Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7063758, 12 pages
https://doi.org/10.1155/2018/7063758

Research Article

Code-Hopping Based Transmission Scheme for Wireless Physical-Layer Security

Liuguo Yin[1,2] **and Wentao Hao**[2,3]

[1]*Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China*
[2]*EDA Laboratory, Research Institute of Tsinghua University in Shenzhen, Shenzhen, China*
[3]*School of Aerospace Engineering, Tsinghua University, Beijing 100084, China*

Correspondence should be addressed to Liuguo Yin; yinlg@tsinghua.edu.cn

Received 23 November 2017; Revised 9 February 2018; Accepted 28 February 2018; Published 3 April 2018

Academic Editor: Zesong Fei

Copyright © 2018 Liuguo Yin and Wentao Hao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the broadcast and time-varying natures of wireless channels, traditional communication systems that provide data encryption at the application layer suffer many challenges such as error diffusion. In this paper, we propose a code-hopping based secrecy transmission scheme that uses dynamic nonsystematic low-density parity-check (LDPC) codes and automatic repeat-request (ARQ) mechanism to jointly encode and encrypt source messages at the physical layer. In this scheme, secret keys at the transmitter and the legitimate receiver are generated dynamically upon the source messages that have been transmitted successfully. During the transmission, each source message is jointly encoded and encrypted by a parity-check matrix, which is dynamically selected from a set of LDPC matrices based on the shared dynamic secret key. As for the eavesdropper (Eve), the uncorrectable decoding errors prevent her from generating the same secret key as the legitimate parties. Thus she cannot select the correct LDPC matrix to recover the source message. We demonstrate that our scheme can be compatible with traditional cryptosystems and enhance the security without sacrificing the error-correction performance. Numerical results show that the bit error rate (BER) of Eve approaches 0.5 as the number of transmitted source messages increases and the security gap of the system is small.

1. Introduction

Information security and reliability are two crucial issues in wireless communications. Traditionally, communication systems correct transmission errors at the physical layer based on channel codes and cope with eavesdropping at the application layer based on cryptographic algorithms. In practical scenarios, there will be residual errors in the decoded messages due to the time-varying nature of wireless channels, which may cause severe error diffusion in the decryption. In addition, with the rapid increase of the eavesdropper's computing power, these computational-complexity based encryption algorithms will be easier to break, such as A5/1 in the GSM.

Alternatively, the schemes based on physical-layer security aim to tackle these two crucial issues at the physical layer. Shannon [1] first studied secure communication from an information theoretic perspective in which a preshared secret key between the legitimate parties is used to encrypt the source message. To avoid the key agreement and exploit the inherent randomness of wireless channels, Wyner [2] presented the degraded wiretap channel model in which a transmitter wants to send a secret message to a legitimate receiver through the main channel. This message is also perceived by an eavesdropper through the degraded wiretap channel. The secrecy capacity is defined as the supremum of all the achievable secure and reliable transmission rates. Then, Wyner's original work was generalized to broadcast channels [3] and Gaussian channels [4]. Moreover, the secrecy capacity of fading wiretap channels [5], MIMO wiretap channels [6], and multiuser wiretap channels [7, 8] has been derived in the literature. In these works, the equivocation of Eve is a widely accepted metric for security, which is defined as the conditional entropy of the source message given her noisy observation [9].

Many coding techniques are applied to wiretap channels to make the secrecy transmission rate approach the secrecy capacity, in other words, for the equivocation of Eve to

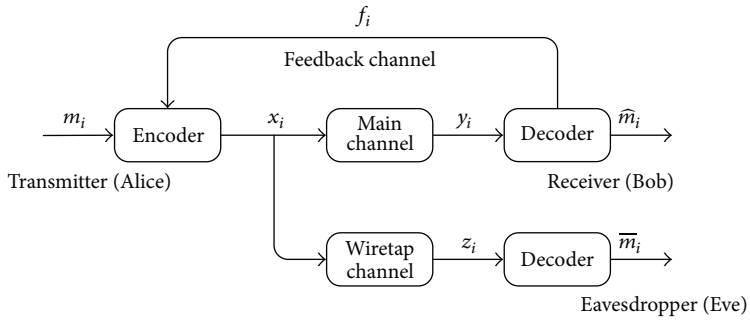


Figure 1: Wiretap channel model with public feedback.

approximate the entropy of the source message. For binary erasure wiretap channels, Thangaraj et al. [10] proposed a coding technique based on the dual of LDPC codes and showed that the secrecy capacity can be achieved by this technique. For symmetric discrete memoryless wiretap channels, Andersson et al. [11] proved that nested polar codes can achieve the whole rate-equivocation region. In addition, this coding technique is further applied to relay-eavesdropper channels [12], block fading channels [13], and multiuser channels [14]. These schemes are really effective when the code length is sufficient, but may be difficult to implement in practical systems.

When we consider the design of practical coding schemes, another valuable metric is the BER [15, 16]. In fact, it is difficult for the eavesdropper to recover any information from the decoded message when she experiences a BER of about 0.5 and the errors are randomly distributed. Security gap is defined as the quantity difference between Bob's and Eve's channels required to achieve a sufficient level of physical-layer security, while ensuring that Bob reliably receives the information [17]. In [17], punctured systematic LDPC codes were exploited to obtain a small security gap. Furthermore, a nonsystematic solution based on scrambled systematic LDPC codes was proposed in [18]. It was proved that the achievable security gap of the scrambled scheme is smaller than that of the punctured method. In [19], scrambling, concatenation, and hybrid automatic repeat-request (HARQ) were combined to reduce the security gap even further. In addition, dynamic LDPC codes are used to enhance the security of the communication system [20]. And protograph LDPC codes [21, 22] can also be used to guarantee the security of the transmission.

In this paper, we propose a scheme based on code-hopping for secrecy transmission over wireless wiretap channels. In the proposed scheme, with ARQ mechanism, the transmitter and the legitimate receiver can select the source messages in real time to distill the secret key. This secret key is then mapped into the parity-check matrix of LDPC codes, which is used to encode the source message. As for the eavesdropper, the uncorrectable decoding errors prevent her from generating the same secret key as the transmitter and the legitimate receiver. Therefore, she cannot obtain the correct parity-check matrix to recover the source message. Theoretical analysis demonstrates that it is difficult for the eavesdropper to generate the same secret key as legitimate parties. Simulation results show that the BER of Eve approaches 0.5 as the number of transmissions increases and the security gap of the system is small.

The remainder of the paper is organized as follows. We introduce our system model and the design of the encoder and decoder in Section 2. In Section 3, the dynamic secret key generation algorithm is proposed and the security of the secret key is well examined. In Section 4, we construct a large number of parity-check matrices of LDPC codes based on the technique we called structured-random protograph expanding. Encoder and Decoder implementation of structured-random LDPC codes are discussed in Section 5. In Section 6, we analyze the reliability and the security performance of our scheme. And some numerical results are given in Section 7. Finally, concluding remarks are provided in Section 8.

2. The Proposed Secrecy Transmission Scheme

In this section, we will first introduce the wiretap channel model with public feedback and the concept of security gap. Then, we will propose our secrecy transmission scheme along with the design of encoder and decoder.

