Designing Near-Optimal Steganographic Codes in Practice Based on Polar Codes

Weixiang Li, Weiming Zhang, Li Li, Hang Zhou, and Nenghai Yu

***Abstract*— Steganography is an information hiding technique for covert communication. So far Syndrome-Trellis Codes (STC), a convolutional codes-based method, is the only near-optimal coding method, i.e., it can approach the rate-distortion bound of content-adaptive steganography in practice. However, as a secure communication application, steganography needs the diversity of coding methods. This paper proposes another and a better near-optimal steganographic coding method based on polar codes, using Successive Cancellation List (SCL) decoding algorithm to minimize additive distortion in steganography. Considering a steganographic channel as a binary symmetric channel, the proposed Steganographic Polar Codes (SPC) chooses parity-check matrix by setting embedding payload as the initial value of Arikan's heuristic and computes decoding channel metric from the optimal modification probability of minimal distortion model. To overcome the inherent defect of polar codes only suiting for code length of a power of 2, we introduce three strategies to generalize SPC for arbitrary length. Experimental results validate the versatility of SPC to minimize arbitrary distortion. When compared with STC, the overall coding performance of SPC is more superior with low embedding complexity. This work verifies the availability of polar codes for the practical construction of steganographic codes and provides a methodology for designing better steganographic codes based on any advance of polar coding/decoding.**

***Index Terms*— Covert communication, steganography, syndrome coding, polar codes, successive cancellation list.**

I. Introduction

In RECENT years, information hiding techniques have been widely used in the fields of covert communication, copyright protection and content authentication [1]–[5]. Steganography, as a branch of information hiding, aims to embed a covert message in a cover object (e.g., image, audio, video, texts) by slightly changing its original elements without drawing suspicions from steganalysis [6]. Currently, the most effective steganographic schemes are categorized as content-adaptive steganography [7], which usually consists of a heuristically-defined multi-level distortion function and a method for encoding the message to minimize the total distortion. A distortion function is considered *additive* when it is expressed as a sum of individual costs that element-wisely evaluate the effect of independent embedding modifications. Payload-Limited Sender (PLS) and Distortion-Limited Sender (DLS) are two forms for message embedding while minimizing additive distortion. And both of them can be realized in practice using a general methodology called *syndrome coding* [8], which is also called *matrix embedding* because it is realized by using the parity-check matrix of error-correcting codes. In other words, the decoding method of error-correcting codes can be used as the coding method of steganography.

Designing coding methods has always been the core issue in the development of steganography. Matrix embedding was conceptually proposed by Crandall [9] in 1998. For a constant distortion model where all pixels are assumed to have the same impact when changed, various syndrome coding methods based on linear codes, such as Hamming [10], Golay [11], BCH [12], [13], and non-linear codes [14] were proposed to minimize the number of changed pixels. As for an evolutionary wet paper model where all pixels are split into the risky (wet) pixels and safe (dry) pixels, the syndrome coding can also be used in wet paper codes [15]–[19].

The wet paper model is essentially a two-level distortion model only containing constant and infinite costs. But a general distortion model to define multi-level costs is more suitable for multimedia data, because the effects of modifications on different elements are distinguishing in reality. And this is what content-adaptive steganography seeks to withstand steganalysis by confining modifications to the elements with low costs. Modified Matrix Embedding (MME) [20] was proposed to reduce the distortion significantly, but the performance is still far from the rate-distortion bound of general distortion model. Filler *et al.* [8] used linear convolutional codes equipped with Viterbi decoding algorithm and proposed Syndrome-Trellis Codes (STC), which can asymptotically approach the theoretical bound for arbitrary additive distortion function.

STC achieves near-optimal coding performance of content-adaptive steganography because the performance of convolutional codes is close to the channel capacity. Note that polar codes [21] are the first provably channel capacity achieving codes for arbitrary binary-input discrete memoryless channel (B-DMC). A natural idea is to design a better steganographic coding method based on polar codes, hopefully for achieving the bound of embedding efficiency in steganography. Just as pointed out in [8], polar codes are known to be optimal for the

Manuscript received September 26, 2019; revised February 10, 2020; accepted March 12, 2020. Date of publication March 23, 2020; date of current version July 15, 2020. This work was supported in part by the Natural Science Foundation of China under Grant U1636201 and 61572452, by the Anhui Initiative in Quantum Information Technologies under Grant AHY150400, and by the Fundamental Research Funds for the Central Universities under Grant WK6030000135 and WK6030000136. The associate editor coordinating the review of this article and approving it for publication was R. Thobaben. *(Corresponding author: Weiming Zhang.)*

The authors are with the CAS Key Laboratory of Electro-magnetic Space Information, University of Science and Technology of China, Hefei 230026, China (e-mail: zhangwm@ustc.edu.cn).

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

Digital Object Identifier 10.1109/TCOMM.2020.2982624

PLS problem thanks to their capacity-achieving property and the advantage of low complexity of encoding and decoding. On the other hand, designing another kind of steganographic codes can significantly increase the diversity of coding methods in steganography, as STC is currently the only near-optimal coding method for content-adaptive steganographic schemes [22]–[24]. Since steganography is a secure communication application, the unicity of coding method is potentially dangerous to the development of steganography. Therefore, polar codes are the optimum candidate for constructing another and a better near-optimal steganographic coding method in practice.

To design a steganographic coding method based on error-correcting codes, two critical problems have to be solved: how to choose the parity-check matrix and how to incorporate the steganographic distortion into the decoding algorithm to minimize distortion. According to the characteristics of polar coding and decoding, the two problems become: 1) how to choose the frozen indices of polar codes for constructing the parity-check matrix and 2) how to calculate the initial channel metrics needed for polar decoding, on the basic of the steganographic embedding payload and distortion function. In addition, polar codes are inherently designed for binary codes and length of a power of 2, while a steganographic coding method should be applicable to various embedding amplitudes and arbitrary cover length. Thus 3) how to extend binary embedding to q-ary embedding operation and 4) how to deal with arbitrary cover length is another two key problems for designing a practical steganographic coding method.

Polar codes were first used in steganography by Diouf *et al.* [25] who introduced a coding method using Successive Cancellation (SC) decoding algorithm to minimize the embedding impact. However in [25], the solutions to the first two key problems neglected the impact of the embedding payload so that cannot produce a satisfactory coding performance. Besides, the other two problems regarding non-binary embedding and arbitrary cover length were not investigated in [25]. In contrast to [25], this paper tactfully deals with all these four problems, and employs the superior and flexible Successive Cancellation List (SCL) decoding algorithm to design a near-optimal and versatile coding method. The proposed steganographic coding method named Steganographic Polar Codes (SPC) is applicable to various distortion functions with high embedding efficiency and low embedding complexity. Extensive experimental results on various simulated distortion profiles and image distortion functions are reported to validate the superior coding performance of SPC when compared with STC.

The significance of this paper lies in that it verifies the feasibility of polar codes for designing steganographic codes and proposes another and a better set of near-optimal and versatile steganographic codes in practice. This paper also presents a design methodology to make it easy to incorporate any advance of polar coding and decoding algorithms for designing better steganographic polar codes. The main concrete contributions of this paper are listed below.

- Based on two frozen indices determination methods of polar codes, propose to construct the steganographic parity-check matrix by setting steganographic embedding payload as the initial value of Arikan's heuristic [21], [26] and choosing a resultful β for β-expansion [27].
- Propose a valid formula mapping the steganographic distortion function to the channel metric for polar decoding, taking advantage of the optimal modification probability under the minimal distortion model.
- Improve the embedding efficiency by using the superior Successive Cancellation List (SCL) decoding algorithm, owning the flexible design parameter of list size l that affects the embedding efficiency and speed.
- Introduce three strategies to generalize SPC for arbitrary cover length, and recommend the cover-padding strategy.

The rest of this paper is organized as follows. In Section II, minimal steganographic distortion model, polar codes and a relationship between Binary Symmetric Channel and steganographic channel are briefly reviewed. We elaborate the proposed SPC specialized for cover length of a power of 2 in Section III. Three strategies of generalizing SPC to arbitrary cover length are then introduced in Section IV. To verify the feasibility of SPC, we carry out extensive simulation experiments and apply it to image steganography in Section V and Section VI, respectively, with sufficient comparisons and analysis. The paper is concluded in Section VII.

II. Preliminaries

In this paper, sets, vectors and matrices are written in boldface. Vector $\mathbf{a} = \mathbf{a}_1^n$, and the vector $\mathbf{a}_i^j = (a_i, \cdots, a_j)$ is a subsequence of $\mathbf{a}$ from its i-th element to j-th element. Let $\mathbf{u}$, $\mathbf{c}$, $\mathbf{r}$ represent the source word, the codeword, the received word in polar codes, respectively. Let $\mathbf{m}$, $\mathbf{x}$, $\mathbf{y}$ represent the message, the cover sequence, the stego sequence in steganography, respectively. The embedding operation on x_i is formulated by the dynamic range $\mathcal{I}_i$. For *binary* embedding, $\mathcal{I}_i = \{x_i, \bar{x}_i\}$ where $\bar{x}_i$ is x_i after flipping its Least Significant Bit (LSB), while $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\}$ is for *ternary* embedding [8]. A q-ary entropy function is denoted by $H(\pi_1, \cdots, \pi_q)$ for $\sum_{i=1}^{q} \pi_i = 1$, where binary entropy function is $H(\pi) = -\pi \log_2 \pi - (1-\pi)\log_2(1-\pi)$. The symbol $\ln \pi$ denotes the natural logarithm.

A content-adaptive steganographic system is depicted in Fig. 1. At the sending side, the sender uses a distortion function to calculate the modification cost $\boldsymbol{\rho}$ of cover $\mathbf{x}$, and then obtains stego $\mathbf{y}$ by using a coding method on encoding message $\mathbf{m}$ associted with $\mathbf{x}$ and $\boldsymbol{\rho}$. The stego $\mathbf{y}$ is transmitted to the receiver through a lossless channel. At the receiving side, the receiver extracts $\mathbf{m}$ directly by using the corresponding decoding method on $\mathbf{y}$. Through such a steganographic communication process, the sender and receiver can realize a covert sharing of the message. And this paper is focusing on the core coding problem for message embedding and extraction.

A. Minimal Distortion Model and Syndrome Coding

Under an additive distortion scenario of content-adaptive steganography, the impacts of embedding changes are assumed to be mutually independent, so the total distortion for

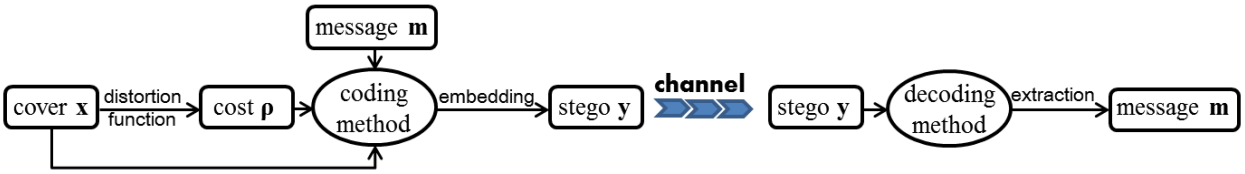


Fig. 1. Communication diagram of content-adaptive steganography.

embedding is the sum of the costs $\rho(y_i)$ at x_i changed to y_i [8]:

$$D(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \rho(y_i). \tag{1}$$

Denote $\pi(y_i)$ as the probability of modifying x_i to y_i, the PLS problem can be formulated as the optimization problem:

$$\underset{\boldsymbol{\pi}}{\text{minimize}}\ E_{\boldsymbol{\pi}}(D) = \sum_{i=1}^{n}\sum_{t_i\in\mathcal{I}_i} \pi(t_i)\rho(t_i) \tag{2}$$

$$\text{subject to } H(\boldsymbol{\pi}) = -\sum_{i=1}^{n}\sum_{t_i\in\mathcal{I}_i} \pi(t_i)\log_2\pi(t_i) = m, \tag{3}$$

where the sender can send up to $H(\boldsymbol{\pi}) = m$ bits of message with the minimal average distortion. Following the maximum entropy principle, the optimal π_λ has a Gibbs distribution [7]:

$$\pi_\lambda(y_i) = \frac{\exp(-\lambda\rho(y_i))}{\sum_{t_i\in\mathcal{I}_i}\exp(-\lambda\rho(t_i))}, \quad 1\le i\le n, \tag{4}$$

where the scalar parameter λ ($\lambda > 0$) is determined by (3).

