6 Coding for secrecy

In this chapter, we discuss the construction of practical codes for secrecy. The design of codes for the wiretap channel turns out to be surprisingly difficult, and this area of information-theoretic security is still largely in its infancy. To some extent, the major obstacles in the road to secrecy capacity are similar to those that lay in the path to channel capacity: the random-coding arguments used to establish the secrecy capacity do not provide explicit code constructions. However, the design of wiretap codes is further impaired by the absence of a simple metric, such as a bit error rate, which could be evaluated numerically. Unlike codes designed for reliable communication, whose performance is eventually assessed by plotting a bit-error-rate curve, we cannot simulate an eavesdropper with unlimited computational power; hence, wiretap codes must possess enough structure to be provably secure. For certain channels, such as binary erasure wiretap channels, the information-theoretic secrecy constraint can be recast in terms of an algebraic property for a code-generator matrix. Most of the chapter focuses on such cases since this algebraic view of secrecy simplifies the analysis considerably.

As seen in Chapter 4, the design of secret-key distillation strategies is a somewhat easier problem insofar as reliability and security can be handled separately by means of information reconciliation and privacy amplification. Essentially, the construction of coding schemes for key agreement reduces to the design of Slepian–Wolf-like codes for information reconciliation, which can be done efficiently with low-density parity-check (LDPC) codes or turbo-codes.

We start this chapter by clarifying the connection between secrecy and capacity-achieving codes (Section 6.1), which was used implicitly in Chapter 3 and Chapter 4, to highlight the insight that can be gained from the information-theoretic proofs. We then briefly recall some fundamental properties of linear codes and LDPC codes (Section 6.2), and we use these codes as building blocks for the construction of wiretap codes over the binary erasure wiretap channel (Section 6.3) and efficient Slepian–Wolf codes for information reconciliation in secret-key distillation strategies (Section 6.4 and Section 6.5). We conclude with a discussion of secure communication over wiretap channels using secret-key distillation strategies (Section 6.6).

6.1 Secrecy and capacity-achieving codes

A natural approach by which to construct practical wiretap codes is to mimic the code structure used in the achievability proofs of Theorem 3.2 and Theorem 3.3. Specifically, we established the existence of codes for a WTC $(\mathcal{X}, p_{\mathrm{YZ|X}}, \mathcal{Y}, \mathcal{Z})$ by partitioning a codebook with $\lceil 2^{nR} \rceil \lceil 2^{nR_{\mathrm{d}}} \rceil$ codewords into $\lceil 2^{nR} \rceil$ bins of $\lceil 2^{nR_{\mathrm{d}}} \rceil$ codewords each. The $\lceil 2^{nR} \rceil \lceil 2^{nR_{\mathrm{d}}} \rceil$ codewords were chosen so that the legitimate receiver could decode reliably. In addition, the bins were constructed so that an eavesdropper knowing which bin is used could decode reliably, as well.

Each bin of codewords can be thought of as a subcode of a "mother code," which is known in coding theory as a *nested code* structure. More importantly, a closer look at the proofs shows that these subcodes are implicitly capacity-achieving codes for the eavesdropper's channel, since the rate R_{d} of each subcode is chosen in (3.24) such that

$$R_{\mathrm{d}} = \mathbb{I}(\mathsf{X}; \mathsf{Z}) - \delta(\epsilon) \quad \text{for some small } \epsilon > 0.$$

This condition is somewhat buried in the technical details and it is worth clarifying the connection between secrecy and capacity-achieving codes with a more direct proof.