2.1. System Model. As shown in Figure 1, for $i = 1, 2, \ldots$, message m_i is a sequence of uncoded bits and the length of m_i is s. A transmitter named Alice wants to send m_i to a legitimate receiver named Bob through the main channel, but her transmission is also perceived by an eavesdropper named Eve through the wiretap channel. To keep m_i as secret as possible, Alice encodes each length-s message m_i to a length-n codeword x_i by her encoder. The corresponding received codewords by Bob and Eve are denoted by y_i and z_i, which are recovered by the decoder as $\widehat{m}_i$ and $\overline{m}_i$, respectively. Additionally, in our model, Bob can use a public feedback channel to inform Alice whether the current codeword is decoded successfully with a feedback signal f_i. If there occurs a decoding error at Bob, Alice will retransmit the source message until Bob successfully recovers it or the number of retransmissions reaches the maximum. Taking into account the application in practical scenarios, both channels are assumed to be Gaussian or fading channels:

$$\begin{aligned} y_i &= h_i^B x_i + n_i^B, \\ z_i &= h_i^E x_i + n_i^E, \end{aligned} \tag{1}$$

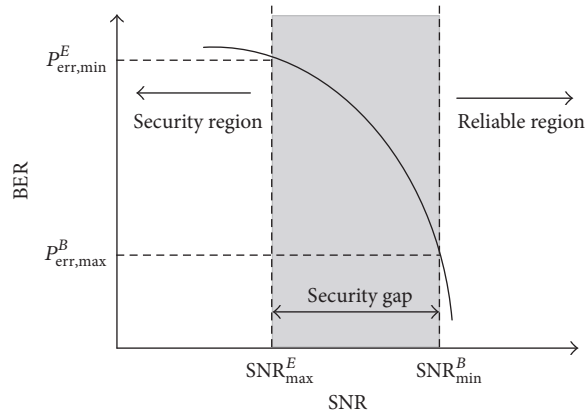


Figure 2: Security gap established by the BER curve.

where h_i^B and h_i^E are the fading coefficients, which are equal to one for Gaussian scenario and follow a certain distribution for fading scenario and n_i^B and n_i^E are zero mean Gaussian noise; $n_i^B \sim \mathcal{N}(0, \sigma_B^2)$ and $n_i^E \sim \mathcal{N}(0, \sigma_E^2)$.

Let P_e^B and P_e^E denote the average BER of Bob and Eve, respectively. As shown in Figure 2, to guarantee the reliability, P_e^B should be lower than a given threshold $P^B_{\text{err,max}}$ ($\approx$0). And to achieve the confidentiality, P_e^E should be larger than a given threshold $P^E_{\text{err,min}}$ ($\approx$0.5). Particularly, if P_e^E is close to 0.5 and the errors are randomly distributed, Eve cannot extract any information from the decoded messages. Based on this observation, the reliability and security of the transmission are guaranteed if conditions (2) and (3) can be satisfied [17], respectively:

$$P_e^B \le P^B_{\text{err,max}} = P_e\left(\text{SNR}^B_{\text{min}}\right), \tag{2}$$

$$P_e^E \ge P^E_{\text{err,min}} = P_e\left(\text{SNR}^E_{\text{max}}\right), \tag{3}$$

where $\text{SNR}^B_{\text{min}}$ is the lowest signal-to-noise ratio at Bob to guarantee reliability, $\text{SNR}^E_{\text{max}}$ is the highest signal-to-noise ratio at Eve to guarantee security, and $P_e(\cdot)$ denotes the BER as the function of SNR. Then, the security gap is defined as follows [17]:

$$\text{Sg} = \text{SNR}^B_{\text{min}} - \text{SNR}^E_{\text{max}}, \tag{4}$$

where the SNRs are expressed in decibels (dB). Without sacrificing the error-correcting performance of the transmission system, our design targets are making the BER of Eve approach 0.5 and reducing the security gap as much as possible.

2.2. Design of the Coding Scheme. To exploit the inherent randomness of wireless channels and the uncorrectable decoding errors of Eve, our scheme is implemented such that the secret keys are distilled from the un-retransmitted source messages, which are then used to generate the parity-check matrices of LDPC codes. During the transmission, the source messages are encoded and decoded by these dynamic parity-check matrices. The block diagrams of the encoder and the decoder are illustrated in Figures 3 and 4, respectively.

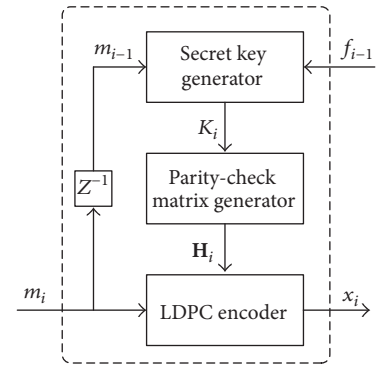


Figure 3: Block diagram of the encoder. Note that the z^{-1} block denotes a delay unit.

In the encoder of Alice, the secret key K_i is updated dynamically according to the received feedback signal f_{i-1} and the source message m_{i-1}. If f_{i-1} = ACK, then K_i will be updated according to m_{i-1}. If f_{i-1} = NACK, K_i will remain unchanged. The detailed procedure of key update will be discussed in Section 3. Then, the secret key K_i will be used to generate the parity-check matrix of LDPC codes as follows:

$$\mathbf{H}_i = f_H\left(K_i\right), \tag{5}$$

where $f_H(\cdot)$ is the mapping from the secret key to the parity-check matrix. For each source message m_i, it will be encoded by the corresponding $\mathbf{H}_i$.

In the decoder of Bob, the integrity of the decoded source message $\widehat{m}_i$ will be checked. If $\widehat{m}_i$ is recovered without errors, the public feedback signal f_i = ACK; otherwise, f_i = NACK. Instead of using the syndrome of the decoded codeword to determine the correctness of $\widehat{m}_i$, we use the cyclic redundancy check (CRC) algorithm to perform integrity check. This is because when the decoded codeword converges to another valid codeword of $\mathbf{H}_i$, the method based on the syndrome cannot detect errors. As for the symmetric key K_i and the

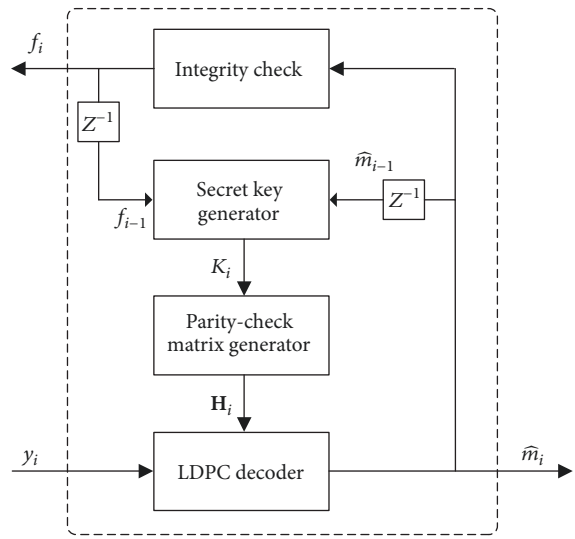


FIGURE 4: Block diagram of the decoder. Note that the z^{-1} block denotes a delay unit.

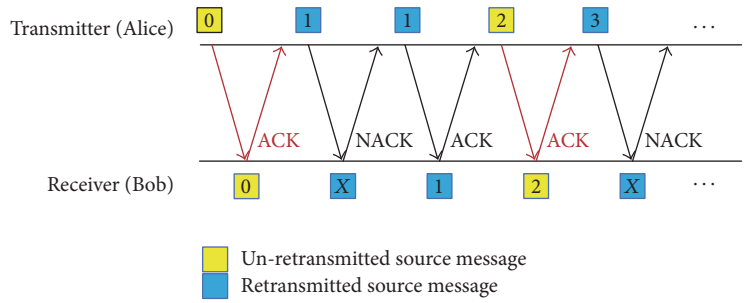


FIGURE 5: The process of automatic source massage selection.

parity-check matrix $\mathbf{H}_i$, they will be generated as in Alice's encoder.

3. Dynamic Secret Key Generation Scheme

In this section, we will introduce the dynamic secret key generation algorithm and the mathematical rationales behind it. With this algorithm, Alice and Bob can select the appropriate source messages during the transmission and then distill the secret key based on the universal hashing family.

3.1. Automatic Source Message Selection. In this subsection, we will show how Alice and Bob select appropriate source messages in real time during the transmission, which is then hashed into the dynamic secret key. We define ψ_i^t and ψ_i^r as the source message set that is used to generate the secret key K_i at Alice and Bob, respectively. To give Alice and Bob an advantage over Eve, only un-retransmitted source messages will be included in ψ_i^t and ψ_i^r. Before the communication begins, $\psi_0^t = \psi_0^r = (\psi_{0,0}, \psi_{0,1}, \ldots, \psi_{0,D-1})$, where D is the number of source messages in the set and $\psi_{0,j}$ is the public agreed initialized binary vector of length-s, $j = 0, 1, \ldots, D-1$.

