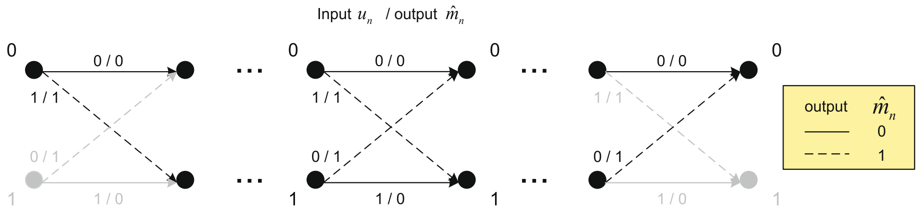

Fig. 6 Trellis diagram corresponding to FF code with generator polynomial $g_{FF}(D) = 1 + D$

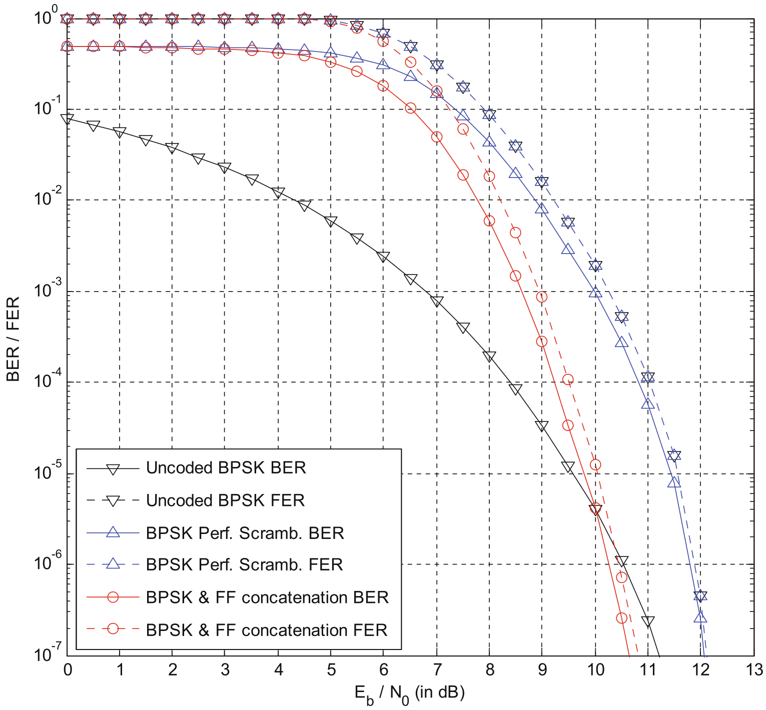

Fig. 7 BER and FER performance without forward error correction ($s = 479$ bits, tail bit 1, and $k = n = 480$ bits), in the presence of BPSK modulation, perfect scrambling, and FF code

total of $(2tn)$ multiplication operations and (tn) addition operations are required. Likewise, the operations required to backward recursion are also equal to forward recursion.

- Branch metric (probability): to compute the branch metric on the probability domain, $(2t)$ branch metrics are needed since there are t states and each state has two outgoing branches. For each branch, two multiplications are required. Therefore, a total of $(4tn)$ multiplications are needed for a trellis length n.
- LLR computation: the numerator (denominator) of LLR computation is the total sum of the probability of branch metric corresponding to 0 (1). Since the pre-code has two states and two outgoing branches per each state, there are four branch metrics of probability domain. Among the metrics, two branch metrics are corresponded to the probability of 0. For each numerator and denominator, $(t - 1)$ addition operations are needed. Then, 1 logarithm operation and 1 division operation are needed to compute LLR. In total, $2(t - 1)n$ addition, n logarithm, and n division operations are needed.

To compute the perfect scrambling scheme (randomly generated), $1 \times n$ hard decision vector and $n \times n$ descrambling matrix are needed. For the 1st decoded (descrambled) bit, n multiplication operations and $n - 1$ addition operations are needed. In total, n^2 multiplication and $n(n - 1)$ addition operations are needed to obtain the descrambled message.

The computational complexity could be decreased by using G_{FF}^{-1} as perfect scrambling matrix (FF hard decoding). In the previous section, we provide that the matrix G_{FF}^{-1} guarantees the consideration of perfect scrambling. From the Eq. (11), the sequence detection can be used. Then, in total, only $n - 1$ addition operations are needed to compute the descrambled message. The type of operations required by these algorithms (randomly generated perfect scrambling, FF hard, soft decoding) and the number of times each operation is executed are summarized in the Table 2.

From Table 2, it is possible to incorrectly evaluate that the perfect scrambling scheme (random matrix) has more complexity than the FF soft decoding, since it only provides the types and numbers of operations for real value computation. In terms of the hardware implementation, the perfect scrambling only uses binary operations (modulo-2 operations); however, BCJR algorithm of FF soft decoding requires the operations of the real values and it needs more cost per one operation than the perfect scrambling. For those reasons, it is difficult to precisely compare the algorithms with the data in Table 2. Therefore, the matrix G_{FF}^{-1} is used as perfect scrambling for a fair comparison in this paper.

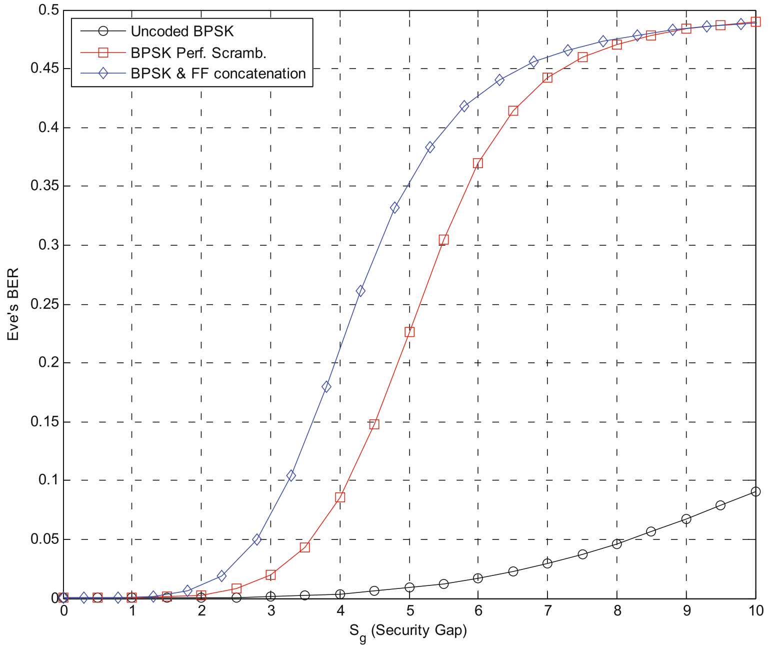

Fig. 8 Security gap performance without forward error correction ($s = 479$ bits, tail bit 1, and $k = n = 480$ bits), in the presence of BPSK modulation, perfect scrambling, and FF code