Consider a WTC $(\mathcal{X}, p_{\mathrm{YZ|X}}, \mathcal{Y}, \mathcal{Z})$, and let $\mathcal{C}$ be a code of length n with $\lceil 2^{nR} \rceil$ disjoint subcodes $\{\mathcal{C}_i\}_{\lceil 2^{nR} \rceil}$ such that

$$\mathcal{C} = \bigcup_{i=1}^{\lceil 2^{nR} \rceil} \mathcal{C}_i.$$

For simplicity, we assume that $\mathcal{C}$ guarantees reliable communication over the main channel and analyze only its secrecy properties. Following the stochastic encoding suggested by the proof in Section 3.4.1, a message $m \in [\![1, 2^{nR}]\!]$ is sent by transmitting a codeword chosen uniformly at random in the subcode $\mathcal{C}_{\mathrm{m}}$. The following theorem provides a *sufficient condition* for this coding scheme to guarantee secrecy with respect to the eavesdropper.

Theorem 6.1 (Thangaraj *et al.*)**.** *If each subcode in the set* $\{\mathcal{C}_i\}_{\lceil 2^{nR} \rceil}$ *stems from a sequence of capacity-achieving codes over the eavesdropper's channel as* n *goes to infinity, then*

$$\lim_{n \to \infty} \frac{1}{n} \mathbb{I}(\mathsf{M}; \mathsf{Z}^n) = 0.$$

Proof. Let C_{e} denote the capacity of the eavesdropper's channel. If each subcode in $\{\mathcal{C}_i\}_{\lceil 2^{nR} \rceil}$ stems from a sequence of capacity-achieving codes for the eavesdropper's channel then, for any $\epsilon > 0$, there exists n large enough that

$$\forall i \in [\![1, 2^{nR}]\!] \quad \frac{1}{n} \mathbb{I}(\mathsf{X}^n; \mathsf{Z}^n | \mathsf{M} = i) \geqslant C_{\mathrm{e}} - \epsilon.$$

Consequently, $(1/n)\mathbb{I}(\mathsf{X}^n; \mathsf{Z}^n|\mathsf{M}) \geqslant C_{\mathrm{e}} - \epsilon$ as well. We now expand the mutual information $\mathbb{I}(\mathsf{M}; \mathsf{Z}^n)$ as

$$\begin{aligned} \mathbb{I}(\mathsf{M}; \mathsf{Z}^n) &= \mathbb{I}(\mathsf{Z}^n; \mathsf{X}^n \mathsf{M}) - \mathbb{I}(\mathsf{X}^n; \mathsf{Z}^n | \mathsf{M}) \\ &= \mathbb{I}(\mathsf{X}^n; \mathsf{Z}^n) + \mathbb{I}(\mathsf{M}; \mathsf{Z}^n | \mathsf{X}^n) - \mathbb{I}(\mathsf{X}^n; \mathsf{Z}^n | \mathsf{M}). \end{aligned}$$

Note that $\mathbb{I}(M; Z^n|X^n) = 0$ since $M \to X^n \to Z^n$ forms a Markov chain. In addition, $\mathbb{I}(X^n; Z^n) \leqslant nC_e$ since the eavesdropper's channel is memoryless. Therefore,

$$\begin{aligned}\frac{1}{n}\mathbb{I}(M; Z^n) &= \frac{1}{n}\mathbb{I}(X^n; Z^n) - \frac{1}{n}\mathbb{I}(X^n; Z^n|M) \\ &\leqslant C_e - (C_e - \epsilon) \\ &= \epsilon.\end{aligned}$$

□

Theorem 6.1 naturally suggests a code-design methodology based on nested codes and capacity-achieving codes over the eavesdropper's channel. Unfortunately, practical families of capacity-achieving codes are known for only a few channels, such as LDPC codes for binary erasure channels and polar codes for binary input symmetric channels; even for these channels, constructing a nested code with capacity-achieving subcodes remains a challenging task. Despite this pessimistic observation, note that the use of capacity-achieving codes for the eavesdropper's channel is merely a *sufficient condition* for secrecy, which leaves open the possibility that alternative approaches might turn out to be more successful. For instance, the code constructions for binary erasure wiretap channels presented in Section 6.3 are based on a somewhat different methodology.