As illustrated in Figure 5, during the transmission, Alice transmits a source message m_i and waits for the corresponding feedback signal f_i before transmitting any new source message. If the received feedback signal f_i = NACK, ψ_{i+1}^t will remain unchanged compared to ψ_i^t:

$$\begin{aligned}\psi_{i+1}^t &= (\psi_{i+1,0}, \psi_{i+1,1}, \ldots, \psi_{i+1,D-1}) \\ &= (\psi_{i,0}, \psi_{i,1}, \ldots, \psi_{i,D-1}).\end{aligned} \tag{6}$$

If the received feedback signal f_i = ACK, ψ_{i+1}^t will be updated as follows:

$$\begin{aligned}\psi_{i+1}^t &= (\psi_{i+1,0}, \ldots, \psi_{i+1,D-2}, \psi_{i+1,D-1}) \\ &= (\psi_{i,1}, \ldots, \psi_{i,D-1}, m_i).\end{aligned} \tag{7}$$

As for Bob, if he recovers the source message successfully, he will also update the set ψ_{i+1}^r in the same way and send a feedback signal f_i = ACK. If he fails, he will keep $\psi_{i+1}^r = \psi_i^r$ and send a feedback signal f_i = NACK. This strategy guarantees that $\psi_i^t = \psi_i^r = \psi_i$.

Because there are totally D elements in ψ_i and the length of each element is s bits, the space complexity of storing ψ_i is $O(Ds)$. The update of ψ_i is similar to that of a queue. In the update process, the first element in ψ_i will be removed and discarded. The second element in ψ_i will be moved to

the first location and so on. As for the new element, that is, the source message that has been successfully transmitted, it will be moved to the last location. Considering that the length of each element is s bits, only additional s bits of space are needed to store the element that is being moved. Therefore, the space complexity of updating ψ_i is $O(s)$.

It is very difficult for Eve to reproduce ψ_i. She must eavesdrop on not only every source message, but also all of the feedback signals. Whenever the eavesdropper has uncertainty about ψ_i, the uncertainty is reflected in the corresponding secret key.

3.2. Secret Key Distillation. In this subsection, we will introduce how to distill a secret key from the source message set ψ_i. Our target is retaining as much of the eavesdropper's information loss as possible in the secret key. The theory of universal hash family (UHF) provides a powerful solution for us. A UHF is a family of functions such that the random mapping obtained by uniformly choosing a function from this family is almost invertible [23]. In other words, regardless of the actual input distribution, by uniform-randomly choosing a function from a universal hash family, the expected hash output distribution will be close to the uniform. In our considered scenario, ψ_i is hashed into a secret key K_i by using a function f_{key} that is selected from the universal hash function families F. And the conditional distribution of K_i given the eavesdropper's knowledge about ψ_i can be close to the uniform distribution. Because a nearly uniform distribution means nearly maximum entropy, the eavesdropper knows almost nothing about K_i. Based on the generalized result from [24], the security of K_i can be evaluated by

$$H\left(K_i \mid F, E_i = e_i\right) \geq H_2\left(K_i \mid F, E_i = e_i\right) \tag{8}$$

$$\geq l - \log_2\left(1 + 2^{l-l_r}\right) \tag{9}$$

$$\geq l - \frac{2^{l-l_r}}{\ln 2}, \tag{10}$$

where $E_i = e_i$ is the eavesdropper's knowledge about ψ_i, l is the length of K_i in bits, and $H_2(\cdot)$ is the Renyi entropy of order 2 [24]. When the probability that $E_i = e_i$ is at least $(1 - \delta)$, formula (9) can be generalized as

$$H\left(K_i \mid F, E_i\right) \geq (1 - \delta)\left(l - \log_2\left(1 + 2^{l-l_r}\right)\right). \tag{11}$$

Formulas (9) and (11) show that if the length of the secret key does not exceed l_r, K_i is secure because averagely Eve will have less than one-bit information about K_i. And l_r can be estimated as follows:

$$l_r \leq H_2\left(\psi_i \mid E_i = e_i\right). \tag{12}$$

It is noteworthy that (9) and (11) are averaged over all uniformly choices of hash functions. It is possible that, for some specific values of F, $H(K_i \mid F, E_i)$ is not negligible when $l \leq l_r$. However, it appears with negligible probability [24].

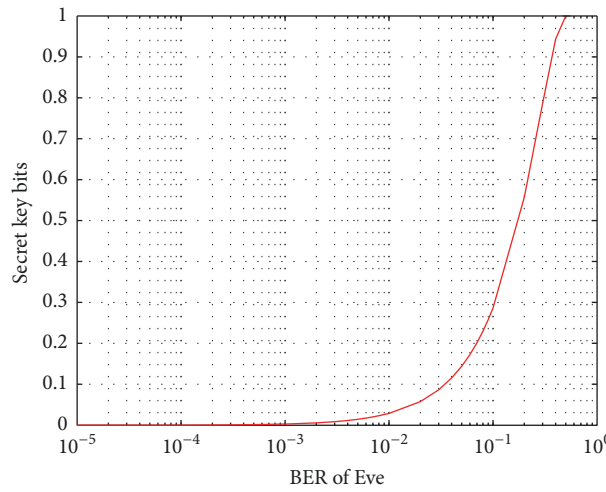


Figure 6: The number of secret key bits we can distill from each source message bit versus Eve's BER.

Because of the randomness of the wireless channel, it is impossible for Eve to recover each source message in ψ_i. $H_2(\psi_i \mid E_i = e_i)$ can be calculated as follows [24]:

$$H_2\left(\psi_i \mid E_i = e_i\right) = -d \cdot \log_2\left(\left(1 - \left(P_e^E\right)\right)^2 + \left(P_e^E\right)^2\right), \tag{13}$$

where $d = Ds$ is the length of ψ_i in bits.

Figure 6 illustrates the relationship between Eve's BER and the number of secret key bits we can distill from each source message bit. We can see that, with the increase of Eve's BER, we can distill more secret key bits averagely from each source message bit. It shows that the eavesdropper's information loss is retained in the secret key.

In our considered wiretap channel model, Eve's BER can be calculated by Bob's maximum BER and the security gap:

$$\begin{aligned} P_e^E &= P_e\left(\text{SNR}_E\right) = P_e\left(\text{SNR}_B - \text{Sg}\right) \\ &= P_e\left(P_e^{-1}\left(P_e^B\right) - \text{Sg}\right). \end{aligned} \tag{14}$$

Then, we can calculate l_r as follows:

$$l_r = -d \cdot \log_2\left(\left(1 - P_e^E\right)^2 + \left(P_e^E\right)^2\right) \tag{15}$$

$$\begin{aligned} &= -d \cdot \log_2\left[\left(1 - P_e\left(P_e^{-1}\left(P_e^B\right) - \text{Sg}\right)\right)^2\right. \\ &\quad \left. + \left(P_e\left(P_e^{-1}\left(P_e^B\right) - \text{Sg}\right)\right)^2\right]. \end{aligned} \tag{16}$$

Considering that $d = Ds$, according to (16), we can choose the value of D as follows:

$$D = \frac{-l_r}{\log_2\left[\left(1 - P_e\left(P_e^{-1}\left(P_e^B\right) - \text{Sg}\right)\right)^2 + \left(P_e\left(P_e^{-1}\left(P_e^B\right) - \text{Sg}\right)\right)^2\right] s}. \tag{17}$$

3.3. Implementation of UHF. In this subsection, we will show how to implement the universal hash function $f_{\text{key}}(\cdot)$ in

practical scenarios. A Toeplitz matrix is a matrix in which each descending diagonal from left to right is constant and is a kind of UHF that can be implemented with low complexity [25]. In our proposed scheme, we try to generate secret key K_i with length l from the source message set ψ_i with length d. The corresponding Toeplitz matrix is as follows:

$$\mathbf{T} = \begin{bmatrix} t_l & t_{l+1} & \cdots & t_{l+d-2} & t_{l+d-1} \\ t_{l-1} & t_l & \cdots & t_{l+d-3} & t_{l+d-2} \\ \vdots & t_{l-1} & \ddots & \vdots & \vdots \\ t_2 & \vdots & \ddots & t_d & t_{d+1} \\ t_1 & t_2 & \cdots & t_{d-1} & t_d \end{bmatrix}, \tag{18}$$

where $t_1, t_2, \ldots, t_l, \ldots, t_{l+d-1}$ is the randomly generated element over GF(2). The secret key can be generated by multiplying $\mathbf{T}$ and ψ:

$$K_i = f_{\text{key}}(\psi_i) = \mathbf{T} \times (\psi_i)^T. \tag{19}$$

The computational complexity of (19) is $O(d^2)$. To reduce the computational complexity, we can use the improved algorithm based on fast Fourier transformation (FFT) [26]. Based on the Toeplitz matrix $\mathbf{T}$, we can obtain a new circular matrix $\mathbf{T}_C$ as follows:

$$\begin{aligned} \mathbf{T}_C &= \begin{bmatrix} \mathbf{T} & \mathbf{R}_1 \\ \mathbf{R}_2 & \mathbf{R}_3 \end{bmatrix} \\ &= \text{Circu}\left(t_l, t_{l+1}, \ldots, t_{l+d-1}, t_1, \ldots, t_{l-1}\right), \end{aligned} \tag{20}$$

where $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{R}_3$ are the submatrices defined in [26], which make the extended matrix $\mathbf{T}_C$ a circular matrix. Circu$(\cdot)$ denotes the circular matrix, which can be represented by its first row.