For a binary embedding operation, the PLS problem can be realized in practice using syndrome coding with the embedding and extraction mappings:

$$\begin{cases} \text{Emb}(\mathbf{x},\mathbf{m}) = \arg\min\limits_{\mathcal{P}(\mathbf{y})\in\mathcal{C}(\mathbf{m})} D(\mathbf{x},\mathbf{y}) \\ \text{Ext}(\mathbf{y}) = \mathcal{P}(\mathbf{y})\mathbb{H}^{\mathrm{T}} = \mathbf{m}, \end{cases} \tag{5}$$

where $\mathcal{P}: \mathcal{X}\to\{0,1\}$ is a parity function shared between the sender and the receiver (e.g., the LSB layer $\mathcal{P}(x) = x \bmod 2$). $\mathbb{H}^{\mathrm{T}} \in \{0,1\}^{n\times m}$ is the parity-check matrix of a binary code $\mathcal{C}(n, n-m)$. $\mathcal{C}(\mathbf{m}) = \{\mathbf{z}\in\{0,1\}^n | \mathbf{z}\mathbb{H}^{\mathrm{T}} = \mathbf{m}\}$ is the coset corresponding to syndrome $\mathbf{m}$.

It is well known in the community that the decoding method of error-correcting codes can be used as the coding method of steganography [8]–[20], [28]–[32]. Specifically, with satisfying the syndrome constraint, the closest stego along with small distortion can be found by the decoding process of error-correcting codes, e.g., the Viterbi decoding method for designing STC [8], [33].

B. Polar Coding and Decoding

1) Construction of Polar Codes: A polar code may be specified completely by $(n, k, \mathcal{A}, \mathbf{u}_{\mathcal{A}^c})$. Set $\mathcal{A}$ of dimension k ($k < n$) is the set of *information* indices that carry information bits $\mathbf{u}_{\mathcal{A}}$, while its complement is the *frozen* indices $\mathcal{A}^c$ that carry frozen bits $\mathbf{u}_{\mathcal{A}^c}$ of dimension $n-k$. The choice of $\mathcal{A}^c$ is a critical step in polar coding, which corresponds to the selection of k "worst" polarized channels [21], [27].

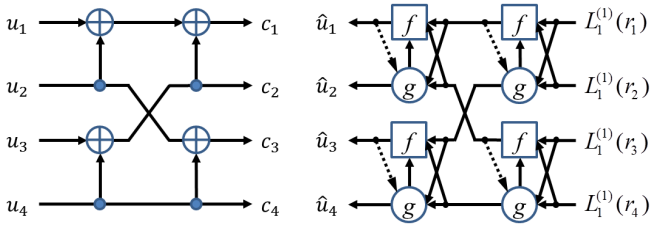


Fig. 2. Illustration of encoder and SC decoder implementation of polar codes for $n = 4$, where nodes f and g in decoder correspond to nodes $\oplus$ and $\bullet$ in encoder, respectively. A concrete example of polar encoding and decoding with numerical calculation is presented in Fig. 5.

Frozen bits $\mathbf{u}_{\mathcal{A}^c}$ can be arbitrary and is known both to the sender and receiver. Bits $\mathbf{u}_{\mathcal{A}}$ and $\mathbf{u}_{\mathcal{A}^c}$ together constitute the source word $\mathbf{u} = (\mathbf{u}_{\mathcal{A}}, \mathbf{u}_{\mathcal{A}^c})$. As depicted in Fig. 2(a), a codeword $\mathbf{c}$ is generated by polar encoding $\mathbf{u}$, i.e., $\mathbf{c} = \mathbf{u}\mathbf{G}_n$, in time complexity $O(n\log_2 n)$. The generator matrix is $\mathbf{G}_n = \mathbf{B}_n\mathbf{F}^{\otimes s}$ for any $n = 2^s$, where $\mathbf{B}_n$ is a bit-reversal permutation matrix, $\mathbf{F}^{\otimes s}$ denotes the sth Kronecker power of $\mathbf{F}$, and $\mathbf{F} \triangleq \begin{bmatrix} 1, 0 \\ 1, 1 \end{bmatrix}$. In Fig. 2(a), the polar encoding structure can be expressed as the generator matrix

$$\mathbf{G}_4 = \begin{bmatrix} 1,0,0,0 \\ 1,0,1,0 \\ 1,1,0,0 \\ 1,1,1,1 \end{bmatrix}.$$

2) Successive Cancellation Decoding and Its List Version: Given frozen bits $\mathbf{u}_{\mathcal{A}^c}$, received word $\mathbf{r}$ and the estimates $\widehat{\mathbf{u}}_1^{i-1}$ of $\mathbf{u}_1^{i-1}$, Successive Cancellation (SC) decoder [21] attempts to estimate u_i. As illustrated in Fig. 2(b), this can be implemented by computing Log-Likelihood Ratio (LLR) $L_n^{(i)}(\mathbf{r}_1^n, \widehat{\mathbf{u}}_1^{i-1}) \triangleq \ln\frac{W_n^{(i)}(\mathbf{r}_1^n, \widehat{\mathbf{u}}_1^{i-1}|0)}{W_n^{(i)}(\mathbf{r}_1^n, \widehat{\mathbf{u}}_1^{i-1}|1)}$ $(1\le i\le n)$ according to the recursive formula:

$$\begin{cases} L_n^{(2j-1)}(\mathbf{r}_1^n, \widehat{\mathbf{u}}_1^{2j-2}) \\ = f\big(L_{n/2}^{(j)}(\mathbf{r}_1^{n/2}, \widehat{\mathbf{u}}_{1,o}^{2j-2}\oplus\widehat{\mathbf{u}}_{1,e}^{2j-2}), L_{n/2}^{(j)}(\mathbf{r}_{n/2+1}^n, \widehat{\mathbf{u}}_{1,e}^{2j-2})\big) \\ L_n^{(2j)}(\mathbf{r}_1^n, \widehat{\mathbf{u}}_1^{2j-1}) \\ = g\big(L_{n/2}^{(j)}(\mathbf{r}_1^{n/2}, \widehat{\mathbf{u}}_{1,o}^{2j-2}\oplus\widehat{\mathbf{u}}_{1,e}^{2j-2}), L_{n/2}^{(j)}(\mathbf{r}_{n/2+1}^n, \widehat{\mathbf{u}}_{1,e}^{2j-2}), \widehat{u}_{2j-1}\big) \end{cases} \tag{6}$$

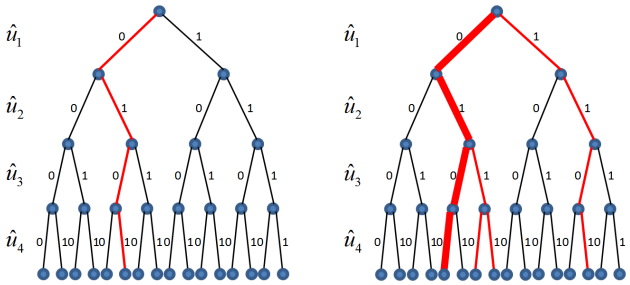
for $1\le j\le n/2$ with $f(\theta,\omega) \triangleq \ln\left(\frac{exp(\theta+\omega)+1}{exp(\theta)+exp(\omega)}\right)$, $g(\theta,\omega,u) \triangleq (-1)^u\theta + \omega$. $\widehat{\mathbf{u}}_{1,o}^i$ and $\widehat{\mathbf{u}}_{1,e}^i$ are subvectors of $\widehat{\mathbf{u}}_1^i$ with odd and even indices respectively. $L_1^{(1)}(r_i) \triangleq \ln\frac{W(r_i|0)}{W(r_i|1)}$ is the initial channel metric. Decisions are made by **Algorithm 1** in time complexity $O(n\log_2 n)$ and space complexity $O(n)$ [34].

Successive Cancellation List (SCL) decoder [34], [35] is a generalization and improvement version of the classic

Algorithm 1 SC Decoder: $(\widehat{\mathbf{u}},\widehat{\mathbf{c}}) = \mathsf{SC}\big(\mathbf{u}_{\mathcal{A}^c}, \mathcal{A}^c, \mathbf{r}, L_1^{(1)}(\mathbf{r})\big)$

Input: frozen bits $\mathbf{u}_{\mathcal{A}^c}$, frozen indices $\mathcal{A}^c$, received word $\mathbf{r}$ and its channel metric LLR $L_1^{(1)}(\mathbf{r})$.
Output: estimates $\widehat{\mathbf{u}}$ and its decoded codeword $\widehat{\mathbf{c}}$.

define $n = \text{Length}(\mathbf{r})$;
for $i = 1$ to n **do**
  calculate $L_n^{(i)}(\mathbf{r}_1^n, \widehat{\mathbf{u}}_1^{i-1})$ by (6);
  if $i \in \mathcal{A}^c$ **then** $\widehat{u}_i = u_i$; // frozen bits
  else // information bits
    if $L_n^{(i)}(\mathbf{r}_1^n, \widehat{\mathbf{u}}_1^{i-1}) \geq 0$ **then** $\widehat{u}_i = 0$;
    else $\widehat{u}_i = 1$;
    end if
  end if
end for
obtain $\widehat{\mathbf{c}} = \widehat{\mathbf{u}}\mathbf{G}_n$;
return $(\widehat{\mathbf{u}}, \widehat{\mathbf{c}})$.



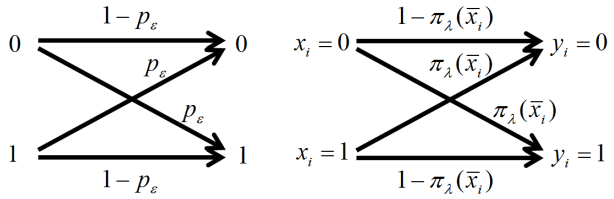
(a) SC decoding tree (b) SCL decoding tree

Fig. 3. Illustration of SC and SCL ($l = 4$) decoding on the code tree of $n = 4$, where the red-marked paths are the candidate decoding paths. SC decodes the only path of $\widehat{\mathbf{u}} = (0101)$, while SCL decodes the most probable path (red-bolded path) of $\widehat{\mathbf{u}} = (0100)$.

SC decoder. As shown in Fig. 3(a), SC decoder can be represented as a greedy search algorithm on a code tree. Since $\widehat{u}_i$ must be decided at each phase, the decoding path obtained by SC decoder is not guaranteed to be the most probable one. Instead, SCL decoder can be regarded as a breadth-first search algorithm on the code tree. As in Fig. 3(b), SCL decoder splits the decoding path into two paths when decoding an information bit. Since each split doubles the number of paths to be examined, we must prune them, and the maximum number of paths allowed is the specified list size l. Finally, a n-length path with the largest metric is selected among all the l candidate paths. Corresponding with **Algorithm 1** of SC, SCL decoder is denoted by $(\widehat{\mathbf{u}},\widehat{\mathbf{c}}) = \mathsf{SCL}\big(\mathbf{u}_{\mathcal{A}^c}, \mathcal{A}^c, \mathbf{r}, L_1^{(1)}(\mathbf{r}), l\big)$ in time complexity $O(l \cdot n\log_2\ n)$ and space complexity $O(l \cdot n)$ [34]. Naturally, larger value of l means lower decoding error rate but longer run time, and SCL is degraded to SC when $l = 1$. For general references to SC and SCL, we orientate the reader toward [21], [34], [35]. The SCL algorithm used in the proposed steganographic codes is formulated exclusively using the LLR, please see Algorithm 3 in [35].

C. Relation Between BSC and Steganographic Binary Channel

In Fig. 4(a), a binary channel W with $W(0|0) = W(1|1) = 1 - p_\epsilon$ and $W(0|1) = W(1|0) = p_\epsilon$ is Binary



(a) BSC (b) Steganographic binary channel

Fig. 4. Illustration of binary symmetric channel (BSC) and steganographic binary channel.

Symmetric Channel (BSC), where p_ϵ ($0 \leq p_\epsilon \leq 0.5$) is the channel error probability. In Fig. 4(b), a steganographic binary channel is described by $W(0|0) = W(1|1) = 1 - \pi_\lambda(\bar{x}_i)$ and $W(0|1) = W(1|0) = \pi_\lambda(\bar{x}_i)$, where $\pi_\lambda(\bar{x}_i)$ is the modification probability. Obviously, the steganographic binary channel has the same structure as the BSC. Since

$$\pi_\lambda(\bar{x}_i) = \frac{\exp(-\lambda\rho(\bar{x}_i))}{\exp(-\lambda\rho(x_i)) + \exp(-\lambda\rho(\bar{x}_i))} = \frac{\exp(-\lambda\rho(\bar{x}_i))}{1 + \exp(-\lambda\rho(\bar{x}_i))} \tag{7}$$

(in (4) with $\rho(x_i) = 0$ by default) and $\rho(\bar{x}_i) \geq 0$, the value range of $\pi_\lambda(\bar{x}_i)$ is $0 \leq \pi_\lambda(\bar{x}_i) \leq 0.5$, which is also the same as that of p_ϵ. Therefore, we can treat the steganographic binary channel as the BSC along with $p_\epsilon = \pi_\lambda(\bar{x}_i)$. This important relationship, which expediently connects the steganographic channel with a classic communication channel for polar coding and decoding, will be applied to determine the frozen indices and initial channel metrics.

III. Steganographic Polar Codes Based on SCL Decoding Algorithm

To design a practical steganographic coding method based on polar codes, four problems will be investigated in this paper:

1) how to determine the frozen indices $\mathcal{A}^c$ for constructing steganographic parity-check matrix,
2) how to calculate the decoding initial channel metric LLR by steganographic distortion,
3) how to extend binary embedding to q-ary embedding for various embedding amplitudes,
4) how to generalize the steganographic coding method to a cover object of arbitrary length.