Remark 6.1. *The connection between secrecy and capacity-achieving codes also holds for secret-key distillation strategies. In fact, for the secret-key distillation strategies based on Slepian–Wolf codes analyzed in Section 4.2.2, the number of bins in* (4.16) *was chosen arbitrarily close to the fundamental limit of source coding with side information. Nevertheless, the alternative approach based on sequential key-distillation circumvents this issue, and provides a design methodology that does not depend on Slepian–Wolf codes achieving the fundamental limits of source coding with side information.*

6.2 Low-density parity-check codes

Low-density parity-check (LDPC) codes constitute a family of graph-based block codes, whose performance approaches the fundamental limits of channel coding or source coding when the block length is large, and which can also be decoded efficiently with an iterative algorithm. Since we use LDPC codes extensively in the remainder of this chapter, we devote this section to a brief review (without proofs) of binary LDPC codes and their properties. We refer the interested reader to the textbook by Richardson and Urbanke [108] for a comprehensive and in-depth exposition on the subject.

6.2.1 Binary linear block codes and LDPC codes

Before discussing LDPC codes, we review some basics of binary linear block codes; in particular, the notions of dual code and coset code will be useful for secrecy codes.

Definition 6.1. *A binary* $(n, n-k)$ *block code is a set* $\mathcal{C} \subseteq \mathrm{GF}(2)^n$ *of cardinality*[1] $|\mathcal{C}| = 2^{n-k}$. *The elements of* $\mathcal{C}$ *are called codewords. Associated with the code is a bijective mapping between* $\mathrm{GF}(2)^{n-k}$ *and* $\mathcal{C}$, *which is called an encoder. The elements of* $\mathrm{GF}(2)^{n-k}$ *are called messages. An* $(n, n-k)$ *code is linear if* $\mathcal{C}$ *is an* $(n-k)$*-dimensional subspace of* $\mathrm{GF}(2)^n$. *The rate of a code* $\mathcal{C}$ *is defined as* $R \triangleq (n-k)/n$.

A linear code $\mathcal{C}$ is represented concisely by a matrix $\mathbf{G} \in \mathrm{GF}(2)^{n-k \times n}$, called the *generator matrix*, whose rows form a basis of $\mathcal{C}$. An encoder can then be described by the matrix operation

$$\mathbf{m} \mapsto \mathbf{G}^{\intercal}\mathbf{m}.$$

A generator matrix $\mathbf{G}$ specifies a code completely, but notice that $\mathbf{G}$ is not unique (any basis of $\mathcal{C}$ can be used to construct $\mathbf{G}$). Different generator matrices define different encoders.

Definition 6.2. *The dual of an* $(n, n-k)$ *linear code* $\mathcal{C}$ *is the set* $\mathcal{C}^{\perp}$ *defined as*

$$\mathcal{C}^{\perp} \triangleq \left\{ \mathbf{c} \in \mathrm{GF}(2)^n : \forall \mathbf{x} \in \mathcal{C} \quad \sum_{i=1}^{n} c_i x_i = 0 \right\}.$$

In other words, $\mathcal{C}^{\perp}$ *contains all vectors of* $\mathrm{GF}(2)^n$ *that are orthogonal to* $\mathcal{C}$.

The reader can check that $\mathcal{C}^{\perp}$ is actually an (n, k) linear code. A generator matrix of $\mathcal{C}^{\perp}$ is denoted by a matrix $\mathbf{H} \in \mathrm{GF}(2)^{k \times n}$ and is called the *parity-check matrix* of $\mathcal{C}$. Note that $\mathbf{H}$ satisfies $\mathbf{G}\mathbf{H}^{\intercal} = \mathbf{0}$ and that all codewords $\mathbf{x} \in \mathcal{C}$ must satisfy the parity-check equations $\mathbf{H}\mathbf{x} = \mathbf{0}$.

Definition 6.3. *For a linear* $(n, n-k)$ *code* $\mathcal{C}$ *with parity-check matrix* $\mathbf{H}$ *and for* $\mathbf{s} \in \mathrm{GF}(2)^k$, *the set*

$$\mathcal{C}(\mathbf{s}) \triangleq \{\mathbf{x} \in \mathrm{GF}(2)^n : \mathbf{H}\mathbf{x} = \mathbf{s}\}$$

is called the coset code of $\mathcal{C}$ *with syndrome* $\mathbf{s} \in \mathrm{GF}(2)^k$. *In particular,* $\mathcal{C} = \mathcal{C}(\mathbf{0})$.