Then, we generate a new vector $\widehat{\psi}_i = (\psi_i, \mathbf{0})$ by combining ψ_i with a zero vector $\mathbf{0}$, where the length of $\widehat{\psi}_i$ equals the columns of $\mathbf{T}_C$. The secret key can be generated by multiplying $\mathbf{T}_C$ and $\widehat{\psi}_i$, which can be calculated using the FFT-based method:

$$K_i = \mathbf{T}_C \times \widehat{\psi}_i^T = \mathscr{F}^{-1}\left(\mathscr{F}\left(\mathbf{T}_C(1)\right) \circ \mathscr{F}\left(\widehat{\psi}_i\right)\right), \tag{21}$$

where $\mathscr{F}(\cdot)$ is the Fourier transform and $\mathscr{F}^{-1}(\cdot)$ is the inverse, $\mathbf{T}_C(1)$ is the first row of $\mathbf{T}_C$, and $\circ$ denotes the operation that multiplies the corresponding elements in the vector. The computational complexity of (21) is $O(d \log d)$.

4. Design of Structure-Random LDPC Codes

In this section, we will show how to construct a large number of parity-check matrices of LDPC codes based on the technique we called structured-random protograph expanding. A protograph is a Tanner graph with a relatively small number of nodes [27], which can be used to construct the parity-check matrix of LDPC codes. Because systematic codes directly expose the secret message bits, all of the s information bits will be punctured and the n parity bits will be transmitted.

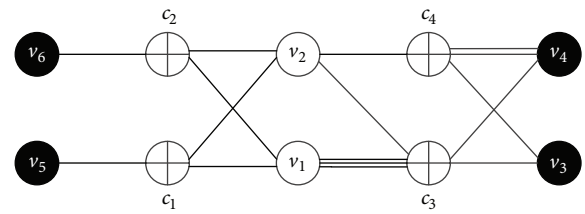


Figure 7: Protograph P for the nonsystematic LDPC code. The rate is 1/2.

We use the code doping method in [28] to design and optimize our protograph to ensure that the iterative decoding of the designed LDPC codes can be triggered successively. Figure 7 shows our optimized protograph $P = (V, C, E)$ for a rate-1/2 nonsystematic LDPC code. We denote V as the set of variable nodes $\{v_1, v_2, \ldots, v_6\}$, C as the set of check nodes $\{c_1, c_2, \ldots, c_4\}$, and E as the set of edges $\{e_1, e_2, \ldots, e_{16}\}$. In the designed protograph, we will puncture the information nodes denoted by v_1 and v_2 among all the variable nodes to avoid systematic transmission.

To guarantee the convergence of the brief propagation (BP) decoding algorithm, the connection relationship of the check node c_4 is specially designed. In our designed protograph, the check node c_4 is connected to only one punctured variable node v_2. Equivalently, we can use a base parity-check matrix $\mathbf{H}_{B,0}$ with size 4×6 to represent this protograph.

$$\mathbf{H}_{B,0} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 3 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 2 & 0 & 0 \end{bmatrix}. \tag{22}$$

A "copy-and-permute" operation can be applied to the protograph P to obtain a large derived Tanner graph. We define T as the expanded factor; the "copy-and-permute" operation firstly makes T copies of the protograph P and then permutes the endpoints of each edge among the T variable nodes and T check nodes connected to the set of T edges copied from the same edge from the original protograph P.

After this operation, we can obtain a large Tanner graph, where the T copies of the original protograph are connected to each other. Equivalently, we can expand each element of value w in the base matrix $\mathbf{H}_{B,0}$ to a $T \times T$ matrix with w ones in each row or column. As a result, we can obtain a large matrix with size $4T \times 6T$.

Because random permutation is not easy to describe and implement efficiently, in our scheme, we adopt the structured type of permutation, such as cyclic permutation. In other words, we expand each element of value w in the base matrix $\mathbf{H}_{B,0}$ to $T \times T$ circulant permutation matrices $\mathbf{I}_T(t)$. As a result, the expanded parity-check matrix will become a T-circulant matrix.

To construct a large number of parity-check matrices of LDPC codes, it is not enough to expand the protograph P with just one single stage. Therefore, we develop a structured-random protograph expanding technique. This technique expands the protograph P with $L > 1$ stages. We denote $T_1, T_2, \ldots, T_L$ as the expanding factors for stages $1, 2, \ldots, L$, respectively. The total expanding factor T can be calculated as $T = T_1 T_2 \cdots T_L$. Finally, the base matrix $\mathbf{H}_{B,0}$ is expanded to the parity-check matrix $\mathbf{H}_{B,L}$.

(i) *Structured expanding*: in the procedure of structured expanding, we expand the protograph P in the first $L-1$ stages to avoid parallel edges, short cycles, and low-weight codewords. As a result, all the nonzero elements in $\mathbf{H}_{B,L-1}$ will be equal to 1.

(ii) *Random expanding*: in the procedure of random expanding, we expand $\mathbf{H}_{B,L-1}$ in the L stage based on the value of the dynamic secret key K_i. For each zero element in $\mathbf{H}_{B,L-1}$, we will expand it by a $T_L \times T_L$ zero matrix $\mathbf{0}_{T_L \times T_L}$. For each nonzero element, we will expand it by a $T_L \times T_L$ circulant permutation matrix $\mathbf{I}_{T_L}(t)$. The total number of zero and nonzero elements is $J = |E|T/T_L$.

As for the parameters that are used in the procedure of structured expanding, that is, all the shift values and expanding factors, they are constant and will be shared between Alice and Bob publicly in advance. Now, we rewrite the dynamic secret key K_i as a binary vector:

$$K_i = \left(k_{i,0}, k_{i,1}, \ldots, k_{i,j}, \ldots, k_{i,J-1}\right), \tag{23}$$

where each element $k_{i,j} \in [\![0, T_L - 1]\!]$ is represented by $\log_2 T_L$ bits. Regarding the parameters that are used in the procedure of random expanding, that is, all the random shift values, they are controlled by the dynamic secret key K_i, whose length is required to be $l = J \log_2 T_L$ bits.

After expanding the protograph P with $L > 1$ stages, the base matrix $\mathbf{H}_{B,0}$ is expanded to an $n \times (n+s)$ parity-check matrix $\mathbf{H}_i$, where $n = 4T$ and $s = 2T$. As mentioned above, $\mathbf{H}_i$ is a T_L-circulant matrix and can be written as $\mathbf{H}_i = [\mathbf{A}(K_i), \mathbf{B}(K_i)]$ such that

$$\mathbf{A}\left(K_i\right) = \left[\mathbf{A}^w_{\alpha\beta}\right]_{2\times 4} = \begin{bmatrix} \mathbf{A}^1_{11} & \mathbf{A}^1_{12} \\ \mathbf{A}^1_{21} & \mathbf{A}^1_{22} \\ \mathbf{A}^3_{31} & \mathbf{A}^1_{32} \\ \mathbf{0} & \mathbf{A}^1_{42} \end{bmatrix},$$
$$\mathbf{B}\left(K_i\right) = \left[\mathbf{B}^w_{\alpha\beta}\right]_{4\times 4} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{B}^1_{13} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}^1_{24} \\ \mathbf{B}^1_{31} & \mathbf{B}^1_{32} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}^1_{41} & \mathbf{B}^2_{42} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{24}$$

The first $s = 2T$ nodes are punctured as information nodes among all the $n + s = 6T$ variable nodes.