In this section, we will elaborate our solutions to the first three problems, while the solution to the last problem will be specifically introduced in the next section.

A. Two Methods for Determining Frozen Indices

It has been proved in [36] that the parity-check matrix $\mathbb{H}^{\mathrm{T}}$ of polar codes is formed from the columns of the generator matrix $\mathbf{G}_n$ with indices in $\mathcal{A}^c$, i.e., $\mathbb{H}^{\mathrm{T}} = \mathbf{G}_n^{\mathcal{A}^c}$. According to the particular role of $\mathbb{H}^{\mathrm{T}}$ in syndrome coding (5), the syndrome $\mathbf{m}$ should be placed as the frozen bits, i.e., $\mathbf{u}_{\mathcal{A}^c} = \mathbf{m}$. Given $\mathbf{u}_{\mathcal{A}^c}$ and $\mathbf{r}$, SCL estimates $\widehat{\mathbf{u}} = (\widehat{\mathbf{u}}_{\mathcal{A}}, \mathbf{u}_{\mathcal{A}^c})$ having corresponding decoded codeword $\widehat{\mathbf{c}} = \widehat{\mathbf{u}}\mathbf{G}_n$. Since $\mathbf{G}_n$ is an invertible matrix (i.e., $\mathbf{G}_n^{-1} = \mathbf{G}_n$ [21]), we have $\widehat{\mathbf{u}} = \widehat{\mathbf{c}}\mathbf{G}_n$ with $(\widehat{\mathbf{u}}_{\mathcal{A}}, \mathbf{u}_{\mathcal{A}^c}) = (\widehat{\mathbf{c}}\mathbf{G}_n^{\mathcal{A}}, \widehat{\mathbf{c}}\mathbf{G}_n^{\mathcal{A}^c})$. Naturally, to find the stego $\mathbf{y}$ with the extraction constraint $\mathbf{m} = \mathcal{P}(\mathbf{y})\mathbb{H}^{\mathrm{T}}$ in syndrome coding (5), we can use

SCL associated with $\mathbf{u}_{\mathcal{A}^c} = \mathbf{m} = \widehat{\mathbf{c}}\mathbb{H}^{\mathrm{T}}$, so that $\mathcal{P}(\mathbf{x}) = \mathbf{r}$ becomes the input and $\mathcal{P}(\mathbf{y}) = \widehat{\mathbf{c}}$ is the output of SCL decoder.

From the analysis, the selection of parity-check matrix in polar codes equates to the determination of $\mathcal{A}^c$. Intuitively, preferable $\mathcal{A}^c$ is vital for steganographic codes. Here we introduce two efficient methods for determining $\mathcal{A}^c$ in steganographic codes as follows.

1) Arikan's Heuristic Method for Approximate Calculation of BSC's Capacity: For the Binary Erasure Channel (BEC) with erasure probability ϵ, Arikan [21] introduced a precise and efficient formula for calculating Bhattacharyya parameters $\mathbf{Z} = \big(Z(W_n^{(1)}), Z(W_n^{(2)}), \cdots, Z(W_n^{(n)})\big)$ in the recursive properties of the channel polarization:

$$\begin{cases} Z(W_n^{(2j-1)}) = 2Z(W_{n/2}^{(j)}) - Z(W_{n/2}^{(j)})^2 \\ Z(W_n^{(2j)}) = Z(W_{n/2}^{(j)})^2, \quad 1 \le j \le n/2, \end{cases} \tag{8}$$

with the initial value $Z(W_1^{(1)}) = \epsilon$ in time complexity $O(n)$. And the indices of $n-k$ largest $Z(W_n^{(i)})$ $(1 \le i \le n)$ are then selected as the frozen indices $\mathcal{A}^c$. However, (8) is theoretically perfect for the BEC rather than other communication channels, such as the BSC. In [26], Arikan suggested a heuristic method instead: given an arbitrary binary channel with capacity C bits, the construction of polar codes can be matched to the BEC with erasure probability $\epsilon = 1 - C$, i.e., the frozen indices of the given channel can be the same as that of the BEC with $\epsilon = 1 - C$. This makes it possible to employ (8) for the BSC as long as we know the capacity C of BSC.

In information theory, the capacity of BSC is $C = 1 - H(p_\epsilon)$. Since the steganographic binary channel is the BSC with $p_\epsilon = \pi_\lambda(\bar{x}_i)$, the capacity of the steganographic binary channel is $C = 1 - H(\pi_\lambda(\bar{x}_i))$. As for the constant distortion model in steganography, all cover elements have the same modification probability, so we deduce $H(\pi_\lambda(\bar{x}_i)) = m/n = \alpha$ (*embedding payload*) by (3). Because $C = 1 - H(\pi_\lambda(\bar{x}_i)) = 1 - \alpha$, we have $\epsilon = 1 - C = \alpha$ for the BEC via Arikan's heuristic method, which is served as the initial value of (8):

$$Z(W_1^{(1)}) = \epsilon = \alpha, \tag{9}$$

to determine $\mathcal{A}^c$ for the steganographic binary channel. Although (9) is deduced from the constant distortion model, experiments will show that it also works for general distortion model (see Fig. 7). We denote this method by Arikan-BSC for short.

2) β-Expansion With Base $\beta = 1.21$: β-expansion [27] is a notion borrowed from number theory, and it studies a fast construction of polar codes based on a recursive structure of universal partial order (UPO) and polarization weight (PW) algorithm. The advantage of PW algorithm is that it provides a neat and low-complex method to fully rank the reliability of synthetic channels for polar codes while keeping the property of nested frozen indices when the code length grows. See Definition 3 (PM algorithm) in [27]: consider a synthetic channel index id ($id = i - 1$ and $1 \le i \le n$) and its binary expansion $\mathbf{B} = (b_{s-1} \cdots b_1\ b_0)_2$ over $s = \lfloor \log_2\ id \rfloor + 1$ bits, its polarization weight is defined as $f^{\mathrm{PM}} : id \mapsto w_{id} = \sum_{j=0}^{s-1} b_j \beta^j$. A smaller w_{id} indicates a lower reliability of the synthetic channel, which enables the selection of frozen indices by sorting w_{id} and choosing the indices of smaller w_{id}. It has been pointed out that the value of base β should be carefully chosen [27]. Different from Arikan-BSC (i.e., (8) + (9)) that is linked to the embedding payload, β can be fixed to 1.21 for β-expansion according to the experiments.

B. Calculating LLR From Optimal Modification Probability

As for steganography, the initial channel metric LLR of steganographic binary channel for decoding can be computed via the modification probability $W(x_i|y_i) = \pi_\lambda(y_i)$ as shown in Fig. 4(b). Since $\pi_\lambda(y_i)$ is theoretically optimal for a given payload and distortion function in (4), calculating LLR from $\pi_\lambda(y_i)$ should be optimal for designing steganographic codes as well. This step is very critical to minimize steganographic distortion for embedding, because it is through these LLRs computed by their steganographic costs that SCL algorithm can recursively calculate to find a preferable stego with small distortion. According to the definition of LLR and the LSB layer $\mathcal{P}(x_i)$ of x_i $(1 \le i \le n)$, the LLR of steganographic binary channel is deduced:

$$\begin{aligned} L_1^{(1)}(x_i) &\triangleq \ln \frac{W\big(\mathcal{P}(x_i)|0\big)}{W\big(\mathcal{P}(x_i)|1\big)} \\ &= \begin{cases} \ln \dfrac{W(0|0)}{W(0|1)} = \ln \dfrac{\pi_\lambda(x_i)}{\pi_\lambda(\bar{x}_i)}, & \mathcal{P}(x_i) = 0 \\ \ln \dfrac{W(1|0)}{W(1|1)} = \ln \dfrac{\pi_\lambda(\bar{x}_i)}{\pi_\lambda(x_i)}, & \mathcal{P}(x_i) = 1. \end{cases} \end{aligned}$$

It can be further simplified by $\pi_\lambda(x_i) + \pi_\lambda(\bar{x}_i) = 1$:

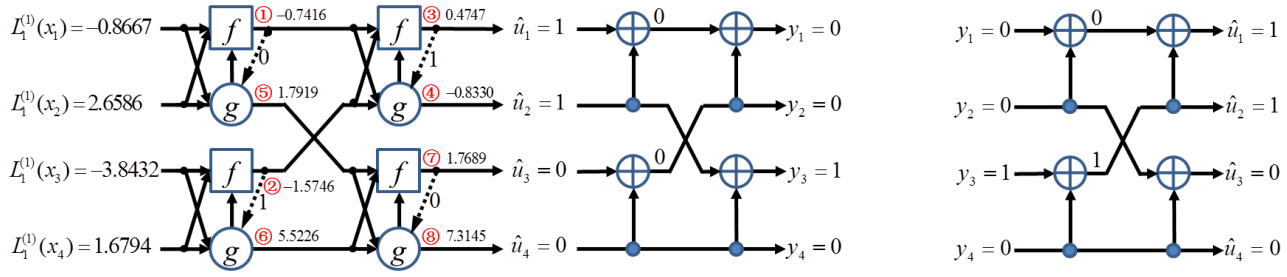
$$L_1^{(1)}(x_i) = \big(2\mathcal{P}(x_i) - 1\big) \cdot \ln \frac{\pi_\lambda(\bar{x}_i)}{1 - \pi_\lambda(\bar{x}_i)}, \quad 1 \le i \le n, \tag{10}$$

with (7) optimally relating to the steganographic distortion and payload.

C. Description and Analysis of the Proposed Coding Method

1) Algorithm Description and Application Example: The complete implementation steps of the proposed Steganographic Polar Codes (SPC) with binary embedding and extraction operations are presented in **Algorithm 2** and **Algorithm 3**, respectively. Since Arikan-BSC performs slightly better than β-expansion for determining the frozen indices according to the experiments, we recommend Arikan-BSC in SPC. Note that the cover sequence should be scrambled using a key (shared between the sender and receiver) before executing SCL algorithm, i.e., step 4 in **Algorithm 2** (symmetrically scrambling the stego sequence of step 3 in **Algorithm 3**), to achieve a satisfactory coding performance. It is also noteworthy that the sender does not need to communicate the used value of l to the receiver while the value of h in STC is needed for message extraction [8], and this less communication cost is a practical advantage of SPC.

For better understanding, we provide an example for a binary cover of length $n = 4$ to display the necessary steps required to implement message embedding and extraction of SPC. Suppose a cover sequence $\mathbf{x} = (1, 0, 1, 0)$, its



(a) Embedding with $(\hat{u}_1, \hat{u}_2) = \hat{\mathbf{u}}_{\mathcal{A}^c} = \mathbf{u}_{\mathcal{A}^c} = (u_1, u_2) = \mathbf{m} = (1, 1)$ (b) Extraction with $\mathbf{m} = (\hat{u}_1, \hat{u}_2)$

Fig. 5. Illustration of message embedding and extraction by SPC. (a) Embedding: the estimate $\hat{\mathbf{u}}$ is polar decoded from initial $L_1^{(1)}(\mathbf{x})$ and then polar encoded to the stego $\mathbf{y}$, where the red-marked numbers display the calculation orders in decoding process. (b) Extraction: the stego $\mathbf{y}$ is polar encoded to obtain the same $\hat{\mathbf{u}}$ thanks to the invertibility of the generator matrix of polar codes.

Algorithm 2 Steganographic Polar Codes (SPC) for Binary Embedding: $\big(D(\mathbf{x}, \mathbf{y}), \mathbf{y}\big) = \mathsf{SPC}_{\text{emb}}\big(\mathbf{m}, \mathbf{x}, \rho(\bar{\mathbf{x}}), l\big)$

Input: message $\mathbf{m}$, cover $\mathbf{x}$ and its cost $\rho(\bar{\mathbf{x}})$, list size l.
Output: total distortion $D(\mathbf{x}, \mathbf{y})$ and stego $\mathbf{y}$.

1: define $m = \text{Length}(\mathbf{m})$, $n = \text{Length}(\mathbf{x})$, $\alpha = m/n$ and $\mathcal{P}(\mathbf{x}) = \mathbf{x}$ mod 2;
2: calculate $\mathbf{Z}$ by (8) and (9); sort $\mathbf{Z}$ and select indices of m largest $Z(W_n^{(i)})$ as $\mathcal{A}^c$; set $\mathbf{u}_{\mathcal{A}^c} = \mathbf{m}$;
3: calculate initial LLR $L_1^{(1)}(\mathbf{x})$ by (7),(10) with m and n;
4: scramble $\mathcal{P}(\mathbf{x})$ and $L_1^{(1)}(\mathbf{x})$ to $\mathcal{P}(\mathbf{x}')$ and $L_1^{(1)}(\mathbf{x}')$ (by a key shared with the receiver);
5: embed $\mathbf{m}$ into $\mathcal{P}(\mathbf{x}')$ by SCL decoder: $\big(\hat{\mathbf{u}}, \mathcal{P}(\mathbf{y}')\big) = \mathsf{SCL}\big(\mathbf{u}_{\mathcal{A}^c}, \mathcal{A}^c, \mathcal{P}(\mathbf{x}'), L_1^{(1)}(\mathbf{x}'), l\big)$;
6: Inversely scramble $\mathcal{P}(\mathbf{y}')$ to $\mathcal{P}(\mathbf{y})$ corresponding to step 4; obtain $\mathbf{y} = \mathbf{x} - \mathcal{P}(\mathbf{x}) + \mathcal{P}(\mathbf{y})$; calculate $D(\mathbf{x}, \mathbf{y}) = \sum_1^n \rho(y_i)$;
7: **return** $\big(D(\mathbf{x}, \mathbf{y}), \mathbf{y}\big)$.