A coset code is also described by a translation of the original code. In fact, if $\mathbf{x}' \in \mathrm{GF}(2)^n$ is such that $\mathbf{H}\mathbf{x}' = \mathbf{s}$, then

$$\mathcal{C}(\mathbf{s}) = \{\mathbf{x}' \oplus \mathbf{x} : \mathbf{x} \in \mathcal{C}\}.$$

A sequence $\mathbf{x}' \in \mathcal{C}(\mathbf{s})$ with minimum weight is called a *coset leader* of $\mathcal{C}(\mathbf{s})$. It is possible to show that an $(n, n-k)$ code has 2^k disjoint cosets, which form a partition of $\mathrm{GF}(2)^n$.

Binary LDPC codes are a special class of binary linear codes, characterized by a sparse parity-check matrix $\mathbf{H}$, which contains a much smaller number of ones than zeros. In other words, the parity-check equations defining the code involve only a small number of bits. Rather than specifying the LDPC code in terms of its parity-check matrix, it is convenient to use a graphical representation of $\mathbf{H}$ called the *Tanner graph*. The Tanner

[1] The usual convention is to consider (n, k) block codes so that the number of codewords is 2^k rather than 2^{n-k}; nevertheless, this alternative convention simplifies our notation later on.

$$\mathbf{H} = \begin{pmatrix} 1&1&1&1&0&1&1&0&0&0 \\ 0&0&1&1&1&1&1&1&0&0 \\ 0&1&0&1&0&1&0&1&1&1 \\ 1&0&1&0&1&0&0&1&1&1 \\ 1&1&0&0&1&0&1&0&1&1 \end{pmatrix}$$

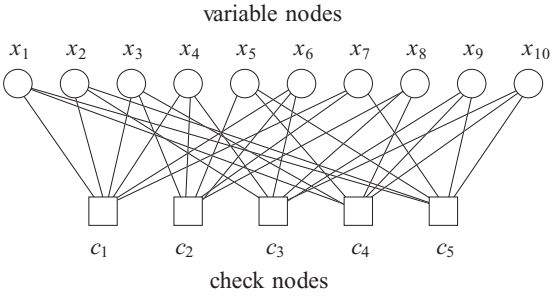

Figure 6.1 A parity-check matrix and its corresponding Tanner graph for a code with blocklength $n = 10$.

graph of $\mathbf{H} \in \mathrm{GF}(2)^{k\times n}$ is a bipartite graph with n *variable nodes* and k *check nodes* connected by *edges*. Each variable node represents a bit x_i in a codeword, while each check node represents a parity-check equation satisfied by the codewords. Specifically, letting $\mathbf{H} = \left(h_{ji}\right)_{kn}$, the jth check node represents the equation

$$c_j = \bigoplus_{i=1}^{n} x_i h_{ji}.$$

An edge connects variable node x_i to check node c_j if and only if x_i is involved in the jth parity-check equation, that is $h_{ji} = 1$. The *degree* of a node is defined as the number of edges incident to it. As an example, Figure 6.1 illustrates a parity-check matrix and its corresponding Tanner graph for a binary linear code of length $n = 10$, in which all variable nodes have degree 3 and all check nodes have degree 6.