4.1. An Example. In this subsection, we construct a large number of nonsystematic (2048, 1024) LDPC codes via $L = 3$ stages. The total expanding factor $T = T_1 T_2 T_3 = 4 \times 4 \times 32 = 512$. With the factor $T_1 = 4$, the first stage aims to separate all the parallel edges. With the factor $T_2 = 4$, the second stage aims to avoid the existence of the cycle of girth 4. With the factor $T_3 = 32$, the third stage aims to randomly expand all the $|E|T/T_3 = 256$ edges. Finally, we get a set of parity-check matrices $\mathscr{H} = \{\mathbf{H}(\mathbf{r}) : \mathbf{r} \in [\![0, 2^{256} - 1]\!]\}$.

During the transmission, we randomly select a parity-check matrix for each source message. The number of iterations is restricted by 63. In Figure 8, we show the average BER of the structured-random nonsystematic (2048, 1024) LDPC codes with different number of retransmissions r.

5. Encoder and Decoder Implementation of Structured-Random LDPC Codes

5.1. Encoder Implementation. To implement the encoder of structured-random LDPC codes, we need to derive the $s \times n$ nonsystematic generator matrix $\mathbf{G}_i$ according to the parity-check matrix $\mathbf{H}_i$. According to (24), $\mathbf{G}_i$ can be derived by

$$\mathbf{G}_i = \left(\mathbf{B}\left(K_i\right)^{-1} \cdot \mathbf{A}\left(K_i\right)\right)^T = \begin{bmatrix} (\mathbf{G}_i)_{11} & (\mathbf{G}_i)_{12} & (\mathbf{G}_i)_{13} & (\mathbf{G}_i)_{14} \\ (\mathbf{G}_i)_{21} & (\mathbf{G}_i)_{22} & (\mathbf{G}_i)_{23} & (\mathbf{G}_i)_{24} \end{bmatrix}$$
$$= \begin{bmatrix} \left(\mathbf{A}^3_{31}\right)^T \mathbf{D}_1 & \left(\mathbf{A}^3_{31}\right)^T \mathbf{D}_3 & \left(\mathbf{A}^1_{11}\right)^T \mathbf{B}_{13} & \left(\mathbf{A}^1_{21}\right)^T \mathbf{B}_{24} \\ \left(\mathbf{A}^1_{32}\right)^T \mathbf{D}_1 \oplus \left(\mathbf{A}^1_{42}\right)^T \mathbf{D}_2 & \left(\mathbf{A}^1_{32}\right)^T \mathbf{D}_3 \oplus \left(\mathbf{A}^1_{42}\right)^T \mathbf{C}^T & \left(\mathbf{A}^1_{12}\right)^T \mathbf{B}_{13} & \left(\mathbf{A}^1_{22}\right)^T \mathbf{B}_{13} \end{bmatrix}, \tag{25}$$

where $\mathbf{D}_1 = (\mathbf{I} \oplus \mathbf{B}^1_{31}(\mathbf{B}^1_{41})^T \mathbf{C}^T (\mathbf{B}^1_{32})^T)\mathbf{B}^1_{31}$, $\mathbf{D}_2 = \mathbf{C}^T(\mathbf{B}^1_{32})^T \mathbf{B}^1_{31}$, $\mathbf{D}_3 = \mathbf{B}^1_{31}(\mathbf{B}^1_{41})^T \mathbf{C}^T$, and $\mathbf{C} = ((\mathbf{B}^2_{42})^T \oplus (\mathbf{B}^1_{32})^T \mathbf{B}^1_{31}(\mathbf{B}^1_{41})^T)^{-1}$.

The multiplication between m_i and $\mathbf{G}_i$ can be calculated in blocks:

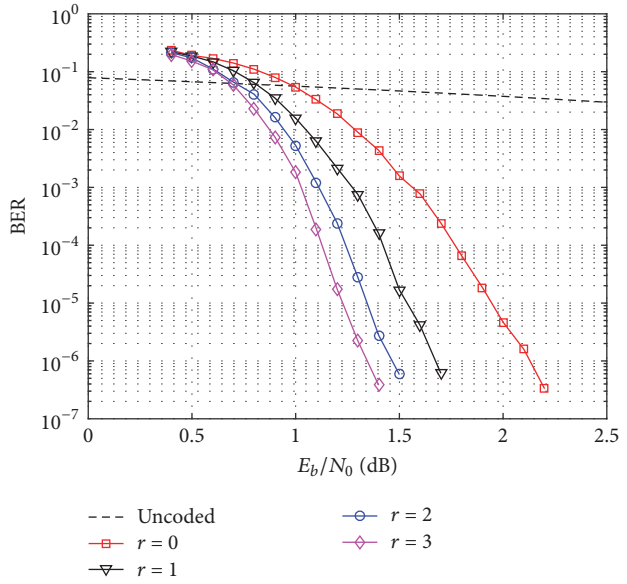


FIGURE 8: The average BER of the nonsystematic (2048, 1024) LDPC codes with different maximum retransmission number r.

$$m_i \cdot \mathbf{G}_i = \left[(m_i)_{11} \;\; (m_i)_{12}\right] \times \begin{bmatrix} (\mathbf{G}_i)_{11} & (\mathbf{G}_i)_{12} & (\mathbf{G}_i)_{13} & (\mathbf{G}_i)_{14} \\ (\mathbf{G}_i)_{21} & (\mathbf{G}_i)_{22} & (\mathbf{G}_i)_{23} & (\mathbf{G}_i)_{24} \end{bmatrix} = \left[(x_i)_{11} \;\; (x_i)_{12} \;\; (x_i)_{13} \;\; (x_i)_{14}\right], \tag{26}$$

where $(x_i)_{11} = (m_i)_{11}(\mathbf{G}_i)_{11} + (m_i)_{12}(\mathbf{G}_i)_{21}$, $(x_i)_{12} = (m_i)_{11}(\mathbf{G}_i)_{12} + (m_i)_{12}(\mathbf{G}_i)_{22}$, $(x_i)_{13} = (m_i)_{11}(\mathbf{G}_i)_{13} + (m_i)_{12}(\mathbf{G}_i)_{23}$, and $(x_i)_{14} = (m_i)_{11}(\mathbf{G}_i)_{14} + (m_i)_{12}(\mathbf{G}_i)_{24}$. The multiplication between $(m_i)_{ab}$ and $(\mathbf{G}_i)_{\acute{a}\acute{b}}$ can be further divided as in [29]. For example, to multiply by $(\mathbf{G}_i)_{12}$ can be divided into four steps by successively multiplying by $(\mathbf{A}_{31}^3)^T$, $\mathbf{B}_{31}^1$, $(\mathbf{B}_{41}^1)^T$, and $\mathbf{C}^T$. Because all those submatrices are circulant, all the required multiplications in the encoding process can be finished in $O(n)$ time. The additional computational complexity is from the inversion operation to derive $\mathbf{C}$. In [30], authors have shown the inversion of a binary matrix can be finished in $O(n)$ time by using a parallel hardware architecture. Therefore, the encoding process can be finished in $O(n)$ time.

Consider that the size of $\mathbf{C}$ is about 1/8 of the size of $\mathbf{G}_i$. Thus, the designed encoder for structured-random LDPC codes will increase by 1/8 of the storage compared to the traditional encoder for QC-LDPC codes with a fixed parity-check matrix [31, 32].

5.2. Decoder Implementation. As for the decoder of struc\-tured-random LDPC codes, it can be extended from the conventional decoder of quasi-cyclic LDPC codes with a fixed parity-check matrix [33, 34]. This is because the parity-check matrix $\mathbf{H}_i$ of structured-random LDPC codes is also quasi-cyclic as shown in Section 4. The only difference is that the shift values of the circulant permutation matrices in $\mathbf{H}_i$ will be updated according to the dynamic secret key K_i. When the shift values are successfully updated, the iterative decoding process is the same. Therefore, the decoder implementation complexity of structured-random LDPC codes will be the same as that of quasi-cyclic LDPC codes with a fixed parity-check matrix.