4 Joint iterative decoding for improved reliability

Joint iterative decoding (JID) in a concatenated system has been used to achieve high reliability [27] in spite of the high complexity. Since the proposed system is a serially concatenated structure, it is possible to use JID. In addition, in Section III-B, we demonstrated that the FF code has a coding gain through the use of a BCJR decoding algorithm for a few (or single) errors, and thus the performance gain from joint iterative decoding between LDPC and FF codes can be predicted in terms of the increasing SNR value. Figure 9 shows a schematic diagram of the joint iterative decoding for LDPC and FF concatenated system. The channel observations of k bit information and $n - k$ bit parity parts are $y_{ch,i}$ and $y_{ch,p}$, respectively. The extrinsic outputs of LDPC and FF codes are E_1 and E_2, and the a priori knowledge of LDPC and FF codes are A_1 and A_2, respectively. The dotted square shows a message transfer node (MTN) that processes the extrinsic information E_1 and E_2 to be a priori knowledge, A_1. The extrinsic output E_2 without high reliability causes performance loss of LDPC decoding due to its error propagation. To reduce the performance loss, MTN uses the extrinsic output E_1, which has higher reliability than E_2. In addition, MTN uses the correction factor α and scaling factor β to minimize error propagation by E_2 at high SNR. We define the log-likelihood ratio (LLR) as $L(x) = ln(P(x = 1)/P(x = 0))$. l_i and l_o are the number of LDPC decoding iterations and LDPC-FF code joint iterations, which we call inner and outer iterations, respectively.

When a decoder performs joint iterative decoding, the initial incoming messages to the channel decoder are given by:

$$\begin{aligned} L^0(C_{1,i}) &= L(y_{ch,i}) \\ L^0(C_{1,p}) &= L(y_{ch,p}) \end{aligned} \tag{12}$$

where $L^0(C_{1,i})$ and $L^0(C_{1,p})$ are the LLR values of information and parity messages, respectively, when l_o equals

Table 1 Security gap performances with uncoded BPSK, perfect scrambling and FF code over the AWGN channel

Code	$SNR_{E,max}[dB]$	$SNR_{B,min}[dB]$	S_g [dB]
uncoded BPSK	−14.94	9.59	24.53
Perf. scramb.	5.15	11.44	6.29
FF coded	4.25	9.8	5.55

Table 2 The types and numbers of operations needed to implement the perfect scrambling (randomly generated), FF soft decoding (BCJR), and FF hard decoding (as perfect scrambling)

Operations	Perfect scrambling (random matrix)	FF soft (BCJR)	FF hard decoding (perfect scrambling)
Addition	$n(n-1)$	$2(2t-1)n$	$n-1$
Multiplication	n^2	$8tn$	
Division		n	
Logarithm		n	

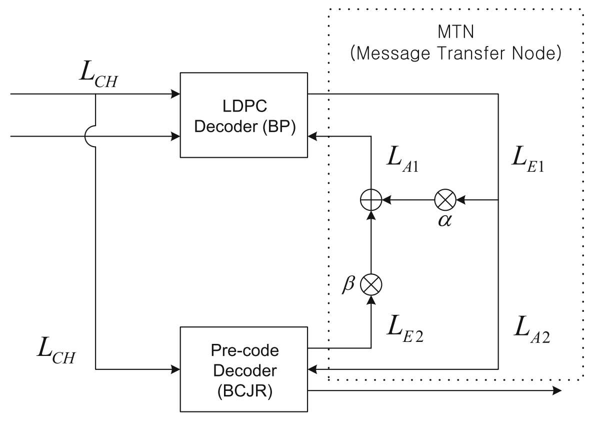

Fig. 9 Receiver structure of the joint iterative decoding for the concatenated (LDPC decoder and pre-code decoder) system

zero (first iteration). Then, the updated messages (a priori knowledge) from the FF decoder in the first iteration must be set up to zero as:

$$L^0(A_1) = 0$$

After LDPC decoding, the extrinsic output E_1 becomes the a priori input A_2. The FF decoder takes channel observations $y_{ch,i}$ and a priori knowledge A_2, and computes the extrinsic output E_2 as:

$$L^0(C_2) = L(y_{ch,i}) + L^0(A_2)$$

where $L^0(C_2)$ is the input LLR values of the FF decoder at the first iteration.

Therefore, the input message of the information part to the channel decoder in the l_o-th iteration, can be calculated recursively using

$$\begin{aligned} L^{l_0}(C_{1,i}) &= L(y_{ch,i}) + L^{l_0-1}(A_1) \\ &= L(y_{ch,i}) + \alpha \cdot L^{l_0-1}(E_1) + \beta \cdot L^{l_0-1}(E_2) \end{aligned} \tag{13}$$

$$\begin{aligned} L^{l_0}(C_2) &= L(y_{ch,i}) + L^{l_0}(A_2) \\ &= L(y_{ch,i}) + L^{l_0}(E_1) \end{aligned} \tag{14}$$

where α and β are the correction and scaling factors, respectively.