Algorithm 3 Steganographic Polar Codes (SPC) for Binary Extraction: $\mathbf{m} = \mathsf{SPC}_{\text{ext}}(m, \mathbf{y})$

Input: message length m and stego $\mathbf{y}$.
Output: message $\mathbf{m}$.

1: define $n = \text{Length}(\mathbf{y})$, $\alpha = m/n$ and $\mathcal{P}(\mathbf{y}) = \mathbf{y}$ mod 2;
2: calculate $\mathbf{Z}$ by (8) and (9); sort $\mathbf{Z}$ and select indices of m largest $Z(W_n^{(i)})$ as $\mathcal{A}^c$;
3: scramble $\mathcal{P}(\mathbf{y})$ to $\mathcal{P}(\mathbf{y}')$ (by a key shared with the sender);
4: set $\mathbf{u} = \mathcal{P}(\mathbf{y}')\mathbf{G}_n$; obtain $\mathbf{m} = \mathbf{u}_{\mathcal{A}^c}$;
5: **return** $\mathbf{m}$.

modification cost $\rho(\bar{\mathbf{x}}) = (0.1363, 0.4181, 0.6044, 0.2641)$, a message $\mathbf{m} = (1, 1)$ and a list size $l = 1$. For message embedding by **Algorithm 2**, $m = 2$ and payload $\alpha = m/n = 0.5$. According to (8) and (9), we calculate $\mathbf{Z} = (0.9375, 0.5625, 0.4375, 0.0625)$ and select $\mathcal{A}^c = (1, 2)$. Then $\mathbf{u}_{\mathcal{A}^c} = (u_1, u_2) = \mathbf{m} = (1, 1)$. With satisfying the constraint $H(\boldsymbol{\pi}) = m$ in (3), the modification probability $\boldsymbol{\pi}_\lambda(\bar{\mathbf{x}}) = (0.2959, 0.0655, 0.0210, 0.1572)$ and the LLR $L_1^{(1)}(\mathbf{x}) = (-0.8667, 2.6586, -3.8432, 1.6794)$ are calculated by (7) and (10), respectively. Suppose that $\mathbf{x}$ and $L_1^{(1)}(\mathbf{x})$ remain unchanged after the scrambling. Then $\mathbf{u}_{\mathcal{A}^c}$, $\mathbf{x}$, $L_1^{(1)}(\mathbf{x})$ and l are sent into the polar decoding algorithm. Since SCL decoder is degraded to SC decoder when $l = 1$, we depict in Fig. 5(a) the SC decoding process in lines with Fig. 2(b) and **Algorithm 1**. Also in Fig. 5(a), a polar encoding of $\hat{\mathbf{u}} = (1, 1, 0, 0)$ is required to obtain the stego sequence $\mathbf{y} = \hat{\mathbf{c}} = \hat{\mathbf{u}}\mathbf{G}_n = (0, 0, 1, 0)$. Compare $\mathbf{y} = (0, 0, 1, 0)$ with $\mathbf{x} = (1, 0, 1, 0)$, only x_1 has been modified with total distortion $D(\mathbf{x}, \mathbf{y}) = 0.1363$. As for message extraction by **Algorithm 3**, $\hat{\mathbf{u}}$ is recovered by polar encoding $\mathbf{y}$, i.e., $\hat{\mathbf{u}} = \mathbf{y}\mathbf{G}_n$ in Fig. 5(b). Then the same $\mathcal{A}^c = (1, 2)$ can be similarly determined to help extract the accurate message $\mathbf{m} = \hat{\mathbf{u}}_{\mathcal{A}^c} = (\hat{u}_1, \hat{u}_2) = (1, 1)$. Obviously, a favourable stego is found by **Algorithm 2** equipped with SCL, and the message can be extracted in a straightforward manner by the receiver using the shared frozen indices.

It is noteworthy that above SPC is designed particularly using Arikan-BSC for polar encoding and SCL for polar decoding. In general, any advance of polar coding and decoding methods can be incorporated into the design of SPC, by substituting step 2, step 5 in **Algorithm 2**.

2) Discussion on Time and Space Complexity: The time complexity of **Algorithm 2** for embedding is mainly due to the time complexity of SCL algorithm, i.e., the time complexity of SPC is $O(l \cdot n \log_2 n)$. When $l = 1$, the time complexity of SPC is reduced to $O(n \log_2 n)$. The time complexity of STC performing Viterbi algorithm is $O(2^h n)$, where h is the constraint height of parity-check submatrix and larger h corresponds to higher security but lower speed. In theory, the complexity $O(n \log_2 n)$ is worse than $O(2^h n)$ under the condition that n is large enough when h is a constant. However, the length of a cover object is finite in reality, meaning $O(n \log_2 n) < O(2^h n) = O(1024n)$ when h is usually set to 10 for STC. For an example of a typical image size $n = 512 \times 512 = 2^{18}$ in BOSSBase [37], $n \log_2 n = 18n \ll 1024n$ predicts the execution time of SPC may be less than that of STC in practice. We will compare the actual run time between SPC and STC for various n, l, h in the experimental section.

Similarly, the space complexity of SPC mainly depends on the space complexity of SCL algorithm, i.e., $O(l \cdot n)$. In practice, $O(l \cdot n)$ is also lower than the space complexity $O(2^h n)$ of STC, indicating a better availability of SPC under the case of less space in real-world applications.

3) Comparison With the Method in [25]: The method in [25] introduced another two solutions to the two problems of determining the frozen indices and calculating the

channel LLRs. For determining the frozen indices, [25] computed $\mathbf{Z}$ by (8) but with the initial value:

$$Z(W_1^{(1)}) = 2\sqrt{p_\epsilon(1-p_\epsilon)} = 0.199, \quad \text{with } p_\epsilon = 0.01. \quad (11)$$

Different from our proposed Arikan-BSC in which $Z(W_1^{(1)})$ equals the embedding payload α (9) dynamically, [25] fixed $Z(W_1^{(1)})$ to 0.199. If (9) is valid, (11) may only work for the case of payload being around 0.199. Indeed, this conjecture will be verified in the experiments, indicating the rationality of the proposed (9) for calculating $\mathbf{Z}$ on different payloads.

For the second problem, the LLR in [25] is calculated by

$$L_1^{(1)}(x_i) = \big(1 - 2\mathcal{P}(x_i)\big) \cdot \ln \frac{\rho(\bar{x}_i)}{\max\limits_{1\le i\le n}\big(\rho(\bar{x}_i)\big) - \rho(\bar{x}_i)}, \quad 1 \le i \le n, \quad (12)$$

with $W(x_i|\bar{x}_i) = 1-\rho(\bar{x}_i)/\max\limits_{1\le i\le n}\big(\rho(\bar{x}_i)\big)$. However, the value range of $W(x_i|\bar{x}_i)$ is $[0, 1]$, which violates the valid range $[0, 0.5]$ of error probability of BSC. Instead, the $W(x_i|\bar{x}_i) = \pi_\lambda(\bar{x}_i)$ in (10) is right in $[0, 0.5]$ from Gibbs distribution (4). In addition, (12) neglects the impact of embedding payload in steganography. This may be improper because the corresponding LLRs will be always the same for arbitrary payload.

As analyzed above, the method in [25] has some technical issues in solving the two key problems. We will compare our solutions (i.e., (9) and (10)) with the solutions in [25] (i.e., (11) and (12)) in the experimental section. Besides, [25] used the elementary SC decoding algorithm, while the proposed SPC assembles the superior and flexible SCL algorithm. Also note that the method in [25] could not well meet the requirements of practical use since it did not address the other two problems regarding arbitrary cover length and q-ary embedding.

D. Multi-Layered Construction for q-Ary SPC

Above SPC is for binary operation, but real-world applications require q-ary operation with various embedding amplitudes. For example, ternary (± 1) embedding is commonly used for digital image steganography since it can achieve the smaller embedding impact [22]–[24]. Filler *et al.* [8] generalized a double-layered method [38] and introduced a multi-layered construction, which enables q-ary embedding operation and is applicable to SPC as well. Note that the marginal modification probability and conditional modification probability are flipped to the cost for Viterbi decoding in binary STC, while the corresponding probabilities are converted to the channel LLR by (10) for SCL decoding in binary SPC.

IV. Strategies for Generalizing SPC to Arbitrary Cover Length

Since polar codes are inherently designed for code length of a power of 2, the above SPC is only suitable for the cover of length $n = 2^s$ (s is a positive integer). In this section, we attempt to generalize SPC to any cover length. Our idea is to adjust the original length as a power of 2, including the strategies of Cover-SegMenting (CSM), Cover-PaDding (CPD) and Cover-ShorTening (CST), so as to execute

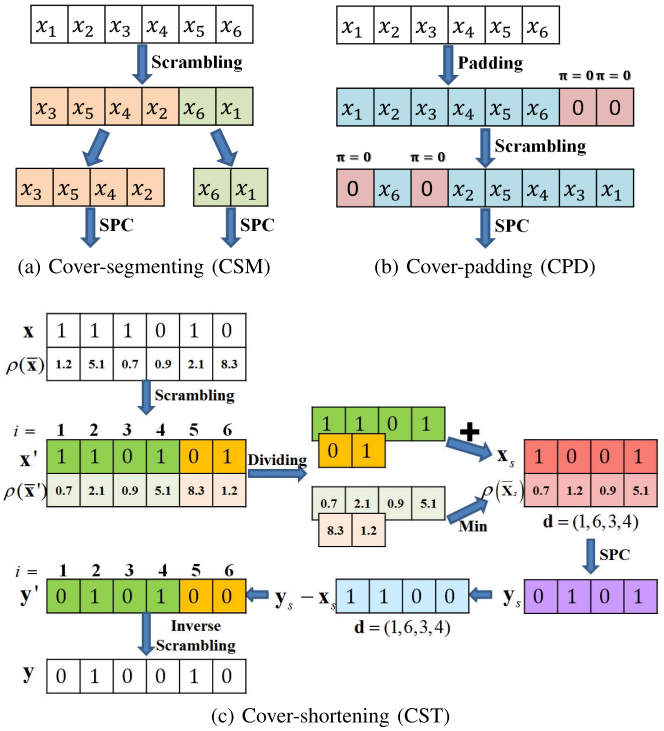


Fig. 6. Illustration of the CSM, CPD and CST strategies used for SPC embedding on a cover of length $n = 6$.

Algorithm 2 and **Algorithm 3** for message embedding and extraction directly.

A. Segmenting the Cover to Several Parts

Consider that any integer n has its binary representation $n = \mathbf{B} = (b_{s-1}\cdots b_1\ b_0)_2 = \sum_{j=0}^{s-1} b_j 2^j$ over $s = \lfloor \log_2 n \rfloor + 1$ bits, we can segment the original cover into several parts of length 2^j, enabling several independent use of above SPC. Note that before segmenting, the original cover should be scrambled in order to make the cost distributions of different parts uniform. Similarly, the message should be segmented to make the payload of each cover part uniform. As for an example of $n = 6 = (110)_2 = 2^2 + 2^1$ in Fig. 6(a), two cover parts of length 2^2 and 2^1 require to embed message respectively. Obviously, the execution times of a complete embedding equal the number of 1 in $\mathbf{B}$. We mark the SPC using cover-segmenting (CSM) as SPC-CSM for short.

B. Padding the Cover as a Larger Cover

In order to avoid multiple embedding, one option is to expand the original cover to a larger cover of length being a power of 2, by padding some wet elements whose modification probabilities are 0 in theory. For $s' = \lceil \log_2 n \rceil$, the length of the expanded cover is $n' = 2^{s'}$. A total of $\eta = n' - n$ wet elements need to be padded to the end of the original cover. In general, the value of wet elements can be optionally chosen because they exists only temporarily and will not be modified after embedding due to their 0 theoretical modification probabilities. Without loss

of generality, we set them to 0, so that the new cover with its modification probability is $\mathbf{x}_w = (x_1, \cdots, x_n, 0, \cdots, 0)$ with $\boldsymbol{\pi}_\lambda(\bar{\mathbf{x}}_w) = \big(\pi_\lambda(\bar{x}_1), \cdots, \pi_\lambda(\bar{x}_n), 0, \cdots, 0\big)$. An example of $n = 6$ is provided in Fig. 6(b). Before performing SPC, we should scramble $\mathbf{x}_w$ and $\boldsymbol{\pi}_\lambda(\bar{\mathbf{x}}_w)$ so as to spread these wet elements evenly throughout the cover sequence. This scrambling is vital for steganographic codes, making SCL more likely to search a better stego without having to change any wet element. Because the wet elements cannot be changed after embedding, the receiver can accordingly construct the same expanded and scrambled stego for extracting the message correctly. We denote this strategy as cover-padding (CPD) and mark the corresponding SPC as SPC-CPD. According to the experiments, we recommend SPC-CPD as the generalized coding method provided in **Algorithm 4** and **Algorithm 5**.