6. Performance Analysis

In this section, we will analyze the security and reliability performances of our proposed scheme. As shown in the previous section, we can construct a large number of nonsystematic LDPC codes that have good error-correction performance. Therefore, we can guarantee that Bob's BER P_e^B will be lower than the given threshold by utilizing these nonsystematic LDPC codes. It guarantees the reliability of the transmission. We will analyze the security of our scheme in two aspects: the complexity when Eve tries to crack the dynamic secret key and Eve's average BER during the whole transmission.

Different from the traditional cryptosystems that have to distribute the secret key before communication begins, our scheme generates the secret key K_i dynamically from the source message set ψ_i. During the transmission, an event which is referred to as synchronization error may happen. That is, there exists an index $i_{\mathrm{TH}} \in \mathbb{N}$, such that $z_{i_{\mathrm{TH}}}$ is not correctly decoded by Eve, but $y_{i_{\mathrm{TH}}}$ is successfully recovered by Bob. At this moment, Eve's source message set $\overline{\psi}_{i+1}$ will be different from Alice's and Bob's source message set ψ_{i+1}. Therefore, Eve cannot generate the same secret key as Alice and Bob.

As analyzed in Section 3, universal hash function makes the conditional distribution of K_i close to the uniform distribution as follows:

$$P\left(K_i = k_i \mid E_i = e_i\right) \approx \frac{1}{\left|K_i\right|}, \quad \forall k_i \in K_i. \tag{27}$$

From the information theoretic perspective, (27) means that the conditional entropy of K_i is close to its self-information

$$H\left(K_i \mid E_i = e_i\right) \approx H\left(K_i\right) = \log_2 \left|K_i\right|. \tag{28}$$

Therefore, the computational complexity of Eve to crack a dynamic secret key is approximated to $2^{|K_i|} = 2^l$. Even if Eve cracks the secret key by the exhaustive search, the similar synchronization error may happen again and she has to repeat the cracking process.

To evaluate the probability that the synchronization error happens, we denote $P_f(\cdot)$ as the frame error rate (FER) as the function of SNR. Bob's FER and Eve's FER can be expressed as $P_f^B = P_f(\mathrm{SNR}_B)$ and $P_f^E = P_f(\mathrm{SNR}_E)$, respectively. And i_{TH} is distributed geometrically; $i_{\mathrm{TH}} \sim G(p_0)$, where $p_0 = (1 - P_f^B)P_f^E$. Thus, the probability distribution of i_{TH} can be calculated as

$$\Pr\left(i_{\mathrm{TH}} = i\right) = \left[1 - \left(1 - P_f^B\right)P_f^E\right]^{i-1}\left(1 - P_f^B\right)P_f^E. \tag{29}$$

As analyzed above, it is difficult for Eve to generate the same secret key as Alice and Bob once the synchronization error happens. In other words, Eve cannot generate the

correct parity-check matrix to decode $z_{i_{\mathrm{TH}}}$. To evaluate Eve's BER during the whole transmission, we can divide the source messages that Eve fails to recover into two categories. The first category contains the source messages that Eve fails to recover before the synchronization error happens. For the messages in the first category, they are recovered by Eve using the correct parity-check matrix. The number of messages in the first category $N_1(i_{\mathrm{TH}})$ obeys the binomial distribution, $N_1(i_{\mathrm{TH}}) \sim B(i_{\mathrm{TH}}-1, p_1)$, where $p_1 = P_f^B P_f^E/(1-(1-P_f^B)P_f^E)$. Thus, the average of $N_1(i_{\mathrm{TH}})$ can be calculated as

$$\overline{N}_1(i_{\mathrm{TH}}) = (i_{\mathrm{TH}})\, p_1 = \frac{(i_{\mathrm{TH}}-1)\, P_f^B P_f^E}{1-\left(1-P_f^B\right)P_f^E}. \tag{30}$$

And the average number of error bits in each error message can be calculated as

$$k_{\mathrm{ER}} = \frac{P_e(\mathrm{SNR}_E)\cdot s}{P_f(\mathrm{SNR}_E)}. \tag{31}$$

For the messages in the second category, half of their bits are wrong, because Eve cannot generate the correct parity-check matrix as Alice and Bob. Finally, Eve's BER can be calculated as

$$\begin{aligned} P_e^E = &\sum_{i=1}^{N} \frac{k_{\mathrm{ER}}\cdot\overline{N}_1(i) + 0.5s\cdot(N-i+1)}{k\cdot N}\cdot \Pr\left(i_{\mathrm{TH}}=i\right) \\ &+ \frac{k_{\mathrm{ER}}\cdot\overline{N}_1(N+1)}{k\cdot N}\cdot\Pr\left(i_{\mathrm{TH}}\geq N+1\right). \end{aligned} \tag{32}$$

Based on (30), P_e^E can be further calculated as

$$\begin{aligned} P_e^E &= \sum_{i=1}^{N}\frac{k_{\mathrm{ER}}\cdot(i-1)\cdot p_1 + 0.5s\cdot(N-i+1)}{s\cdot N} \\ &\quad\cdot\Pr\left(i_{\mathrm{TH}}=i\right) + \frac{k_{\mathrm{ER}}\cdot N\cdot p_1}{s\cdot N}\cdot\Pr\left(i_{\mathrm{TH}}\geq N+1\right) \\ &\geq \sum_{i=1}^{N}\frac{k_{\mathrm{ER}}\cdot(i-1)\cdot p_1 + 0.5s\cdot(N-i+1)}{s\cdot N} \\ &\quad\cdot\Pr\left(i_{\mathrm{TH}}=i\right) \\ &= \frac{0.5s\cdot N\sum_{i=1}^{N}\Pr\left(i_{\mathrm{TH}}=i\right)}{s\cdot N} \\ &\quad+\frac{\left(k_{\mathrm{ER}}\cdot p_1 - 0.5s\right)\sum_{i=1}^{N}(i-1)\cdot\Pr\left(i_{\mathrm{TH}}=i\right)}{s\cdot N} \\ &\geq 0.5\cdot\left(1-\Pr\left(i_{\mathrm{TH}}\geq N+1\right)\right) \\ &\quad-\frac{0.5\cdot\sum_{i=1}^{\infty} i\cdot\Pr\left(i_{\mathrm{TH}}=i\right)}{N} \\ &= 0.5\cdot\left(1-\Pr\left(i_{\mathrm{TH}}\geq N+1\right)\right) - 0.5\cdot\frac{\bar{i}_{\mathrm{TH}}}{N} \\ &= 0.5\cdot\left(1-(1-p_0)^N - \frac{\bar{i}_{\mathrm{TH}}}{N}\right), \end{aligned} \tag{33}$$

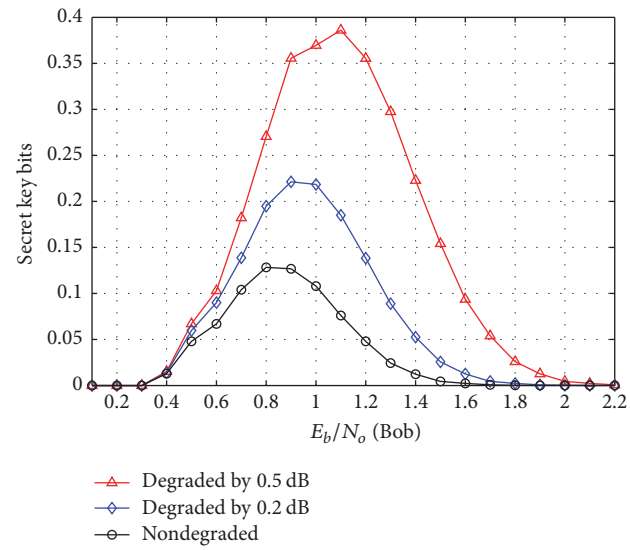


Figure 9: The number of secret key bits we can distill averagely from each transmitted source message bit using (2048, 1024) nonsystematic LDPC codes.

where $\bar{i}_{\mathrm{TH}}$ is defined as follows:

$$\bar{i}_{\mathrm{TH}} = \sum_{i=1}^{\infty} i\cdot\Pr\left(i_{\mathrm{TH}}=i\right) = \frac{1}{p_0} = \frac{1}{\left(1-p_f^B\right)p_f^E}. \tag{34}$$

Finally, P_e^E can be lower bounded as follows:

$$P_e^E \geq 0.5\cdot\left(1-(1-p_0)^N - \frac{1}{N\cdot p_0}\right) \xrightarrow{N\to\infty} 0.5. \tag{35}$$

From the above analysis, we can know that Eve's BER P_e^E will approach 0.5 when the number of the transmitted messages goes to infinity. In addition, when the security gap of the system increases, $(1-P_f^B)$ and P_f^E will increase, and thus p_0 will increase. Therefore, we can make Eve's BER P_e^E approach 0.5 with faster speed by increasing the security gap of the system.