4.1 Computing the correction and scaling factors via Monte Carlo simulation

Both the α and β values are adopted to control the effect of the extrinsic messages, E_1 and E_2. As mentioned above, E_2 has an error propagation property and causes performance loss when it is used as a priori knowledge without any corrections of LDPC code. The extrinsic output E_1 used for correction of E_2 can also cause performance loss when LDPC output messages are taken as input messages because this would then oppose the general iterative decoding rule. Thus, the correction factor α must be less than one, $0 \leq \alpha < 1$. Since LDPC code as FEC used in this paper is linear, we can assume without loss of generality that the all-zero codeword is transmitted for a simple analysis. For the FF decoder, channel error ($L(y_{ch,i}) < 0$) or LDPC decoding failure ($L(y_{ch,i}) + L(E_1) < 0$) can cause error propagation. To reduce the loss by these impacts, the correction factor α should be taken as

$$|L(y_{ch,i})| \geq \alpha \cdot |L(E_1)|. \tag{15}$$

For joint iterative decoding, we assume 10 inner iterations (LDPC itr $= l_i = 10$) and the extrinsic message E_1 is the value after the inner iterations. To reduce the channel error and decoding failure after the inner iterations, the corrections factor α is chosen as

$$\alpha = \begin{cases} |\frac{L(y_{ch,i})}{L(E_1)}|, & \text{if } |L(y_{ch,i})| < |L(E_1)|, \\ 0.999999, & \text{otherwise.} \end{cases} \tag{16}$$

The value of the scaling factor β is derived based on α. Both α and β must be larger than zero and can take the maximum value of one. The channel error or decoding failure should be minimized by using the correction and scaling factors with extrinsic messages E_1 and E_2, respectively, so we have

$$L(y_{ch,i}) + \alpha \cdot L(E_1) + \beta \cdot L(E_2) \geq 0. \tag{17}$$

Since we assume that all-zero codeword modulated into $\mathbf{x} = \mathbf{+1} = [+1, +1, \cdots, +1]$ by BPSK $\{+1, -1\}$ is transmitted, the left-side of (17) must be larger than zero for the next iteration without errors.

Based on Eqs. (16) and (17), the scaling factor β is computed as

$$\beta = \begin{cases} \frac{|L(y_{ch,i}) + \alpha \cdot L(E_1)|}{|L(E_2)|}, & \text{if } \frac{|L(y_{ch,i}) + \alpha \cdot L(E_1)|}{|L(E_2)|} \leq 1, \\ 1, & \text{otherwise.} \end{cases} \tag{18}$$

To achieve the suitable values of α and β for real LLR values, we use Monte Carlo simulation for simplicity. We use Monte Carlo simulation to achieve the correction and scaling factors because error propagation property of FF code depends on error positions and the estimation of the error position is a difficult task. The inner and outer iterations, l_i and l_o are 10 and 1, respectively, and the number of transmitted frames (trials) is 10^7. Figure 10

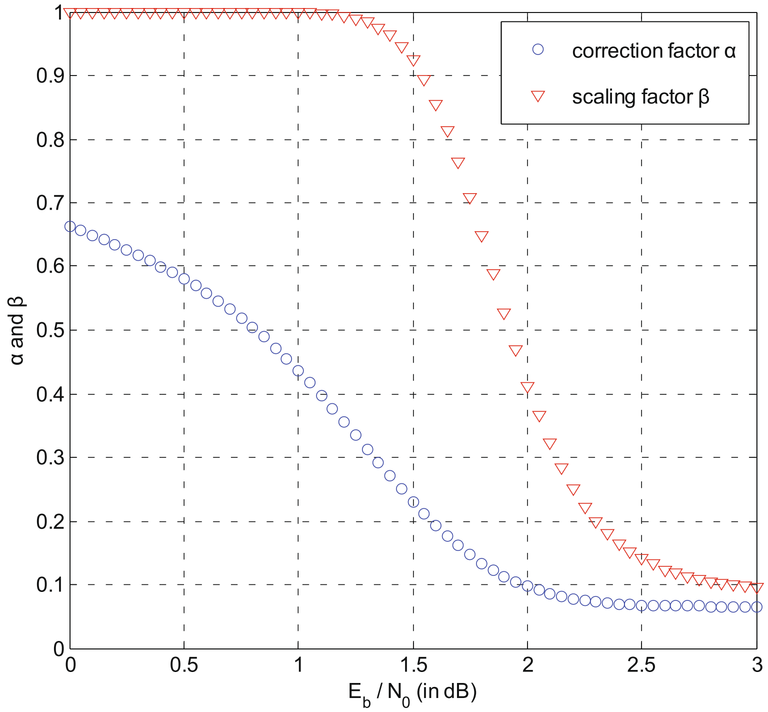

Fig. 10 Patterns of correction factor α and scaling factor β at each SNR values for the proposed JID system

shows the obtained correction and scaling sequences via Eqs. (16) and (18) at each SNR when $l_i = 10$ and $l_o = 1$. The figure shows how the correction and scaling factors differ and how they change for different SNR values. By observing the correction factor α and the scaling factor β of the proposed JID scheme from Fig. 10, we can conclude that: (1) the correction factor α is generally smaller than the scaling factor β for every SNR region, which reflects the general iterative decoding rule; (2) the correction and scaling factor values reduce by increasing SNR because error propagation effect should be reduced; (3) as the value of SNR increases, the correction and scaling factors will become smaller and finally converge to minimum values ($\alpha \rightarrow 0.06$, $\beta \rightarrow 0.08$), which infers that LDPC (FEC) decoding is becoming more reliable with the increase of SNR value.

4.2 Extrinsic information transfer (EXIT) chart analysis

The EXIT chart [20–22] is a useful analysis tool of the iterative decoding system. EXIT charts indicate mutual information exchange between the extrinsic information of two constituent codes. In most cases, the output LLR messages of these codes can be assumed to follow the Gaussian distribution. The extrinsic information between the constituent codes can then be sequentially used to process the computation. In this paper, the information from the channel (intrinsic information) and the output knowledge from the previous iteration (extrinsic information) can be used as the input of the current iteration, and the output of the current iteration can be used as the input of the next iteration. We use LDPC code and FF code as two constituent codes and assume that their input and output LLR are approximated by the Gaussian distribution.