Algorithm 4 SPC-CPD for Binary Embedding: $\big(D(\mathbf{x}, \mathbf{y}), \mathbf{y}\big) = \mathsf{SPC\text{-}CPD}_{\text{emb}}\big(\mathbf{m}, \mathbf{x}, \rho(\bar{\mathbf{x}}), l\big)$

Input: message $\mathbf{m}$, cover $\mathbf{x}$ and its cost $\rho(\bar{\mathbf{x}})$, list size l.
Output: total distortion $D(\mathbf{x}, \mathbf{y})$ and stego $\mathbf{y}$.

1: define $m = \text{Length}(\mathbf{m})$, $n = \text{Length}(\mathbf{x})$ and $\mathcal{P}(\mathbf{x}) = \mathbf{x} \bmod 2$; define $s' = \lceil \log_2\ n \rceil$, $n' = 2^{s'}$, $\eta = n' - n$ and $\alpha = m/n'$;
2: calculate $\mathbf{Z} = \big(Z(W_{n'}^{(1)}), Z(W_{n'}^{(2)}), \cdots, Z(W_{n'}^{(n')})\big)$ by (8) and (9); sort $\mathbf{Z}$ and select indices of m largest $Z(W_{n'}^{(i)})$ as $\mathcal{A}^c$; set $\mathbf{u}_{\mathcal{A}^c} = \mathbf{m}$;
3: calculate $\boldsymbol{\pi}_\lambda(\bar{\mathbf{x}})$ by (7) with m and n; pad η 0 to $\mathcal{P}(\mathbf{x})$ as $\mathcal{P}(\mathbf{x}_w) = (\mathcal{P}(\mathbf{x}), 0, \cdots, 0)$, and pad η 0 to $\boldsymbol{\pi}_\lambda(\bar{\mathbf{x}})$ as $\boldsymbol{\pi}_\lambda(\bar{\mathbf{x}}_w) = (\boldsymbol{\pi}_\lambda(\bar{\mathbf{x}}), 0, \cdots, 0)$; calculate initial LLR $L_1^{(1)}(\mathbf{x}_w)$ by (10);
4: scramble $\mathcal{P}(\mathbf{x}_w)$ and $L_1^{(1)}(\mathbf{x}_w)$ to $\mathcal{P}(\mathbf{x}'_w)$ and $L_1^{(1)}(\mathbf{x}'_w)$ (by a key shared with the receiver);
5: embed $\mathbf{m}$ into $\mathcal{P}(\mathbf{x}'_w)$ by SCL decoder: $\big(\widehat{\mathbf{u}}, \mathcal{P}(\mathbf{y}'_w)\big) = \mathsf{SCL}\big(\mathbf{u}_{\mathcal{A}^c}, \mathcal{A}^c, \mathcal{P}(\mathbf{x}'_w), L_1^{(1)}(\mathbf{x}'_w), l\big)$;
6: Inversely scramble $\mathcal{P}(\mathbf{y}'_w)$ to $\mathcal{P}(\mathbf{y}_w)$ corresponding to step 4; intercept the top n elements of $\mathcal{P}(\mathbf{y}_w)$ as $\mathcal{P}(\mathbf{y})$; obtain $\mathbf{y} = \mathbf{x} - \mathcal{P}(\mathbf{x}) + \mathcal{P}(\mathbf{y})$; calculate $D(\mathbf{x}, \mathbf{y}) = \sum_1^n \rho(y_i)$;
7: **return** $\big(D(\mathbf{x}, \mathbf{y}), \mathbf{y}\big)$.

Algorithm 5 SPC-CPD for Binary Extraction: $\mathbf{m} = \mathsf{SPC\text{-}CPD}_{\text{ext}}(m, \mathbf{y})$

Input: length m of message and stego $\mathbf{y}$.
Output: message $\mathbf{m}$.

1: define $n = \text{Length}(\mathbf{y})$ and $\mathcal{P}(\mathbf{y}) = \mathbf{y} \bmod 2$; define $s' = \lceil \log_2\ n \rceil$, $n' = 2^{s'}$, $\eta = n' - n$ and $\alpha = m/n'$;
2: calculate $\mathbf{Z} = \big(Z(W_{n'}^{(1)}), Z(W_{n'}^{(2)}), \cdots, Z(W_{n'}^{(n')})\big)$ by (8) and (9); sort $\mathbf{Z}$ and select indices of m largest $Z(W_{n'}^{(i)})$ as $\mathcal{A}^c$;
3: pad η 0 to $\mathcal{P}(\mathbf{y})$ as $\mathcal{P}(\mathbf{y}_w) = (\mathcal{P}(\mathbf{y}), 0, \cdots, 0)$;
4: scramble $\mathcal{P}(\mathbf{y}_w)$ to $\mathcal{P}(\mathbf{y}'_w)$ (by a key shared with the sender);
5: set $\mathbf{u} = \mathcal{P}(\mathbf{y}'_w)\mathbf{G}_{n'}$; obtain $\mathbf{m} = \mathbf{u}_{\mathcal{A}^c}$;
6: **return** $\mathbf{m}$.

C. Shortening the Cover as a Shorter Cover

In contrast to the CPD strategy which pads the cover, another option is to shorten the original cover to a shorter cover whose length is a power of 2. For $s'' = \lfloor \log_2\ n \rfloor$, the length of the shortened cover is $n'' = 2^{s''}$ and the shortening amount is $n - n''$. However, it is unadvisable to directly intercept part of cover elements as the shortened cover, since these intercepted elements may not be in complex textured regions that are more suitable for modification. A general segment-sum algorithm [39] was proposed to construct a preferable shortened cover by selecting elements of smaller costs as much as possible. We refer to this algorithm for cover-shortening (CST) and provide an embedding example of a cover with $n = 6$ in Fig. 6(c). The corresponding SPC is marked as SPC-CST for short.

D. Analysis and Comparison of Three Strategies

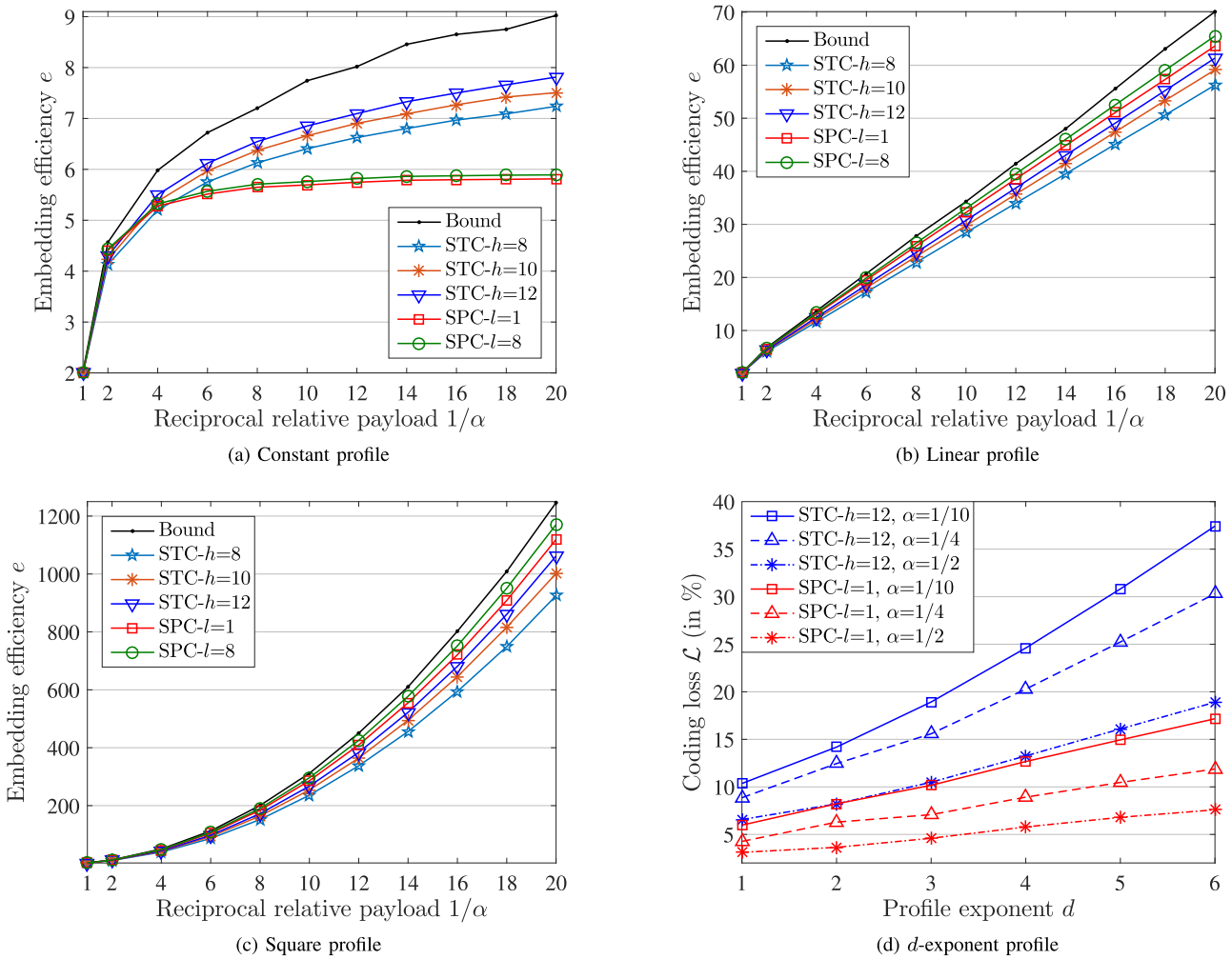
For the CSM strategy, multiple segmented covers can be processed in parallel for a faster execution than on the original cover. However, CSM has two defeats when used in practice. Firstly, since the number of bits hidden in each segmented cover must be communicated to the receiver for message extraction, multiple embedding needs some extra communication loads. Secondly and most importantly, the embedding efficiency of steganography would be affected because a polar code of short length is not as good as that of large length.

The CPD strategy will inevitably increase the embedding time when on a enlarged cover. But fortunately, its coding performance will not be damaged because the increase of wet elements does not lead to any noticeable difference in embedding efficiency for various distortions [8].

The defect of CST lies in that shortening the cover is a lossy operation that will lower the embedding efficiency of steganography, especially for large shortening amount [39]. Although SPC-CST could run faster on a shorter cover, it is not desirable if the coding performance degradation is significant. The embedding efficiencies regarding the three strategies will be examined in the following experimental section.

V. Simulation Experiments

In this section, numerical simulations are conducted by using various distortion profiles including the wet paper version, on studying the coding performance of different steganographic coding methods. Simulations are based on binary embedding operation with randomly generated cover elements and message bits of several sizes. Since STC is currently the only content-adaptive steganographic coding method, the proposed SPC is mainly compared with STC. We also compare SPC with the method in [25]. These coding methods are evaluated by using the actual embedding efficiency $e = m/D(\mathbf{x}, \mathbf{y})$ averaged over 100 simulation trials, compared with the theoretical upper bound $e_{\boldsymbol{\pi}} = m/E_{\boldsymbol{\pi}}(D)$ derived from (4). To further measure the loss degree of actual embedding efficiency to the bound, we define efficiency loss ratio $\mathcal{L} = (e_{\boldsymbol{\pi}} - e)/e_{\boldsymbol{\pi}}$ (called *coding loss* for short). While a distortion profile is spoken of ϱ if $\varrho_i = \varrho(i/n)$ for all i [8], the constant profile $\varrho(x) = 1$, linear profile



(a) Constant profile

(b) Linear profile

(c) Square profile

(d) d-exponent profile

Fig. 7. Performance comparison of STC and SPC for various distortion profiles under $n = 2^{20} \approx 10^6$ cover elements. Embedding efficiency for (a) constant profile, (b) linear profile, (c) square profiles, and coding loss for (d) d-exponent profile.