Given a Tanner graph, it is possible to compute its *variable-node edge-degree distribution* $\{\lambda_i\}_{i\geqslant 1}$, in which λ_i is the fraction of edges incident on a variable node with degree i. Similarly, the *check-node edge-degree distribution* is $\{\rho_j\}_{j\geqslant 1}$, in which ρ_j is the fraction of edges incident on a check node with degree j. These edge-degree distributions (degree distributions for short) are often represented in compact form by the following polynomials in x:

$$\lambda(x) = \sum_{i\geqslant 1} \lambda_i x^{i-1} \quad \text{and} \quad \rho(x) = \sum_{j\geqslant 1} \rho_j x^{j-1}.$$

The rate of the code is directly related to the edge-degree distribution by

$$R = 1 - \frac{\int_0^1 \rho(x)\mathrm{d}x}{\int_0^1 \lambda(x)\mathrm{d}x}.$$

Note that a parity-check matrix $\mathbf{H}$ specifies a unique Tanner graph and thus a unique degree distribution, whereas a degree distribution specifies only an *ensemble* of codes with the same rate R (for instance, all permutations of nodes in a graph have the same degree distribution). Fortunately, for large block lengths, all codes within a given ensemble have roughly the same decoding performance; hence, LDPC codes are often specified by their degree distributions $(\lambda(x), \rho(x))$ alone.

An LDPC code is called *regular* if all variable nodes have the same degree, and all check nodes have the same degree. Otherwise, it is called *irregular*.

Example 6.1. A rate-$\frac{1}{2}$ regular $(3, 6)$ LDPC code is such that all variable nodes have degree 3, and all check nodes have degree 6. Its degree distributions are simply

$$\lambda(x) = x^2 \quad \text{and} \quad \rho(x) = x^5.$$

Example 6.2. The following irregular degree distributions correspond to another rate-$\frac{1}{2}$ LDPC code:

$$\lambda(x) = 0.106\,257x + 0.486\,659x^2 + 0.010\,390x^{10} + 0.396\,694x^{19},$$
$$\rho(x) = 0.5x^7 + 0.5x^8.$$

6.2.2 Message-passing decoding algorithm

Let $\mathcal{C}$ be an $(n, n-k)$ LDPC code with parity-check matrix $\mathbf{H} \in \mathrm{GF}(2)^{k\times n}$. Consider a codeword $\mathbf{x} = (x_1, \ldots, x_n)^\intercal \in \mathcal{C}$ whose bits are transmitted over a binary-input memoryless channel $(\{0, 1\}, p_{Y|X}(y|x), \mathcal{Y})$. Let $\mathbf{y} = (y_1, \ldots, y_n)^\intercal$ denote the vector of received symbols. The success of LDPC codes stems from the existence of a computationally efficient algorithm to approximate the a-posteriori log-likelihood ratios (LLRs)

$$\lambda_i = \log\left(\frac{\mathbb{P}[\mathsf{X}_i = 0|\mathbf{y}]}{\mathbb{P}[\mathsf{X}_i = 1|\mathbf{y}]}\right) \quad \text{for } i \in [\![1, n]\!].$$

The sign of λ_i provides the most likely value of the bit x_i, while the magnitude $|\lambda_i|$ provides a measure of the reliability associated with the decision. For instance, if $|\lambda_i| = 0$, the bit x_i is equally likely to be zero or one; in contrast, if $|\lambda_i| = \infty$, there is no uncertainty regarding the value of x_i.

For $i \in [\![1, n]\!]$, we let $\mathcal{N}(i)$ denote the indices of check nodes connected to the variable node x_i in the Tanner graph; the set can be obtained from the parity-check matrix $\mathbf{H} = (h_{ji})_{kn}$ as

$$\mathcal{N}(i) \triangleq \{j : h_{ji} = 1\}.$$

Table 6.1 Belief-propagation algorithm

☐ **Initialization.**

► For each $i \in [\![1, n]\!]$ and for each $j \in \mathcal{N}(i)$

$$u_{ji}^{(0)} = v_{ij}^{(0)} = 0.$$

► For each $i \in [\![1, n]\!]$

$$\lambda_i^{\mathrm{INT}} = \log\left(\frac{p_{Y|X}(y_i|0)}{p_{Y|X}(y_i|1)}\right).$$

☐ **Iterations.** For each iteration $l \in [\![1, l_{\max}]\!]$

► For each $i \in [\![1, n]\!]$ and for each $j \in \mathcal{N}(i)$

$$v_{ij}^{(l)} = \lambda_i^{\mathrm{INT}} + \sum_{m \in \mathcal{N}(i)\setminus j} u_{mi}^{(l-1)}.$$