Now suppose that I_A is the average mutual information between the coded bits and the a priori information, and I_E is the average mutual information between the coded bits and the extrinsic output. Function $T(I_A, E_b/N_0) = I_E$ is the EXIT chart function of the decoder and $T(\cdot)$ characterizes the information transfer in the decoder. Denoting the mutual information of the extrinsic information at the output of LDPC and the FF code by I_{E_1} and I_{E_2}, and the mutual information of the a priori information at the input of LDPC and the FF code by I_{A_1} and I_{A_2}, respectively, we have $I_{E_1} = T(I_{A_1}, E_b/N_0)$ and $I_{E_2} = T(I_{A_2}, E_b/N_0)$. To obtain the EXIT curve, we assume that the input LLR values, $L(A_1)$ and $L(A_2)$ are both symmetric. The symmetric conditions of LLR values are modeled as $L(A_1) \sim \mathcal{N}(m_1, 2m_1)$ and $L(A_2) \sim \mathcal{N}(m_2, 2m_2)$ such that m_1 and m_2 are the mean of the $L(A_1)$ and $L(A_2)$ messages, respectively. Therefore, the mutual information I_A between X and A can be written as $I_A = I(X; A) \doteq J(\sigma_A)$, as defined in equation (12) in [28]. Similarly, the mutual information $I_E = I(X; E)$ is defined as

$$I_E = \frac{1}{2} \cdot \sum_{x=-1,1} \int_{-\infty}^{+\infty} P_E(z|X=x) \times \log_2 \frac{2 \cdot P_E(z|X=x)}{P_E(z|X=-1) + P_E(z|X=+1)} dz. \quad (19)$$

In general, an analytical evaluation of the mutual information I_E in Eq. (19) is a difficult task. For simplicity, we use an approximated equation of the mutual information. Following [28], Eq. (19) can be arbitrarily closely approximated as

$$I_E \approx 1 - \frac{1}{N} \sum_{n=1}^{N} \log_2(1 + e^{-c_n \cdot L(c_n)}) \quad (20)$$

where N is the number of samples, c_n is the n-th codeword, and $L(c_n)$ is the LLR value of the n-th codeword such that $c_n \in \{+1, -1\}$. Figures 11 and 12 show the EXIT chart curve of the proposed system. In Fig. 11, the EXIT curves between LDPC and FF codes without α and β ($\alpha = 0, \beta = 0$) are plotted. In this case, a *pinch-off limit*[1] is finally at $E_b/N_0 = 2.62$ dB. In Fig. 12, the EXIT curves between LDPC and FF codes are plotted with the obtained α and β in section IV-A. The *pinch-off limit* is then at $E_b/N_0 = 2.26$ dB. As mentioned above, the FF code causes the error propagation. Without the input sequence of high reliability to the FF code, the error correction via the joint iterative decoding cannot be expected. For its error correction capability, we need the suitable correction and scaling factors.

4.3 Simulation results

In the previous subsections, it is suggested that the proposed scheme needs the correction and scaling factors in MTN for joint iterative decoding, and the joint iterative decoding of the proposed system is evaluated through EXIT chart curves. In this subsection, we evaluate the proposed system through BER and security gap performance.

As noted in Fig. 10, the values of α and β are sensitive functions of the signal-to-noise ratio. It may cause the entire system to become very complex and lead to performance loss when both values are wrongly evaluated. Therefore, the simulation results for the fixed values are also presented to avoid the impact of wrong evaluation. The fixed values are selected in the SNR region having the gain of the joint iterative decoding compared to the perfect scrambling. The fixed values selected at low SNR region ($\leq$ 1.5 dB) may cause critical error propagation at high SNR. In addition, the fixed values selected at high SNR region ($\geq$ 2.8 dB) render it difficult to achieve the joint iterative decoding gain since the values are too small. Based on these rules, the values are selected as $\alpha = 0.07$ and $\beta = 0.18$, which are optimized at 2.4 dB. The reasons for these values are as follows: (i) the JID at high SNR

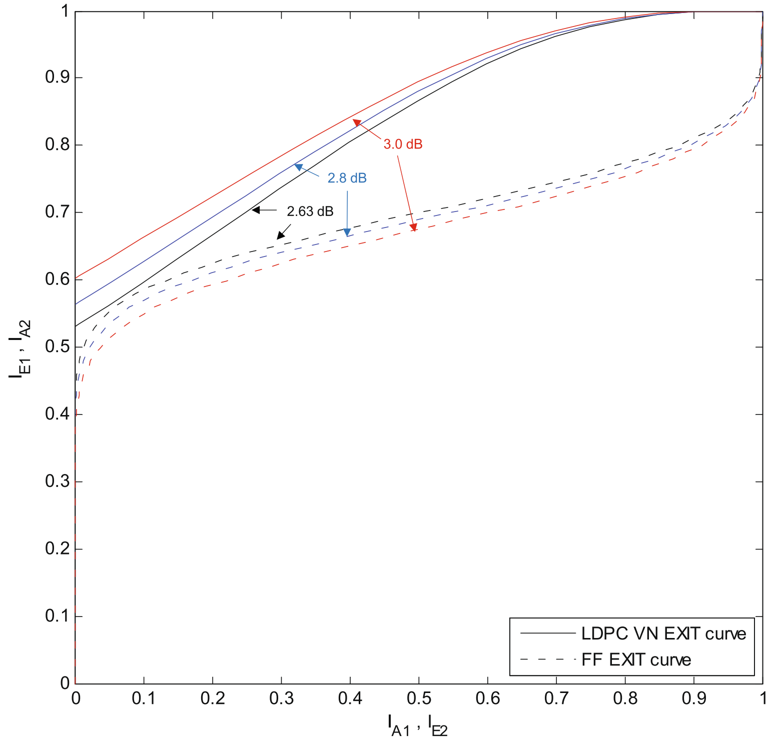

Fig. 11 EXIT chart curves of joint FF and LDPC codes (without using the correction factor α and the scaling factor β)

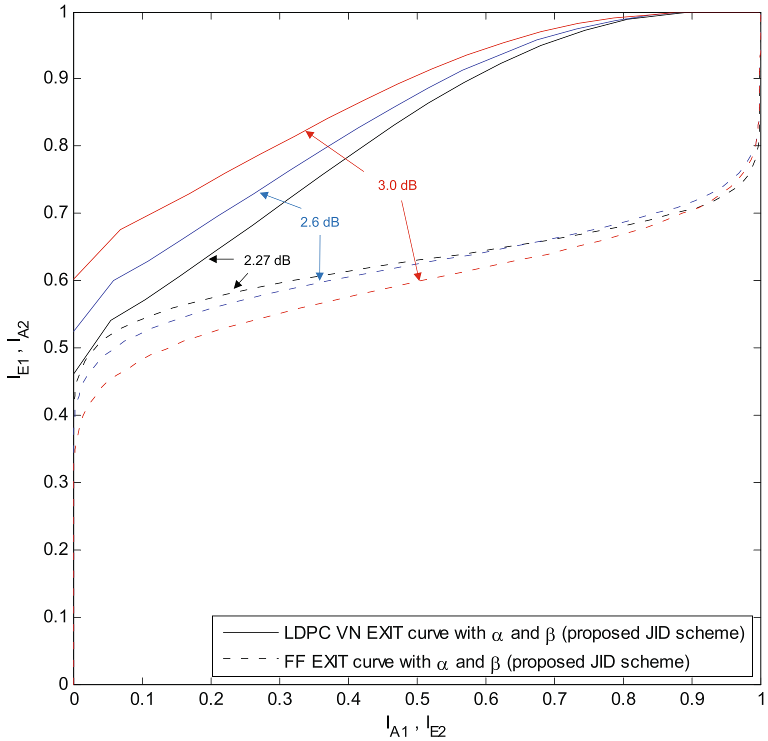

Fig. 12 EXIT chart curves of joint FF and LDPC codes using the proposed JID scheme (with optimal correction factor α and the scaling factor β)