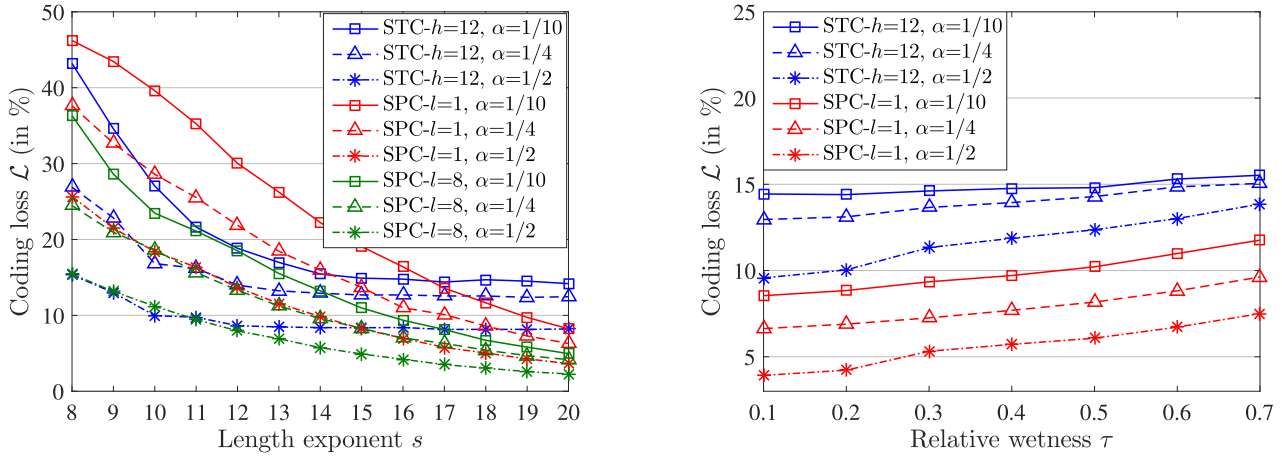
$\varrho(x) = x$, square profile $\varrho(x) = x^2$ and d-exponent profile $\varrho(x) = x^d$ are used to simulate the multi-level distortion model in real-world steganography. Meanwhile, the wet paper model, which is characterized by the profile ϱ of dry elements (with relative payload $\alpha = m/|\{x_i|\varrho_i < \infty\}$) and *relative wetness* $\tau = |\{x_i|\varrho = \infty\}|/n$, will be also examined. According to the previous description of SPC, we will analyze its coding performance for the cover length of a power of 2 and arbitrary length independently. Finally, the actual run time of SPC will be reported to show its comparable speed to STC. The list size $\in \{1, 2, 4, 8\}$ for SPC and the constraint height $h \in \{8, 10, 12\}$ for STC are selected, with three representative payloads of $1/10$, $1/4$, $1/2$ bit per element (relatively small, medium, large payload). Note that $h = 12$ is a sufficiently large value with coding performance having been converged [8].

A. Cover Length of $n = 2^s$

1) Performance for Various Distortion Profiles: The embedding efficiencies of SPC for three common simulated profiles are shown in Fig. 7(a)-(c). For the linear and square profiles, SPC of $l = 1$ outperforms STC of largest $h = 12$, and SPC of $l = 8$ experiences the highest embedding efficiency, working very close to the theoretical bound. But for the constant profile, SPC performs not as well as STC at small payloads, and both SPC and STC experience poor performance that is far from the theoretical bound. In fact, SPC and STC are not specifically designed for the constant profile, while other steganographic codes are superior for that profile, such as the ZZW family [16] (see Figure 8 in [8]).

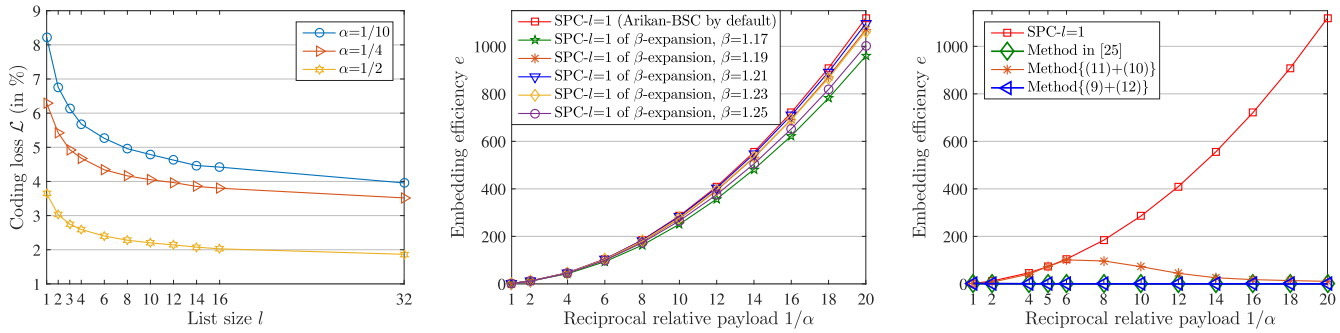
The effect of the profile shape on the coding loss for $\varrho(x) = x^d$ as a function of d is shown in Fig. 7(d). Clearly, the coding loss $\mathcal{L}$ increases with decreasing the payload α, and SPC of $l = 1$ performs much better than STC of $h = 12$ for all examined α and d. With the increase of d, the coding loss of SPC increases gently while that of STC increases rapidly, causing 20% lower coding loss of SPC to STC at $d = 6$ and $\alpha = 1/10$. This demonstrates the much superior versatility of SPC for various distortion profiles. Without loss of generality, we use the common square profile for the following experiments. Similar behaviors can be observed for other profiles.

2) Effect of List Size l: The effect of list size l of SCL algorithm on the coding loss of SPC is exhibited in Fig. 9(a). Quite naturally, the coding loss of SPC can be



(a) Effect of cover length (b) Effect of relative wetness

Fig. 8. Effect of length exponent s and relative wetness τ on the coding loss of STC and SPC under the square profile and payloads $\alpha = 1/10, 1/4, 1/2$. (a) Coding loss for length exponent s ($n = 2^s$). (b) Coding loss for wetness τ of wet paper model under $n = 2^{20} \approx 10^6$ elements.



(a) (b) (c)

Fig. 9. (a) Effect of list size l on the coding loss of SPC under $n = 2^{20} \approx 10^6$ elements, the square profile and $\alpha = 1/10, 1/4, 1/2$. Comparison of embedding efficiency between (b) two methods (Arikan-BSC and β-expansion) for determining frozen indices, and between (c) SPC and the method in [25], under $n = 2^{20} \approx 10^6$ elements and the square profile.

reduced by increasing l. According to SPC's time complexity $O(l \cdot n \log_2 n)$, l is a flexible design parameter that trades off the embedding efficiency and speed, like the constraint height h in STC. In Fig. 9(a), since a larger l does not significantly reduce the coding loss, we recommend $l \leq 8$ in real-world applications to avoid excessive embedding time. In fact, SPC of $l = 1$ can acquire comparable or superior performance to STC of large $h = 12$, and the performance advantage of SPC using $l = 8$ is thus evident.

3) Performance for Various Cover Lengths: Polar codes can achieve channel capacity as the code length is increased, i.e., the decoding performance of polar codes can be improved with the increase of code length [21]. Indeed, the coding loss of SPC decreases with increasing the length and so does STC, as shown in Fig. 8(a). Of $l = 1$, SPC is inferior to STC at short covers, and the turning points come when n reaches 2^{17}, 2^{16}, 2^{15} respectively for $\alpha = 1/10, 1/4, 1/2$. But of $l = 8$, SPC is comparable or superior to STC for almost all n and α. It has been concerned in [8] that n must be very large to apply polar codes for steganography. However, the above results dispel this concern since SPC still works well for the short cover. Note that with increasing n, the coding loss of SPC can be further reduced while that of STC has early converged, which leads to a gradually amplified coding advantage of SPC (with $5\% \sim 10\%$ losses lower than STC of $h = 12$ at $s = 20$). This advantage is practically meaningful for real-world steganography, because a cover object of large size will be more and more common with the rapid development of communication technology.

4) Performance for Wet Paper Channel: SPC can also be used to communicate via the wet paper channel without significant performance loss, as shown in Fig. 8(b). SPC achieves about 5% lower coding loss than that of STC in all cases. Note that the good availability of SPC for the wet paper model enables the use of cover-padding (CPD) on generalizing SPC for arbitrary length where the cover is padded with a number of wet elements.

5) Comparison of Polar Codes-Based Methods: In addition to Arikan-BSC, another method of β-expansion was introduced in subsection III-A-2 for determining the frozen indices. As shown in Fig. 9(b), β-expansion with $\beta = 1.21$ achieves the best performance among different β, but it is slightly inferior to Arikan-BSC (we could not find a value whose performance is better than Arikan-BSC when searching for a wider and more intensive range of β). It's worth mentioning that β-expansion may has one practical advantage. For a fixed cover length, the frozen indices determined by β-expansion are the same regardless of the payload. When communicating

TABLE I

RUN TIME t (IN SECOND) AND EMBEDDING EFFICIENCY e OF STC AND SPC WITH DIFFERENT DESIGN PARAMETERS (I.E., $h = 8, 10, 12$ AND $l = 1, 2, 4, 8$) FOR $n = 2^{15}, 2^{20}$ UNDER THE SQUARE PROFILE AND PAYLOAD $\alpha = 1/4$. THE RUN TIME IS OBTAINED AS AN AVERAGE OVER 100 TERNARY EMBEDDING, WITH MEX FILES (IN C++)[1] EXECUTED BY MATLAB R2015B ON INTEL(R) CORE(TM) I5-4590 CPU @ 3.30GHz. NOTE THAT THE CODES OF STC ARE OPTIMIZED BY USING STREAMING SIMD EXTENSIONS (SSE) INSTRUCTIONS [8], WHILE SPC IS WITHOUT SUCH CODE OPTIMIZATION

Method	Length	$n = 2^{15}$		$n = 2^{20}$	
	Parameter	e	t	e	t
STC	$h = 8$	39.362	0.040 sec	39.506	1.525 sec
	$h = 10$	41.988	0.083 sec	42.096	2.916 sec
	$h = 12$	**43.779**	**0.258 sec**	**43.928**	**8.432 sec**
SPC	$l = 1$	43.280	0.074 sec	**47.018**	**3.290 sec**
	$l = 2$	**44.413**	**0.106 sec**	47.455	4.576 sec
	$l = 4$	45.280	0.162 sec	47.827	6.779 sec
	$l = 8$	45.991	0.262 sec	48.087	10.986 sec

images of a same cover length from a particular library, this enables both the sender and the receiver to determine the frozen indices offline in advance, so that the calculation of frozen indices can be avoided in the embedding and extraction process.

We also compare the proposed SPC with the method in [25]. As shown in Fig. 9(c), the method in [25] has a poor performance for the square profile (as well as for other tested profiles), because its solutions (11) and (12) neglect the impact of the embedding payload and (12) goes against the valid value range of error probability of BSC as analyzed in subsection III-C-3. In order to further verify the feasibility of our solutions, (9) and (10) are intersected with (11) and (12) to form some assembled coding methods denoted by Method$\{s1 + s2\}$ using solutions $s1 \in \{(9), (11)\}$ and $s2 \in \{(10), (12)\}$. Obviously, SPC is the Method$\{(9) + (10)\}$ while the method in [25] is the Method$\{(11) + (12)\}$. In Fig. 9(c), Method$\{(9) + (12)\}$ also performs poorly, once again verifying the irrationality of (12). With (10), Method$\{(11) + (10)\}$ works only for $1/\alpha \approx 5$ (payload $\alpha \approx 0.199$). This also verifies the serious defect of (11) fixing the initial value of (8). Instead, Arikan-BSC (9) is dynamically linked to the payload. Therefore, the proposed solutions (9) and (10) are suitable for designing a versatile steganographic coding method based on polar codes.

6) Comparison on Run Time: As discussed in subsection III-C-2, the execution time of SPC may be less than that of STC in practice according to SPC's time complexity $O(l \cdot n \log_2 n)$ versus STC's $O(2^h n)$. TABLE I reports the actual run time of SPC and STC w.r.t. some design parameters on two cover lengths. See bolded data that SPC (without code optimization) achieves higher embedding efficiency with higher embedding speed (twice faster) than the code-optimized STC of $h = 12$ for both cover lengths. Therefore, SPC is obviously applicable in practice with higher embedding efficiency and lower embedding complexity.

[1]The codes of SPC will be made available at https://github.com/WeixiangLi-93/Steganographic-Polar-Codes, while the codes of STC are downloaded from http://dde.binghamton.edu/download/syndrome/.