► For each $j \in [\![1, k]\!]$ and for each $i \in \mathcal{M}(j)$

$$u_{ji}^{(l)} = 2\tanh^{-1}\left(\prod_{m \in \mathcal{M}(j)\setminus i} \tanh\left(\frac{v_{mj}^{(l)}}{2}\right)\right).$$

☐ **Extrinsic information**. For all $i \in [\![1, n]\!]$

$$\lambda_i^{\mathrm{EXT}} = \sum_{m \in \mathcal{N}(i)} u_{mi}^{(l_{\max})}.$$

☐ **Hard decisions.** For all $i \in [\![1, n]\!]$

$$\hat{x}_i = \frac{1}{2}\left(1 - \mathrm{sign}\left(\lambda_i^{\mathrm{INT}} + \lambda_i^{\mathrm{EXT}}\right)\right).$$

Similarly, for $j \in [\![1, k]\!]$, we let $\mathcal{M}(j)$ denote the indices of variable nodes connected to the check node c_j in the Tanner graph; that is,

$$\mathcal{M}(j) \triangleq \{i : h_{ji} = 1\}.$$

The LLRs $\{\lambda_i\}_n$ can be approximated using the iterative algorithm described in Table 6.1. This algorithm is called the "belief-propagation" algorithm or the "sum–product" algorithm.

As illustrated in Figure 6.2, the belief-propagation algorithm belongs to the class of "message-passing" algorithms, since the quantities $v_{ij}^{(l)}$ and $u_{ji}^{(l)}$, which are updated at each iteration l, can be understood as "messages" exchanged between the variable nodes and check nodes along the edges of the Tanner graph. The final hard decision for each bit x_i is based on two terms: λ_i^{INT}, which is called the *intrinsic information* (or intrinsic LLR) because it depends only on the current observation y_i; and λ_i^{EXT}, which is called the *extrinsic information* (or extrinsic LLR) because it contains the information about x_i provided by other observations. The usefulness of the algorithm is justified by the following result.

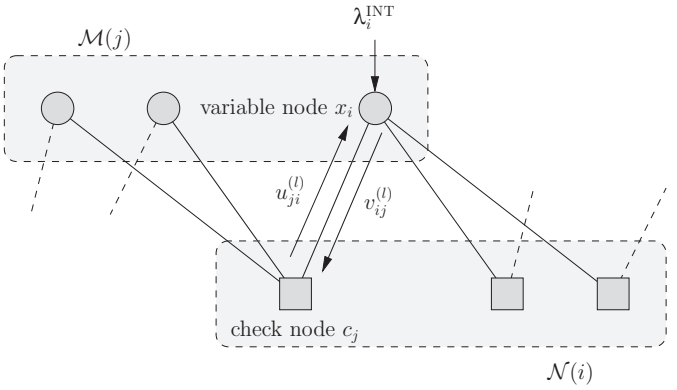

Figure 6.2 Illustration of message-passing behavior for the belief-propagation algorithm.

Theorem 6.2. *If the Tanner graph of an LDPC code does not contain cycles, then the values* $\left\{\lambda_i^{\mathrm{INT}} + \lambda_i^{\mathrm{EXT}}\right\}_n$ *computed by the message-passing algorithm converge to the true a-posteriori LLRs* $\{\lambda_i\}_n$*. The hard decisions are then equivalent to bit-wise maximum a-posteriori (MAP) estimations.*

In practice, even if the Tanner graph contains a few cycles, the message-passing algorithm performs reasonably well. The complexity of the algorithm is linear in the number of edges, and is particularly useful for codes with sparse Tanner graphs, such as LDPC codes.