region has some of iterative decoding gain compared to the perfect scrambled LDPC code. For use of the fixed values, $SNR_{B,min}$ should possibly be kept as small as $SNR_{B,min}$ for use of the optimized values. That is, despite the use of the fixed values, outstanding reliability performance should be achieved at high SNR region; ii) error propagation effect is properly maintained at low SNR region by using these values. That is, the $SNR_{E,max}$ for use of the fixed values should be kept as close to the $SNR_{E,max}$ as possible for use of the optimized values. These optimized values shown in Fig. 12 are evaluated for the best security performance at each SNR. Although the fixed values ($\alpha = 0.07$, $\beta = 0.18$) are experimentally selected to show the prevention of the wrong evaluation impact as an example, the security performance will differ following the values that are selected. In summary, these values ($\alpha = 0.07$, $\beta = 0.18$) are relevantly selected by considering error propagation at low SNR and error correction at high SNR.

For comparison purposes, LDPC codes are considered to have the same parameters as those mentioned in Section 3, $k = 480$ and $n = 960$. The secrecy rate is $R_s \approx R_d = R_c = 0.5$ and the BPSK modulation $\{+1, -1\}$ is used in our simulations. The maximum LDPC iteration is 100. For the joint iterative decoding, the number of inner and outer iterations, l_i and l_o are 10 and 10, respectively. Figure 13 and Table 3 show the BER performances of the systematic, perfect scrambled, serially concatenated JID scheme through MTN with optimized α and β, and the JID scheme with fixed α and β values. From Fig. 13 and Table 3, it is observed that the proposed concatenated and JID systems have performance improvements of about 0.205 and 0.51 dB over the perfect scrambling scheme in [16, 17] at a BER of 10^{-6}. We can observe that the joint iterative effect of the proposed JID scheme eventually increases SNR, which is in good agreement with the respective EXIT curve (*pinch-off limit*) of Fig. 12. A performance improvement at high SNR means that it can achieve a steep BER curve and a reduced security gap.

The security gap performances for the various systems are plotted in Fig. 14, while Table 3 shows the security gap performances. Due to error floor phenomenon of LDPC code, Bob's maximum bit error probability $P^{B}_{e,max}$ is set to 10^{-6}, and the improved security of the proposed system is thus more tangible. From Fig. 14 and Table 3, the security gap performances of the JID system using MTN (with optimized/fixed α and β) at $P^{E}_{e,min} = 0.4$ are about 2.26 and 2.37 dB. The security gap performance of the LDPC and FF concatenation at $P^{E}_{e,min} = 0.4$ is about 2.615 dB. It is observed that the proposed systems of serial concatenation and JID using optimized α and β have performance improvements of about 0.19 dB and 0.545 dB over the perfect scrambling scheme, respectively. In the case of

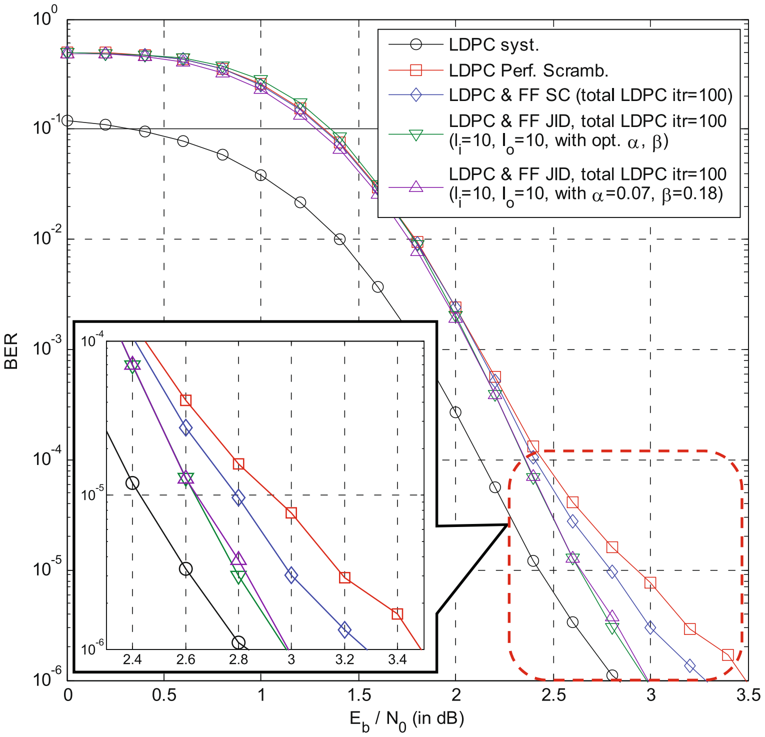

Fig. 13 BER performances versus the LDPC systematic, perfectly scrambled LDPC, LDPC and FF concatenated system (No joint iteration), and the joint iterative decoding (inner iteration $I_i = 10$ and outer iteration $I_o = 10$) between LDPC and FF decoders with values of the correction factor α and the scaling factor β (Total LDPC iteration 100)

the JID scheme using MTN with the fixed α and β, the performance improvement for the security gap is about 0.435 dB over the perfect scrambling scheme.

In conclusion, the SISO decoder of the FF code provides a performance improvement over the scrambled scheme and we can achieve a better performance improvement using the proposed JID scheme. Furthermore, the proposed system using the fixed factors has similar security/reliability performances to that using the optimized factors and can still achieve the performance improvement over the scrambled scheme. From the figure, we can also observe that the security gap advantage vanishes as Eve's BER tends toward the ideal value of 0.5, hereafter $P^E_{e,min} \geq 0.45$. However, since $P^E_{e,min} \geq 0.4$ is sufficiently significant for a practical system, physical layer security as defined in this paper can be achieved.