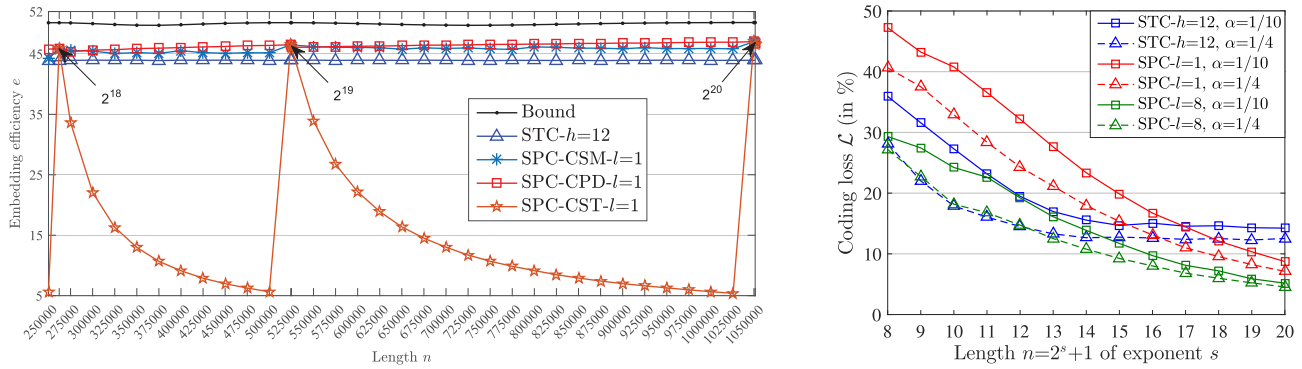
B. Cover Length of Arbitrary n

1) Comparison of Three Generalized SPCs: We first examine the coding performance of three generalized SPCs: SPC-CSM, SPC-CPD and SPC-CST, versus that of STC. As shown in Fig. 10(a), SPC-CPD achieves the highest embedding efficiency for almost all cover lengths when $l = 1$ (the same conclusion can be drawn for a larger l). Obviously, the increase of wet elements does not lead to performance loss for SPC-CPD, again verifying the good adaptability of SPC to the wet paper model. SPC-CSM also experiences the higher embedding efficiency, but it is not recommended in practice since CSM may be subjected to the limitation of the short cover and needs multiple embedding. SPC-CST is only effective for small shortening amounts (i.e., the parts of lengths slightly greater than $2^{18}, 2^{19}, 2^{20}$) while its performance decreases significantly for large shortening amounts, indicating that the ability of CST to select smaller costs becomes more and more limited with the increase of shortening amount $(n - 2^{\lfloor \log_2 n \rfloor})$.

2) Performance of SPC-CPD on Worst Cases: We also test the coding performance of SPC-CPD when it is used in the worst cases of cover lengths being $n = 2^s + 1$. For $n = 2^s + 1$, totally $\eta = 2^{s+1} - n = 2^s - 1$ wet elements are padded to construct an enlarged cover of length 2^{s+1}, whose relative wetness reaches almost 0.5. As illustrated in Fig. 10(b), SPC-CPD can also work well on these worst cases of different s. Of $l = 8$, SPC-CPD has comparable performance to STC of $h = 12$ for short covers. With increasing s, the coding loss of SPC-CPD becomes smaller and smaller, exhibiting the superior performance advantage of SPC versus STC. Consequently, the proposed SPC-CPD is applicable to the cover of any length, with a satisfactory coding performance.

Through the above simulation experiments on various distortion profiles, embedding payloads and cover lengths, polar codes are demonstrated to possess the ability of designing a versatile steganographic coding method. The proposed SPC can achieve near-optimal coding performance with low embedding complexity. Even though the performance of STC is very close to the theoretical bound, SPC performs still better than STC in general. We believe that the superior performance of SPC compared with STC benefits from the superior performance of polar codes compared with convolutional codes. Our work is significantly meaningful for providing not only another but also a better steganographic coding method to increase the diversity of steganographic codes in real-world applications.

While the experiments are conducted for the PLS problem, it should be noted that SPC is also suitable for the DLS problem which maximizes the payload with a constraint on the overall distortion and is dual to the PLS problem [8]. It is also to be noted that the aforementioned four critical problems together constitute a general and complete methodology for the design of steganographic codes based on polar codes, while specific solutions to the four problems correspond to a specific form of steganographic polar codes. Therefore, any advance of polar coding and decoding methods should be easily used to design better steganographic codes under the guidance of such a methodology.



(a) (b)

Fig. 10. (a) Embedding efficiency of STC and three generalized SPCs (SPC-CSM, SPC-CPD, SPC-CST) for n across between 2^{18} and 2^{20} under $\alpha = 1/4$ and the square profile. (b) Coding loss of SPC-CPD for the worst cases $n = 2^s + 1$ under $\alpha = 1/10, 1/4$ and the square profile.

VI. Applications to Image Steganography

In this section, we will show applications of the proposed SPC to spatial image and JPEG image steganography. The coding performance of SPC and STC will be validated by the empirical security in resisting the detection of modern blind steganalysis using rich features [40]–[42].

A. Experimental Setup

Experiments are conducted on two famous steganographic image sets: BOSSBase 1.01 [37] and MRNC [43]. The BOSSBase database contains 10,000 gray-scale images of size $512 \times 512 = 2^{18}$ pixels. The MRNC database includes 8,000 gray-scale images of size $768 \times 768 = 2^{19} + 2^{16}$ pixels, which furnishes a particular cover length not being a power of 2 for testing SPC-CPD. Two databases are also JPEG compressed with quality factor 75 as the image sets for JPEG steganography. As for spatial image steganography, we use the state-of-the-art additive distortion functions of S-UNIWARD [23] and HILL [22], and the steganalytic feature set of SRM-34,671D [40]. As for JPEG image steganography, we employ the mainstream distortion functions of J-UNIWARD [23] and UERD [24], and the steganalytic feature sets of DCTR-8,000D [41] and GFR-17,000D [42]. SPC and STC are used for message embedding in their binary and ternary forms with relative payload $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ bit per pixel (bpp) or bit per nonzero AC coefficient (bpnzac). The optimal embedding simulator [44] is also applied as the upper bound to evaluate the coding performance of SPC and STC. The steganalyzer is trained by using the above feature sets with FLD ensemble [45] by default. The FLD ensemble can minimize the total classification error probability under equal priors $P_{\rm E} = \min_{P_{\rm FA}} \frac{1}{2}(P_{\rm FA} + P_{\rm MD})$ where $P_{\rm FA}$ and $P_{\rm MD}$ are the false-alarm (FA) probability and the missed-detection (MD) probability, respectively. The ultimate security is qualified by average error rate $\overline{P_{\rm E}}$ averaged over 10 random 5000/5000 (BOSSBase) or 4000/4000 (MRNC) splits of the database, and larger $\overline{P_{\rm E}}$ means stronger security.

B. Experimental Results and Analysis

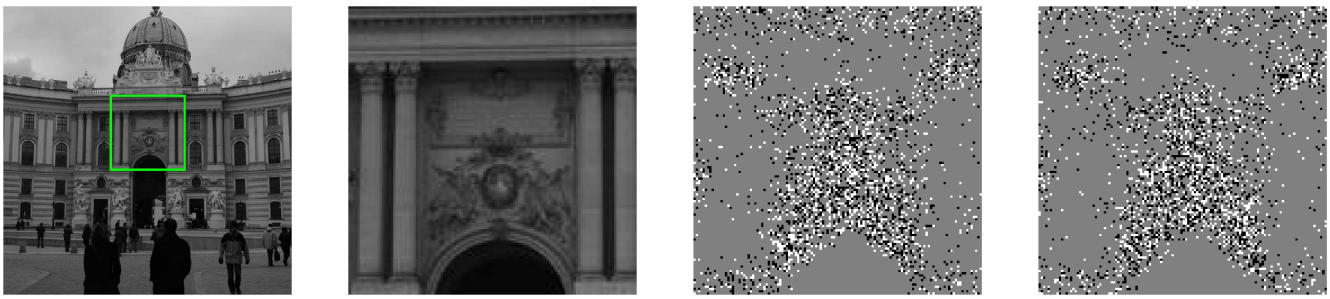
1) Spatial Image Steganography (See TABLE II): For the BOSSBase database, the detection errors of SPC are closer to that of the optimal embedding simulator in almost all cases when compared with STC. SPC of $l = 1$ is securer than STC of $h = 12$ by about 0.5% at most payloads both for binary and ternary S-UNIWARD and HILL. We in Fig. 11 visualize the modifications of a sample cover image embedded by STC of $h = 12$ and SPC of $l = 1$ in their ternary (± 1) forms. Clearly, modifications caused by SPC are mainly distributed in the complex textured regions, as done by STC. With total distortion $D(\mathbf{x}, \mathbf{y}) = 1.569 \times 10^4$ and 31,402 modifications, SPC is verified to perform better than STC having total distortion $D(\mathbf{x}, \mathbf{y}) = 1.623 \times 10^4$ and 32,212 modifications. Since the security of SPC of $l = 1$ is very close (or similar) to the optimal simulator, larger $l = 8$ does not enhance the security of SPC. Namely, there is no room for improving SPC by increasing l.

For the MRNC database with images of length $768 \times 768 = 2^{19} + 2^{16}$, SPC-CPD of $l = 8$ have comparable security to STC of $h = 12$. This validates that SPC can still

TABLE II

Detection Errors $\overline{P_{\rm E}}$ (in %) of STC, SPC in Binary or Ternary Forms for Spatial Steganography, Using SUNI (S-UNIWARD) and HILL Against SRM-34,671D on Two Sets

Database	Method	0.1bpp	0.2bpp	0.3bpp	0.4bpp	0.5bpp
BOSSBase (512×512)	SUNI-Binary-Simulator	37.64	27.58	19.87	13.82	9.39
	SUNI-Binary-STC-h=10	36.74	26.18	18.90	13.13	8.53
	SUNI-Binary-STC-h=12	36.75	26.47	19.02	13.38	8.91
	SUNI-Binary-SPC-l=1	37.55	27.32	19.69	13.79	9.17
	SUNI-Binary-SPC-l=8	**37.91**	**27.33**	**19.83**	**13.93**	**9.24**
	SUNI-Ternary-Simulator	40.75	32.14	25.52	20.27	16.16
	SUNI-Ternary-STC-h=10	39.99	31.02	24.56	19.51	14.89
	SUNI-Ternary-STC-h=12	40.24	31.54	24.75	19.67	15.24
	SUNI-Ternary-SPC-l=1	**40.52**	**31.89**	25.23	20.12	15.72
	SUNI-Ternary-SPC-l=8	40.41	31.63	**25.25**	**20.13**	**15.78**
	HILL-Binary-Simulator	41.50	31.83	24.27	17.68	12.93
	HILL-Binary-STC-h=10	40.58	30.85	23.14	16.70	11.77
	HILL-Binary-STC-h=12	40.57	30.83	23.07	16.93	12.01
	HILL-Binary-SPC-l=1	40.98	31.32	23.72	17.39	12.37
	HILL-Binary-SPC-l=8	**41.22**	**31.35**	**23.72**	**17.50**	**12.41**
	HILL-Ternary-Simulator	43.76	36.15	29.60	24.09	20.03
	HILL-Ternary-STC-h=10	43.28	34.73	28.30	22.90	18.59
	HILL-Ternary-STC-h=12	43.15	35.14	28.55	23.15	18.95
	HILL-Ternary-SPC-l=1	43.36	**35.65**	28.97	23.71	19.33
	HILL-Ternary-SPC-l=8	**43.65**	35.55	**29.16**	**23.85**	**19.53**
MRNC (768×768)	SUNI-Ternary-Simulator	42.95	34.56	28.41	23.29	19.52
	SUNI-Ternary-STC-h=10	41.55	33.63	27.34	22.47	18.76
	SUNI-Ternary-STC-h=12	**42.03**	33.81	27.32	22.73	18.81
	SUNI-Ternary-SPC-CPD-l=1	41.67	33.70	27.39	**22.87**	19.06
	SUNI-Ternary-SPC-CPD-l=8	41.60	**33.82**	**27.71**	22.74	**19.34**



(a) Full-size image (b) Cropped cover image (c) Modifications by STC (d) Modifications by SPC

Fig. 11. Modifications of (b) a cropped cover image embedded by (c) STC of $h = 12$ and (d) SPC of $l = 1$ respectively, using ternary (± 1) embedding, HILL and payload 0.5 bpp, where white represents $+1$ and dark represents -1. The cover image of size 128×128 pixels, containing smooth, edges and textured regions, is cropped from a full-size image "1013.pgm" in BOSSBase.