6.2.3 Properties of LDPC codes under message-passing decoding

Definition 6.4. *Consider a set of binary-input memoryless channels that are all characterized by a parameter* α*. The set of channels is ordered if* $\alpha_1 < \alpha_2$ *implies that the channel with parameter* α_2 *is stochastically degraded with respect to the channel with parameter* α_1*.*

Definition 6.5. *A binary-input memoryless channel* $\left(\{\pm 1\}, p_{Y|X}(y|x), \mathcal{Y}\right)$ *is output-symmetric if the transition probabilities are such that*

$$\forall y \in \mathcal{Y} \quad p_{Y|X}(y|-1) = p_{Y|X}(-y|1).$$

Examples of ordered families of binary-input symmetric-output channels include binary symmetric channels with cross-over probability p and binary erasure channels with erasure probability ϵ (after relabeling of the inputs), and binary-input additive white Gaussian noise channels with noise variance σ^2 under the same input power constraint.

Theorem 6.3 (Richardson and Urbanke)**.** *Consider an LDPC code of length n chosen uniformly at random from the ensemble of codes with degree distributions* $(\lambda(x), \rho(x))$ *and used over an ordered family of binary-input symmetric-output channels with parameter* α*. Then, the following results hold.*

1. ***Convergence to the cycle-free case:*** *as n goes to infinity, the Tanner graph becomes cycle-free.*
2. ***Concentration around the average:*** *as n goes to infinity, the error-decoding capability of the code under message-passing decoding converges to the average error-decoding capability of the code ensemble.*
3. ***Threshold behavior:*** *there exists a channel parameter α^*, called the* threshold*, such that the bit error probability goes to zero as n goes to infinity if and only if $\alpha < \alpha^*$.*

Theorem 6.3 simplifies the design of LDPC codes tremendously because it states that it is sufficient to analyze the average performance of an ensemble of LDPC code with given degree distributions $(\lambda(x), \rho(x))$ rather than focus on an individual code. For large length n, the probability of error with belief-propagation decoding of most codes in the ensemble is close to the probability of error averaged over the ensemble; hence, to construct good codes with rate R, it suffices to optimize the degree distributions $(\lambda(x), \rho(x))$ so as to maximize the threshold α^*. This optimization is, in general, non-convex in the degree distribution, but it can be numerically solved by combining an efficient algorithm to compute the threshold of given degree distributions $(\lambda(x), \rho(x))$, called density evolution, and a heuristic genetic optimization algorithm, called differential evolution.

Example 6.3. Consider the family of binary erasure channels with parameter α, the erasure probability. The threshold of a regular $(3, 6)$ LDPC code is $\alpha^* \approx 0.42$. The threshold of a rate-$\frac{1}{2}$ irregular code with distribution as in Example 6.2 is $\alpha^* \approx 0.4741$.

Note that Theorem 6.3 characterizes a threshold for the asymptotic behavior of the *bit error probability*. This result can be refined and, as stated in the following theorem, one can show that the same threshold also characterizes the behavior of the *block error probability* for some LDPC ensembles.

Theorem 6.4 (Jin and Richardson)**.** *If the degree distributions $(\lambda(x), \rho(x))$ of an LDPC ensemble do not contain any variable nodes of degree 2, then the bit error probability threshold is also the block error probability threshold.*

LDPC ensembles with high thresholds usually have a high fraction of nodes of degree 2, and it may seem that Theorem 6.4 prevents us from obtaining high thresholds for the block error probability. However, it is possible to strengthen Theorem 6.4 to accomodate some fraction of degree-2 nodes. The presentation of this result goes beyond the scope of this chapter and we refer the interested reader to [109] for more details.