7. Simulation Results

In this section, we will evaluate the performance of our proposed scheme by Monte-Carlo simulations.

Figure 9 illustrates the number of secret key bits we can distill averagely from each transmitted source message bit. In the region with very low or very high E_b/N_0, we can see that the number of secret key bits decreases. The reasons are as follows: in the region with very low E_b/N_0, the retransmission happens frequently and the proportion of un-retransmitted source messages is small; in the region with very high E_b/N_0, the BER of Eve is very low and therefore the number of secret key bits we can distill averagely from each source message bit is small. In addition, we can see that the more the channel of Eve is degraded compared to that of Bob, the more secret key bits we can distill.

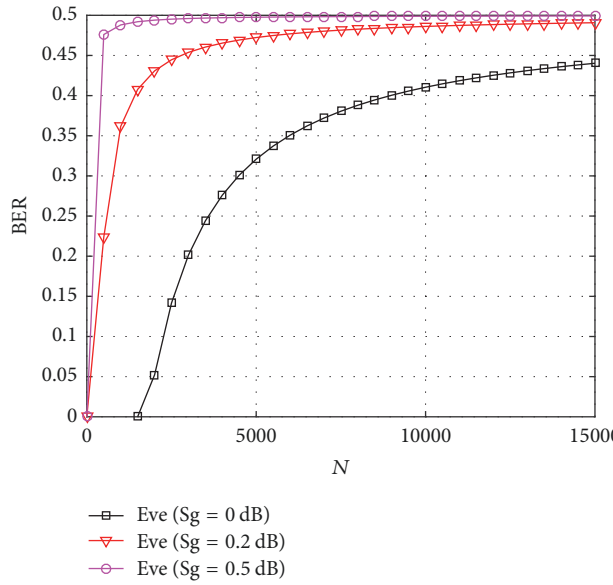


Figure 10: The BER of eavesdropper versus the number of transmitted source messages when $N = 1, 2, \ldots, 15000$.

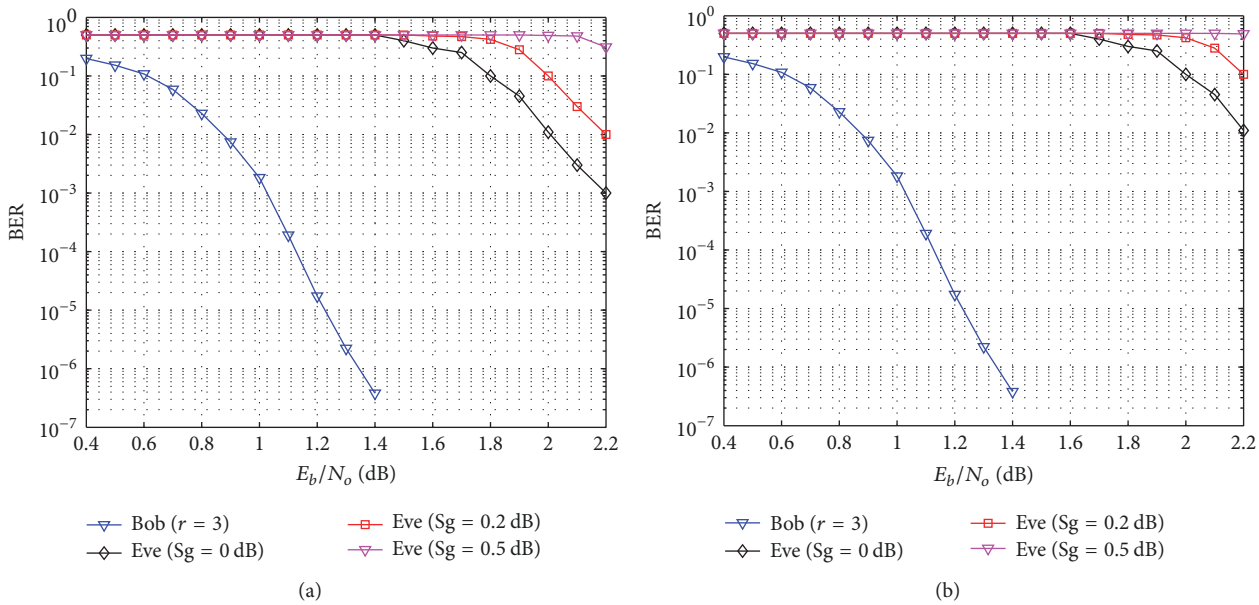


Figure 11: BER of our framework for Gaussian wiretap channel using (2048, 1024) nonsystematic LDPC codes for different security gaps when $N = 1000$ and $N = 10000$ source message are transmitted with maximum retransmission number $r = 3$.

In Figure 10, the BER of Eve versus the number of transmitted source messages N for different security gaps S_g is plotted. We can see from Figure 8 that Bob's BER will be lower than 10^{-6} in four conditions: $r = 0$ and $E_b/N_0 = 2.2$ dB, $r = 1$ and $E_b/N_0 = 1.7$ dB, $r = 2$ and $E_b/N_0 = 1.5$ dB, or $r = 3$ and $E_b/N_0 = 1.4$ dB. Therefore, to guarantee the reliability of the transmission ($P^B_{e,\max} < 10^{-6}$), the quality of the main channel can be fixed to $E_b/N_0 = 1.7$ dB and the maximum transmission number can be fixed to $r = 1$. For different security gaps, we can see that the BER of Eve will always approach 0.5 as the number of transmitted source messages increases. This is owing to the fact that the secret keys generated by Eve are the same as the keys generated by Alice and Bob before the first synchronization error happens. Therefore, she can recover the corresponding source messages successfully. After the first synchronization error happens, Eve can no longer decode the following source messages anymore, because the uncorrected decoding errors prevent her from generating the correct secret key. Thus, as the number of transmitted source messages increases, Eve's average BER will approach 0.5.

In Figure 11(a), the BER curves of Bob and Eve are plotted when $N = 1000$ for different security gaps S_g. The maximum retransmission number is fixed to $r = 3$. If Bob's BER

threshold is set to $P^{B}_{e,\max} < 10^{-6}$ and Eve's BER threshold is set to $P^{E}_{e,\min} = 0.49$, the security gap $S_g = 0$ dB can be achieved. In Figure 11(b), the BER curves of Bob and Eve are plotted when $N = 10000$. We can see that security gap S_g can be further reduced to lower than 0 dB. It means that security of the source message can be guaranteed even when the wiretap channel is better than the main channel. We can see that the security gap performance of our scheme is really small and can be improved by increasing N.

8. Conclusions

In this paper, we have proposed a secrecy transmission scheme based on code-hopping to encrypt and encode the source messages at the physical layer for wireless communications. First, we present a dynamic secret key generation algorithm based on ARQ mechanism. With this algorithm, Alice and Bob can distill the secret keys from the unretransmitted source messages based on the universal hash families. Second, we present a structured-random LDPC codes design algorithm. Based on this algorithm, we generate a large amount of parity-check matrices of LDPC codes. During the transmission, Alice and Bob dynamically select the parity-check matrices of LDPC codes to encode and recover the source messages based on the dynamic secret keys. Theoretical analysis demonstrates that it is difficult for Eve to generate the same secret key as Alice and Bob. Simulation results show that the BER of Eve will approach 0.5 as the number of transmitted source messages increases and the security gap of our system is small.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC, 91538203), the New Strategic Industries Development Projects of Shenzhen City (JCYJ20150403155812833), and the Joint Research Foundation of the General Armaments Department and the Ministry of Education of China (6141A02033322).