Table 3 Security gap performances with systematic LDPC, perfect scrambled LDPC, LDPC-FF serially concatenated (SC) and LDPC-FF JID (opt./fix.) over the AWGN channel

Code	$SNR_{E,max}[dB]$	$SNR_{B,min}[dB]$	S_g [dB]
Syst.	−11.87	2.835	14.705
Perf. scramb.	0.685	3.49	2.805
SC	0.67	3.285	2.615
JID (opt.)	0.72	2.98	2.26
JID (fix.)	0.62	2.99	2.37

Although the proposed JID scheme has the advantage of enhanced reliability/security performances, this is achieved from an extra complexity/decoding delay. Basically, extra decoding complexity is needed since the FF decoding procedure is performed I_o times for JID. If more JID is demanded, the extra complexity will be increased.

In [17], Baldi et al. demonstrated that physical layer security can be achieved by using a very simple feed-back mechanism based on Hybrid Automatic Repeat reQuest (Hybrid ARQ or HARQ) when Bob's channel is not "less noisy" than Eve's channel. They used the soft-combining scheme of HARQ so that Bob can exploit a number of transmissions $Q < Q_{max}$ for decoding each frame and Eve receives all retransmissions requested by Bob. Similarly, we provide a simulation result on the use of the HARQ scheme with the perfect scrambled LDPC and the proposed FF-LDPC JID scheme using fixed factors. The maximum number of transmissions is $Q_{max} = 3$. The numerical results similar to that provided in [17] are observed for HARQ with the proposed scheme. The FER performances with HARQ (soft-combining) for the perfect scrambled LDPC and the proposed JID scheme

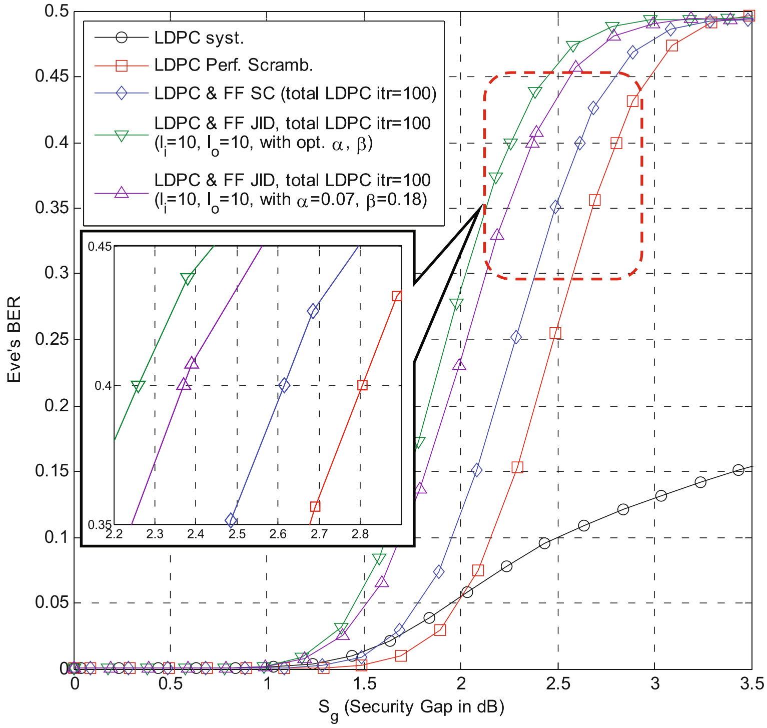

Fig. 14 Security gap performances versus the LDPC systematic, perfectly scrambled LDPC, LDPC and FF concatenated system (No joint iteration), and the joint iterative decoding (inner iteration $l_i = 10$ and outer iteration $l_o = 10$) between LDPC and FF decoders with values of the correction factor $\boldsymbol{\alpha}$ and the scaling factor $\boldsymbol{\beta}$ (total LDPC iteration 100)

are plotted in Fig. 15. From the figure, the FER performances of the perfect scrambled LDPC and the proposed JID scheme are the dotted and solid lines, respectively. The fluctuations are observed in the figure because an average number of transmissions is a decreasing function for Bob's received SNR. Also, in this case, we could observe the improved reliability of the proposed JID scheme for each result. The error correction capability of the proposed scheme could be improved, while the correction capability of the scrambled scheme depends only on LDPC code. Hence, the proposed JID scheme allows us to achieve the desired level of physical layer security.

4.4 Randomness measurement

The proposed precode includes operations between consecutive symbols. From these operations, the proposed precode has a correlation effect and is able to employ soft decision decoding. Due to the correlation effect and soft decision decoding, Eve's decoding performance is better than that of the perfect scrambling scheme. However, the security gap between Bob and Eve is maintained since both performances are equally enhanced. The proposed scheme may have a negative impact on the randomness of the produced sequence since the FF code is highly structured. For this reason, the entire distribution of errors should be analyzed since Eve's average error rate does not guarantee the randomness of the decoded sequence.

From Fig. 3, Eve's received Z decodes LDPC decoded message $\hat{M}_E = M + \mathbf{E}$ by using the BP decoder, where M and $\mathbf{E}$ are LDPC codeword and the error vector after LDPC decoding, respectively. In addition, through the FF decoder, Eve's FF decoder outputs the FF decoded message $\hat{U}_E = U + \mathbf{e}$, where U and $\mathbf{e}$ are the information message and error vector after FF decoding, respectively. For the erroneous frame, let e_i^l be the number of errors for the ith position in the l-th erroneous frame, l_{max} be the total number of erroneous frames, t_i be the total number of errors at the ith position, $t_i = \sum_{l=1}^{l_{max}} e_i^l$, and T be the total number of errors, $T = \sum_{i=1}^{n} t_i$, where n is the length of the information bits. Therefore, the error expectation value (or bit error probability) at the i-th position for an erroneous frame is $P_m^i = t_i/l_{max}$. Therefore, P_m^i is the bit error probability for each position when $\mathbf{e} \neq \mathbf{0}$.