TABLE III

DETECTION ERRORS $\overline{P_E}$ (IN %) OF STC AND SPC IN THEIR BINARY OR TERNARY FORMS FOR JPEG IMAGE STEGANOGRAPHY, USING UERD AND JUNI (J-UNIWARD) AGAINST DCTR-8,000D AND GFR-17,000D ON TWO DATABASES COMPRESSED BY QUALITY FACTOR 75

Database	Method	DCTR					GFR				
		0.1bpnzac	0.2bpnzac	0.3bpnzac	0.4bpnzac	0.5bpnzac	0.1bpnzac	0.2bpnzac	0.3bpnzac	0.4bpnzac	0.5bpnzac
BOSSBase (512×512)	UERD-Binary-Simulator	41.00	28.71	16.08	7.65	2.93	37.87	24.07	13.14	6.46	2.72
	UERD-Binary-STC-h=10	39.71	26.07	13.57	5.83	2.06	36.38	21.16	10.91	4.85	2.03
	UERD-Binary-STC-h=12	40.07	26.52	14.18	6.37	2.28	36.66	21.55	11.30	5.21	2.13
	UERD-Binary-SPC-l=1	39.80	26.12	14.36	6.31	2.37	36.10	21.61	11.49	5.19	2.23
	UERD-Binary-SPC-l=8	**40.27**	**26.87**	**15.08**	**6.81**	**2.57**	**36.74**	**22.12**	**12.07**	**5.69**	**2.47**
	UERD-Ternary-Simulator	42.84	32.80	22.63	14.24	8.29	39.62	27.29	17.39	10.59	6.11
	UERD-Ternary-STC-h=10	41.71	30.47	19.81	12.01	6.57	38.20	24.77	14.66	8.37	4.55
	UERD-Ternary-STC-h=12	**41.99**	30.89	20.69	12.32	6.87	**38.59**	25.48	15.19	8.76	4.83
	UERD-Ternary-SPC-l=1	41.48	30.32	20.22	12.53	7.08	37.74	24.82	15.27	8.72	4.92
	UERD-Ternary-SPC-l=8	41.63	**31.22**	**20.97**	**13.15**	**7.15**	38.30	**25.59**	**15.74**	**9.25**	**4.96**
	JUNI-Binary-Simulator	42.13	29.33	16.50	7.45	2.35	39.07	24.02	12.46	5.59	2.06
	JUNI-Binary-STC-h=10	40.82	26.50	14.03	5.34	1.50	37.22	21.30	10.33	4.12	1.28
	JUNI-Binary-STC-h=12	40.82	27.14	14.67	5.82	1.82	37.44	21.79	10.73	4.53	1.48
	JUNI-Binary-SPC-l=1	40.48	26.90	14.68	5.91	1.70	37.01	21.46	10.87	4.56	1.47
	JUNI-Binary-SPC-l=8	**41.04**	**27.85**	**15.45**	**6.40**	**2.01**	**37.81**	**22.35**	**11.49**	**4.92**	**1.70**
	JUNI-Ternary-Simulator	43.75	33.99	23.92	15.35	8.83	40.81	28.36	17.97	10.43	5.87
	JUNI-Ternary-STC-h=10	42.69	32.10	21.49	12.90	6.99	39.38	26.16	15.22	8.31	4.25
	JUNI-Ternary-STC-h=12	**42.90**	32.46	21.77	13.30	7.45	39.54	26.19	15.87	8.94	4.53
	JUNI-Ternary-SPC-l=1	42.50	31.98	21.67	13.45	7.41	38.92	26.00	15.76	8.63	4.53
	JUNI-Ternary-SPC-l=8	42.74	**32.79**	**22.51**	**14.21**	**7.88**	**39.61**	**26.83**	**16.26**	**9.36**	**4.96**
MRNC (768×768)	UERD-Ternary-Simulator	41.33	28.52	16.75	8.93	3.78	37.70	22.51	12.22	5.78	2.22
	UERD-Ternary-STC-h=10	39.70	25.66	13.98	6.73	2.57	35.40	19.67	9.60	3.94	1.45
	UERD-Ternary-STC-h=12	39.76	26.22	14.44	7.21	2.81	**35.78**	20.44	10.27	4.20	1.56
	UERD-Ternary-SPC-CPD-l=1	39.78	26.26	14.93	7.42	3.12	35.04	20.26	10.23	4.49	1.71
	UERD-Ternary-SPC-CPD-l=8	**39.95**	**26.61**	**15.22**	**7.80**	**3.17**	35.56	**20.51**	**10.37**	**4.71**	**1.77**

be used for the cover of length not being a power of 2 with high security.

2) JPEG Image Steganography (See TABLE III): Unlike spatial image steganography, there is some room in JPEG steganography for improving SPC to approach the security of optimal embedding simulator by increasing l. In the both databases, SPC of $l = 8$ achieves higher securities than STC of $h = 12$ and SPC of $l = 1$, by $0.5\% \sim 1.0\%$ at most cases for different distortion functions and steganalytic feature sets. Consequently, SPC is also suitable and more secure for JPEG image steganography.

Above applications to image steganography demonstrate the availability of SPC for the real-world additive distortion functions. Overall, SPC performs better than STC even though the performance of STC is very close to the optimal embedding simulator. Obviously, the use of SPC is not limited to embedding amplitudes and cover sizes. Since SPC provides an off-the-shelf method with near-optimal coding performance in practice, the only task left to the steganographer is the choice of the distortion function for various cover objects.

VII. CONCLUSION

In this paper, we addressed four critical problems of designing steganographic codes based on polar codes, and employed the superior SCL algorithm to design the near-optimal and versatile Steganographic Polar Codes (SPC) to minimize arbitrary additive distortion with low embedding complexity. Experimental results showed that the overall coding performance of SPC is more superior than that of STC for various distortion functions. The superior performance of SPC for image steganography indicates that SPC should be able to enhance the steganographic security for other kinds of cover objects, such as audio, video and texts. This work provides another and a better choice of near-optimal steganographic codes for real-world applications, which significantly increase the diversity of coding methods in steganography.

Also importantly, this paper introduces a methodology of designing steganographic codes based on polar codes. As mentioned before, any advance of polar codes (i.e., polar coding or decoding algorithm) can be guided by the methodology to

design better steganographic codes. And this is what left for our future research.

REFERENCES

[1] Y.-C. Tseng, Y.-Y. Chen, and H.-K. Pan, "A secure data hiding scheme for binary images," *IEEE Trans. Commun.*, vol. 50, no. 8, pp. 1227–1231, Aug. 2002.

[2] C.-H. Tzeng, Z.-F. Yang, and W.-H. Tsai, "Adaptive data hiding in palette images by color ordering and mapping with security protection," *IEEE Trans. Commun.*, vol. 52, no. 5, pp. 791–800, May 2004.

[3] R. Yazdani and M. Ardakani, "Reliable communication over non-binary insertion/deletion channels," *IEEE Trans. Commun.*, vol. 60, no. 12, pp. 3597–3608, Dec. 2012.

[4] H. Tian, J. Sun, C.-C. Chang, J. Qin, and Y. Chen, "Hiding information into voice-over-IP streams using adaptive bitrate modulation," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 749–752, Apr. 2017.

[5] W. Zhang, S. Wang, and X. Zhang, "Improving embedding efficiency of covering codes for applications in steganography," *IEEE Commun. Lett.*, vol. 11, no. 8, pp. 680–682, Aug. 2007.

[6] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[7] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 705–720, Dec. 2010.

[8] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 920–935, Sep. 2011.

[9] R. Crandall. (1998). *Some Notes on Steganography*. [Online]. Available: http://dde.binghamton.edu/download/Crandall_matrix.pdf

[10] A. Westfeld, "High capacity despite better steganalysis (F5-A steganographic algorithm)," in *Proc. Int. Workshop Inf. Hiding*. New York, NY, USA: Springer-Verlag, 2001, pp. 289–302.

[11] M. Van Dijk and F. Willems, "Embedding information in grayscale images," in *Proc. 22nd Symp. Inf. Commun. Theory*, 2001, pp. 147–154.

[12] D. Schönfeld and A. Winkler, "Embedding with syndrome coding based on BCH codes," in *Proc. 8th Workshop Multimedia Secur.*, 2006, pp. 214–223.

[13] R. Zhang, V. Sachnev, and H. J. Kim, "Fast BCH syndrome coding for steganography," in *Proc. Int. Workshop Inf. Hiding*, vol. 2009, pp. 48–58.

[14] J. Bierbrauer. (1998). On Crandall's Problem. Personal communication. [Online]. Available: http://www.ws.binghamton.edu/fridrich/covcodes.pdf

[15] J. Fridrich, M. Goljan, and D. Soukal, "Efficient wet paper codes," in *Proc. Int. Workshop Inf. Hiding*, 2005, pp. 204–218.

[16] W. Zhang, X. Zhang, and S. Wang, "Maximizing steganographic embedding efficiency by combining Hamming codes and wet paper codes," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 2008, pp. 60–71.

[17] W. Zhang and X. Wang, "Generalization of the ZZW embedding construction for steganography," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 3, pp. 564–569, Sep. 2009.

[18] W. Zhang and X. Zhu, "Improving the embedding efficiency of wet paper codes by paper folding," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 794–797, Sep. 2009.

[19] W. Zhang, X. Zhang, and S. Wang, "Near-optimal codes for information embedding in gray-scale signals," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1262–1270, Mar. 2010.

[20] Y. Kim, Z. Duric, and D. Richards, "Modified matrix encoding technique for minimal distortion steganography," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 2006, pp. 314–327.

[21] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[22] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 4206–4210.

[23] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J. Inf. Secur.*, vol. 2014, no. 1, p. 1, Dec. 2014.

[24] L. Guo, J. Ni, W. Su, C. Tang, and Y.-Q. Shi, "Using statistical image model for JPEG steganography: Uniform embedding revisited," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2669–2680, Dec. 2015.

[25] B. Diouf *et al.*, "Polar coding steganographic embedding using successive cancellation," in *Innovation and Interdisciplinary Solutions for Underserved Areas*. Cham, Switzerland: Springer, 2017, pp. 189–201.

[26] E. Arkan, "A performance comparison of polar codes and reed-muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, Jun. 2008.

[27] G. He *et al.*, "Beta-expansion: A theoretical framework for fast and recursive construction of polar codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[28] C. Fontaine and F. Galand, "How Reed–Solomon codes can improve steganographic schemes," *EURASIP J. Inf. Secur.*, vol. 2009, no. 1, 2009, Art. no. 274845.

[29] C. Munuera, "Steganography from a coding theory point of view," in *Algebraic Geometry Modeling in Information Theory*. Singapore: World Scientific, 2013, pp. 83–128.

[30] W. Zhang and S. Li, "A coding problem in steganography," *Des., Codes Cryptogr.*, vol. 46, no. 1, pp. 67–81, Jan. 2008.

[31] J. Fridrich and P. Lisonek, "Grid colorings in steganography," *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1547–1549, Apr. 2007.

[32] J.-L. Kim, J. Park, and S. Choi, "Steganographic schemes from perfect codes on Cayley graphs," *Des., Codes Cryptogr.*, vol. 87, no. 10, pp. 2361–2374, Oct. 2019.

[33] Z. Zhao, Q. Guan, and X. Zhao, "Constructing near-optimal double-layered syndrome-trellis codes for spatial steganography," in *Proc. 4th ACM Workshop Inf. Hiding Multimedia Secur.*, 2016, pp. 139–148.

[34] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[35] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.

[36] N. Goela, S. B. Korada, and M. Gastpar, "On LP decoding of polar codes," in *Proc. IEEE Inf. Theory Workshop*, Aug. 2010, pp. 1–5.

[37] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Information Hiding*, 2011, pp. 59–70.

[38] W. Zhang, X. Zhang, and S. Wang, "A double layered 'plus-minus one' data embedding scheme," *IEEE Signal Process. Lett.*, vol. 14, no. 11, pp. 848–851, Nov. 2007.

[39] W. Li, W. Zhou, W. Zhang, C. Qin, H. Hu, and N. Yu, "Shortening the cover for fast JPEG steganography," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Apr. 1, 2019, doi: 10.1109/TCSVT.2019.2908689.

[40] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[41] V. Holub and J. Fridrich, "Low-complexity features for JPEG steganalysis using undecimated DCT," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 219–228, Feb. 2015.

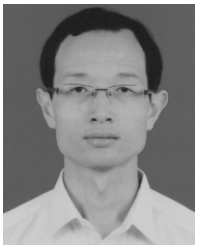
[42] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, "Steganalysis of adaptive JPEG steganography using 2D Gabor filters," in *Proc. 3rd ACM Workshop Inf. Hiding Multimedia Secur.*, 2015, pp. 15–23.

[43] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1905–1917, Sep. 2015.

[44] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," *Proc. SPIE*, vol. 6505, Feb. 2007, Art. no. 650502.

[45] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 432–444, Apr. 2012.

Weixiang Li received the B.S. degree from Xidian University (XDU) in 2016. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China (USTC). His research interests include image processing, steganography, and steganalysis. He received the Best Student Paper Award of 6th ACM IH&MMSec in 2018.



Weiming Zhang received the M.S. and Ph.D. degrees from the Zhengzhou Information Science and Technology Institute, China, in 2002 and 2005, respectively. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include information hiding and multimedia security.

Li Li received the B.S. degree from the School of Communication and Information Engineering, Harbin Engineering University, in 2016. She is currently pursuing the Ph.D. degree in information security with the University of Science and Technology of China (USTC). Her research interests include multimedia security and anomaly detection.

Hang Zhou received the B.S. degree from the School of Communication and Information Engineering, Shanghai University, in 2015. He is currently pursuing the Ph.D. degree in information security with the University of Science and Technology of China (USTC). His research interests include information hiding, image processing, and computer graphics.

Nenghai Yu received the B.S. degree from the Nanjing University of Posts and Telecommunications in 1987, the M.E. degree from Tsinghua University in 1992, and the Ph.D. degree from the University of Science and Technology of China in 2004. He is currently a Professor with the University of Science and Technology of China. His research interests include multimedia security, multimedia information retrieval, video processing, and information hiding.