References

[1] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Technical Journal*, vol. 28, pp. 656–715, 1949.

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

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

[4] S. K. Leung-Yan-Cheong, "On a Special Class of Wiretap Channels," *IEEE Transactions on Information Theory*, vol. 23, no. 5, pp. 625–627, 1977.

[5] P. K. Gopala, L. Lai, and H. El Gamal, "On the secrecy capacity of fading channels," *IEEE Transactions on Information Theory*, vol. 54, no. 10, pp. 4687–4698, 2008.

[6] F. Oggier and B. Hassibi, "The secrecy capacity of the MIMO wiretap channel," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 4961–4972, 2011.

[7] E. Ekrem and S. Ulukus, "The secrecy capacity region of the Gaussian MIMO multi-receiver wiretap channel," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2083–2114, 2011.

[8] A. S. Mansour, R. F. Schaefer, and H. Boche, "The individual secrecy capacity of degraded multi-receiver wiretap broadcast channels," *IEEE International Conference on Communications*, pp. 4181–4186, 2015.

[9] A. Yener and S. Ulukus, "Wireless physical-layer security: lessons learned from information theory," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1814–1825, 2015.

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

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

[12] B. Duo, P. Wang, Y. Li, and B. Vucetic, "Secure transmission for relay-eavesdropper channels using polar coding," *IEEE International Conference on Communications*, pp. 2197–2202, 2014.

[13] H. Si, O. O. Koyluoglu, and S. Vishwanath, "Hierarchical polar coding for achieving secrecy over state-dependent wiretap channels without any instantaneous CSI," *IEEE Transactions on Communications*, vol. 64, no. 9, pp. 3609–3623, 2016.

[14] Y.-P. Wei and S. Ulukus, "Polar Coding for the General Wiretap Channel with Extensions to Multiuser Scenarios," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 278–291, 2016.

[15] Z. Chen, L. Yin, and J. Lu, "Hamming distortion based secrecy systems: To foil the eavesdropper with finite shared key," *IEEE Communications Letters*, vol. 19, no. 5, pp. 711–714, 2015.

[16] P. Wang, L. Yin, and J. Lu, "An efficient helicopter-satellite communication scheme based on check-hybrid ldpc coding," *Tsinghua Science and Technology*, 2018.

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

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

[19] M. Baldi, M. Bianchi, and F. Chiaraluce, "Coding with scrambling, concatenation, and HARQ for the AWGN wire-tap channel: A security gap analysis," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 883–894, 2012.

[20] Z. Chen, L. Yin, Y. Pei, and J. Lu, "CodeHop: physical layer error correction and encryption with LDPC-based code hopping," *Science China Information Sciences*, vol. 59, no. 10, Article ID 102309, pp. 1–15, 2016.

[21] Y. Fang, G. Bi, Y. L. Guan, and F. C. M. Lau, "A survey on protograph LDPC codes and their applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 1989–2016, 2015.

[22] Y. Fang, S. C. Liew, and T. Wang, "Design of distributed protograph LDPC Codes for multi-relay coded-cooperative networks," *IEEE Transactions on Wireless Communications*, pp. 7235–7251, 2017.

[23] H. Tyagi and A. Vardy, "Universal hashing for information-theoretic security," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1781–1795, 2015.

[24] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, "Generalized privacy amplification," *IEEE Transactions on Information Theory*, vol. 41, no. 6, part 2, pp. 1915–1923, 1995.

[25] M. Hayashi, "Exponential decreasing rate of leaked information in universal random privacy amplification," *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3989–4001, 2011.

[26] M. Hayashi and T. Tsurumaru, "More efficient privacy amplification with less random seeds via dual universal hash function," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 2213–2232, 2016.

[27] J. Thorpe, "Low-density parity-check (ldpc) codes constructed from protographs," *IPN progress report*, vol. 42, no. 154, pp. 42–154, 2003.

[28] S. Ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Transactions on Signal Processing*, vol. 51, no. 11, pp. 2764–2772, 2003.

[29] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 638–656, 2001.

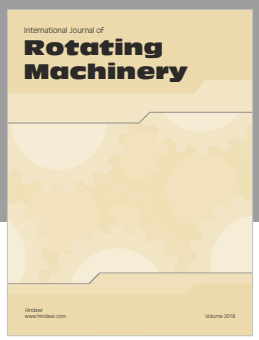
[30] A. Bogdanov, M. C. Mertens, C. Paar, J. Pelzl, and A. Rupp, "Smith-a parallel hardware architecture for fast gaussian elimination over gf (2)," in *Workshop on Special-Purpose Hardware for Attacking Cryptographic Systems*, 2006.

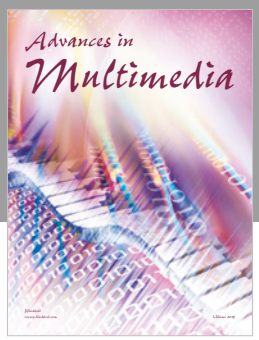
[31] Z. Li, L. Chen, L. Zeng, S. Lin, and W. H. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Transactions on Communications*, vol. 54, no. 1, pp. 71–81, 2006.

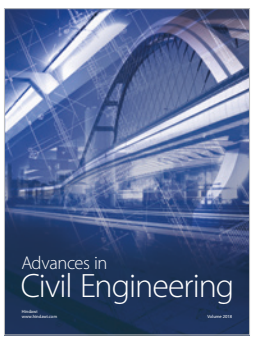
[32] Q. Huang, L. Tang, S. He, Z. Xiong, and Z. Wang, "Low-complexity encoding of quasi-cyclic codes based on Galois Fourier transform," *IEEE Transactions on Communications*, vol. 62, no. 6, pp. 1757–1767, 2014.

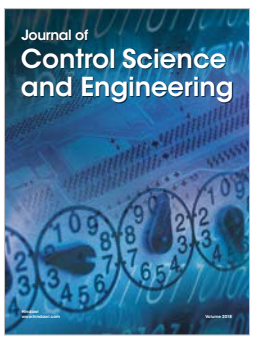
[33] Y.-L. Ueng, B.-J. Yang, C.-J. Yang, H.-C. Lee, and J.-D. Yang, "An efficient multi-standard LDPC decoder design using hardware-friendly shuffled decoding," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 3, pp. 743–756, 2013.

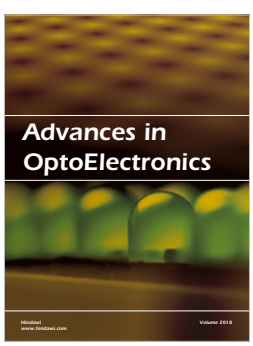
[34] Q. Huang, L. Song, and Z. Wang, "Set message-passing decoding algorithms for regular non-binary LDPC codes," *IEEE Transactions on Communications*, vol. 65, no. 12, pp. 5110–5122, 2017.

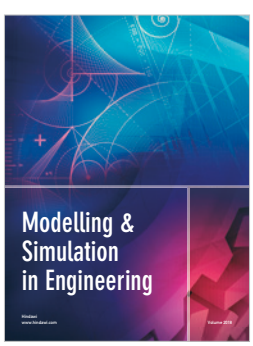


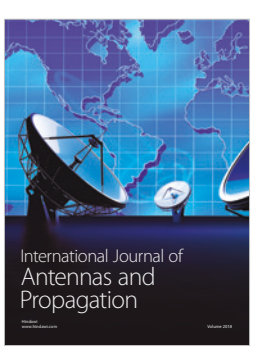


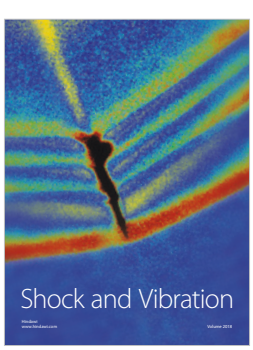


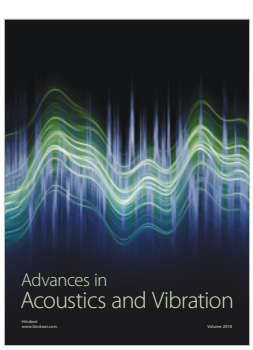












Hindawi

Submit your manuscripts at
www.hindawi.com