Figure 16 shows the results of randomness measurement (the entire distribution of errors) of the final decoded message $\hat{U}_E$ with errors for the perfectly scrambled LDPC and the proposed JID scheme. In the case of the perfectly scrambled LDPC, error positions after

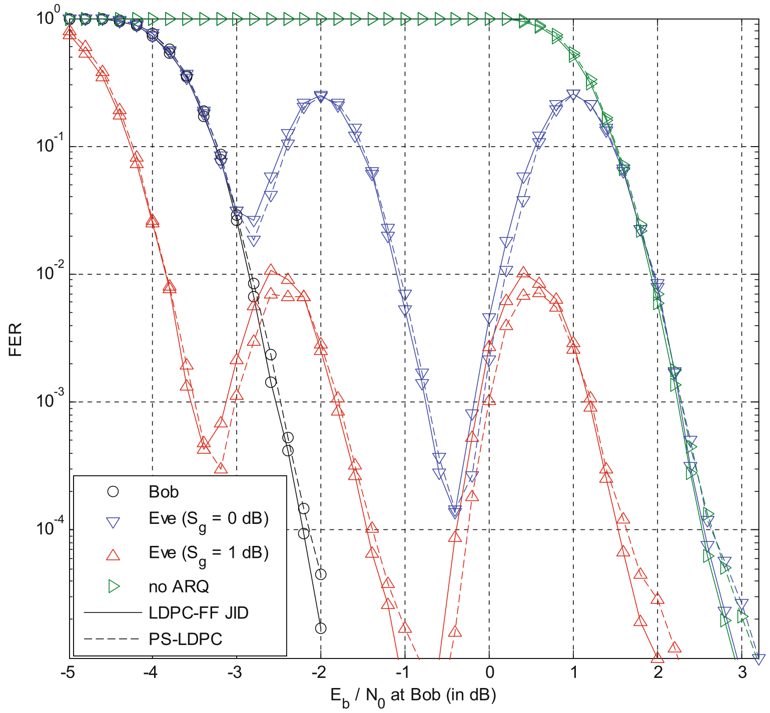

Fig. 15 The FER performances versus Bob's SNR for the perfect scrambled LDPC (*dotted line*) and the proposed FF-LDPC JID (*solid line*) using the fixed factors ($\alpha = 0.07$, $\beta = 0.18$) with soft-combining HARQ ($Q_{max} = 3$) and different values of security gap (S_g)

descrambling are randomly distributed since the descrambling matrix is randomly generated with the density of 0.5. Uniform error distribution is observed for every position, even though error positions are not i.i.d. In the case of the LDPC-FF JID, P_m^i at the front and tail parts has relatively low values since the FF decoder employs BCJR algorithm which is set up with a high probability for the zero state at the first and last step. However, even in the high SNR region, the randomness is kept to $P_m^i \approx 0.5$ for the positions of most of the parts, except a few of the front and last positions. In other words, when Eve uses the same decoder as that used by Bob, she cannot extract any useful information since error positions of the proposed scheme are as randomly distributed as the perfectly scrambled scheme for most of the parts. Because $P_m^i \geq 0.4$ means that the error probability over 0.4 has fairly high uncertainty at the *i*th position (we cannot state about entropy since error positions are not independent).

5 Conclusions

In this paper, we have examined security-processing schemes for physical layer security. We proposed a serially concatenated system that consists of an outer code and conventional FEC as an inner code. Compared with previous works relating to channel coding for physical layer security, the puncturing scheme for LDPC code (or linear block code) has a weakness in that it should be required higher signal power to achieve reliability than scrambling scheme. The disadvantages of the scrambling scheme as unitary rate coding are that it is only capable of reducing the security gap and it does not provide the error correction capability. The proposed security scheme adopts the FF code using a SISO decoding procedure (BCJR algorithm). We demonstrated that the proposed scheme is capable of performing error correction and error propagation simultaneously. Simulation results confirm that the FF code using a BCJR algorithm has an improved reliability performance and reduced security gap.

Furthermore, we proposed a joint iterative decoding algorithm between the FF code and conventional LDPC to improve the reliability performance through the bit and frame error rate with a correction factor α and scaling factor β obtained by using Monte Carlo simulation. These factors are the functions of the signal-to-noise ratio. In the case of the proposed JID using these factors, our best results indicate reliability/security-gap performance improvements of 0.51 and 0.545 dB, respectively. The reliability performance of the proposed JID scheme using these factors is observed to be only 0.145 dB away from the systematic LDPC code. Despite the use of fixed factors to avoid the impact of wrong evaluation,

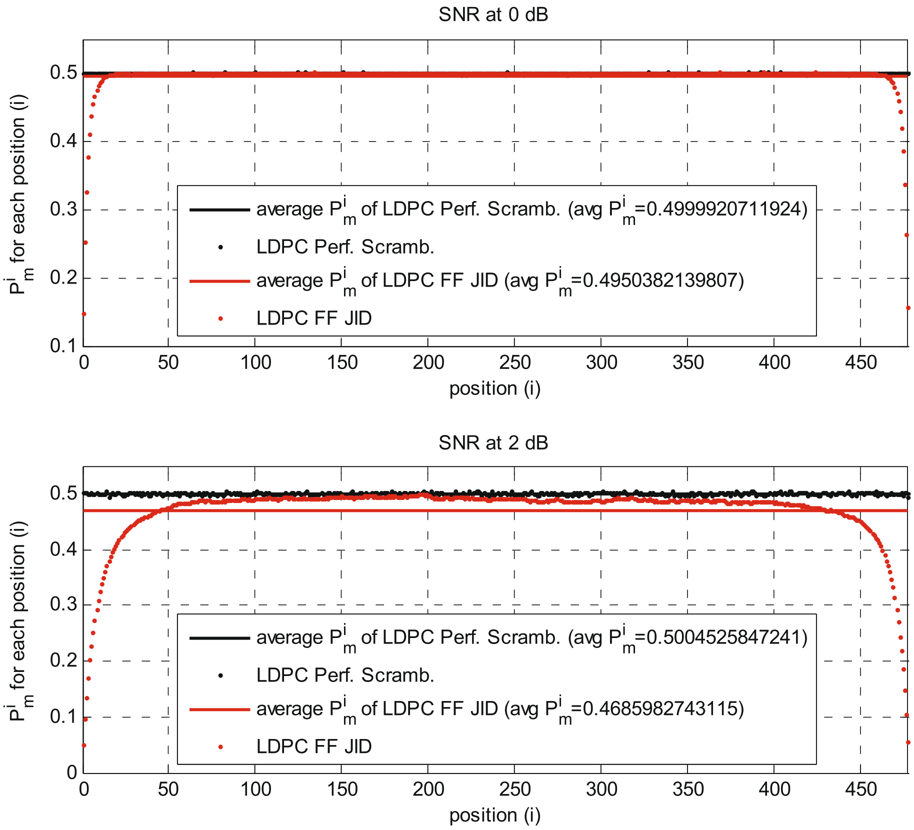

Fig. 16 The results of randomness measurement using the perfectly scrambled LDPC and the proposed JID scheme at 0 and 2 dB region

our results indicate reliability/security-gap performance improvements to perfect scrambled LDPC of 0.5 and 0.435 dB, respectively. It is demonstrated that error floor phenomenon of LDPC code in the high SNR region can be reduced when using joint iterative decoding with the proper α and β; thus, a reduced security gap can be achieved. This is analyzed via the EXIT chart curve. In future works, equivocation rate analysis of the proposed scheme will be performed with α and β for an information-theoretic approach.

Endnote

[1] *pinch-off limit* means that both transfer characteristics of two constituent codes are just about to intersect (see [20, 21]).

Acknowledgements

This work was supported by the ICT R&D program of MSIP/IITP [1711028311, Reliable crypto-system standards and core technology development for secure quantum key distribution network].

Competing interests

The authors declare that they have no competing interests.

Received: 1 September 2015 Accepted: 25 October 2016
Published online: 09 December 2016

References

1. CE Shannon, Communication theory of secrecy systems. Bell Syst. Tech. J. **28**, 656–715 (1949)
2. AD Wyner, The wire-tap channel. Bell Syst. Tech. J. **54**(8), 1355–1387 (1975)
3. L Ozarow, AD Wyner, Wire-tap channel. II.AT&T Bell Laboratories Tech. J. **63**(10), 2135–2157 ((1984))
4. I Csiszar, J Korner, Broadcast channel with confidential messages. IEEE Trans. Inf. Theory. **24**(3), 339–348 (1978)
5. A Thangaraj, S Dihidar, AR Calderbank, S McLaughlin, JM Merolla, Applications of LDPC codes to the wiretap channels. IEEE Trans. Inf. Theory. **53**(8), 2933–2945 (2007)
6. SK Leung-Yan-Cheong, ME Hellman, The Gaussian wiretap channel. IEEE Trans. Inf. Theory. **24**(4), 451–456 (1978)
7. D Klinc, J Ha, S McLaughlin, J Barros, BJ Kwak, LDPC codes for the Gaussian wiretap channel. IEEE Trans. Inf. Forensics Secur. **6**(3), 532–540 (2011)
8. D Klinc, J Ha, S McLaughlin, J Barros, BJ Kwak, in *IEEE Global Telecommunications Conference (GLOBECOM 2009)*. LDPC codes for physical layer security (Honolulu, USA, p. 2009
9. CW Wong, TF Wong, JM Shea, in *IEEE GLOBECOM Workshops*. LDPC code design for the BPSK-constrained Gaussian wiretap channel USA, Houston, p. 2011
10. CW Wong, TF Wong, JM Shea, Secret-sharing LDPC codes for the BPSK-constrained Gaussian wiretap channel. IEEE Trans. Inf. Forensics Secur. **6**(3), 551–564 (2011)
11. M Baldi, F Chiaraluce, N Laurenti, S Tomasin, F Renna, Secrecy Transmission on parallel channels: theoretical limits and performance of practical codes. IEEE Trans. Inf. Forensics Secur. **9**(11), 1765–1779 (2014)
12. RG Gallager, Low-density parity-check codes. IRE Trans Inf. Theory. **8**(1), 21–28 (1962)
13. T Richardson, R Urbanke, The capacity of low-density parity-check codes under message-passing decoding. IEEE Trans. Inf. Theory. **47**(2), 599–618 (2001)

14. SY Chung, JGD Forney, T Richardson, R Urbanke, On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. IEEE Commun. Lett. **5**(2), 58–60 (2001)
15. S ten Brink, G Kramer, A Ashikhmin, Design of low-density parity-check codes for modulation and detection. IEEE Trans. Commun. **52**(4), 670–678 (2004)
16. M Baldi, M Bianchi, F Chiaraluce, in *IEEE Information Theory Workshop (ITW 2010)*. Non-systematic codes for physical layer security Ireland, Dublin, p. 2010
17. M Baldi, M Bianchi, F Chiaraluce, Coding with scrambling, concatenation, and HARQ for the AWGN wire-tap channel: a security gap analysis. IEEE Trans. Inf. Forensics Secur. **7**(3), 883–894 (2012)
18. RJ McEliece, A public-key cryptosystem based on algebraic coding theory. DSN Prog Rep. **44**, 114–116 (1978)
19. M Baldi, N Maturo, G Ricciutelli, F Chiaraluce, Security gap analysis of some LDPC coded transmission schemes over the flat and fast fading Gaussian wire-tap channels.EURASIP. J. Wirel. Commun. **2015**(1), 1–12 (2015)
20. S ten Brink, Convergence behavior of iteratively decoded parallel concatenated codes. IEEE Trans. Commun. **49**, 1727–1737 (2001)
21. S ten Brink, Designing iterative decoding schemes with the extrinsic information transfer chart. AEÜ Int. J. Electron. Commun. **54**(6), 389–398 (2000)
22. S ten Brink, Convergence of iterative decoding. Electron. Lett. **35**(10), 806–808 (1999)
23. JG Proakis, *Digital Communications*, 4th Ed. McGraw-Hill, New York, 2000)
24. Lin Shu, Costello Danial J, *Error Control Coding: Fundamentals and Applications*, 2nd. (Pearson Prentice Hall, Upper Saddle River, 2004)
25. IEEE standard for local and metropolitan area networks-part 16: air interface for fixed and mobile broadband wireless access systems. IEEE Std P802.16e/D12 (2005)
26. L Bahl, J Cocke, F Jelinek, J Raviv, Optimal decoding of linear codes for minimizing symbol error rate. IEEE Trans. Inf. Theory. **20**(2), 284–287 (1974)
27. J Hagenauer, E Offer, L Papke, Iterative decoding of binary block and convolutional codes. IEEE Trans. Inf. Theory. **42**, 429–445 (1996)
28. M Tuechler, J Hagenauer, *EXIT charts and irregular codes. in IEEE Conference on Information Sciences and Systems (CISS'02)*. (Princeton, USA, 2002)