6.3 Secrecy codes for the binary erasure wiretap channel

In this section, we restrict our attention to the binary erasure wiretap channel illustrated in Figure 6.3, in which the main channel is noiseless while the eavesdropper's channel is a BEC with erasure probability ϵ. From Corollary 3.1, the secrecy capacity of this

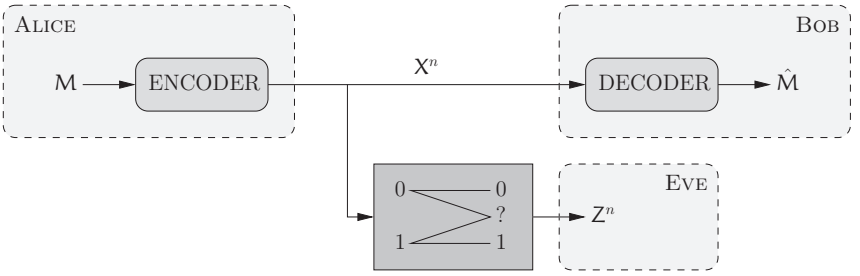

Figure 6.3 Binary erasure wiretap channel.

channel is $C_s = 1 - (1 - \epsilon) = \epsilon$. Following the observations made in Section 6.1, we investigate a code construction based on nested codes. However, since any codeword sent by the transmitter is received without errors by the legitimate receiver, the construction is much simpler than in the general case: for any set of disjoint subcodes $\{\mathcal{C}_i\}$, the mother code $\mathcal{C} = \bigcup_i \mathcal{C}_i$ is always a reliable code for the (noiseless) main channel.

A set of subcodes that leads to a particularly simple stochastic encoder consists of an $(n, n-k)$ binary linear code $\mathcal{C}_0$ and its cosets. For this choice of subcodes, the mother code $\mathcal{C}$ is

$$\mathcal{C} = \bigcup_{\mathbf{s} \in \mathrm{GF}(2)^k} \mathcal{C}_0(\mathbf{s}) = \mathrm{GF}(2)^n.$$

The corresponding stochastic encoding procedure is called *coset coding*, and it consists of encoding a message $\mathbf{m} \in \mathrm{GF}(2)^k$ by selecting a codeword uniformly at random in the coset code $\mathcal{C}_0(\mathbf{m})$ of $\mathcal{C}_0$ with syndrome $\mathbf{m}$. The following proposition shows that the encoding and decoding operations in coset coding can be implemented efficiently with matrix multiplications.

Proposition 6.1. *Let $\mathcal{C}_0$ be an $(n, n-k)$ binary linear code. Then there exists a generator matrix $\mathbf{G} \in \mathrm{GF}(2)^{n-k \times n}$ and a parity-check matrix $\mathbf{H} \in \mathrm{GF}(2)^{k \times n}$ for $\mathcal{C}_0$ and a matrix $\mathbf{G}' \in \mathrm{GF}(2)^{k \times n}$ such that*

- *the encoder maps a message $\mathbf{m}$ to a codeword as*

$$\mathbf{m} \mapsto (\mathbf{G}'^{\mathsf{T}} \quad \mathbf{G}^{\mathsf{T}}) \begin{pmatrix} \mathbf{m} \\ \mathbf{v} \end{pmatrix},$$

  where the vector $\mathbf{v} \in \mathrm{GF}(2)^{n-k}$ is chosen uniformly at random;
- *the decoder maps a codeword $\mathbf{x}$ to a message as*

$$\mathbf{x} \mapsto \mathbf{H}\mathbf{x}.$$

Proof. Let $\{\mathbf{g}_i\}_{n-k}$ be a basis of $\mathcal{C}_0$ and let $\mathbf{G}$ be a generator matrix whose rows are the vectors $\{\mathbf{g}_i^{\mathsf{T}}\}_{n-k}$. The set $\{\mathbf{g}_i\}_{n-k}$, which is linearly independent, can be completed by a linearly independent set $\{\mathbf{h}_i\}_k$ to obtain a basis of $\mathrm{GF}(2)^n$. Let $\mathbf{G}'$ be a matrix whose rows are the vectors $\{\mathbf{h}_i^{\mathsf{T}}\}_k$.

Let us now consider the encoding of a message $\mathbf{m} \in \mathrm{GF}(2)^k$ as

$$\mathbf{m} \mapsto (\mathbf{G}'^{\mathsf{T}} \;\; \mathbf{G}^{\mathsf{T}}) \begin{pmatrix} \mathbf{m} \\ \mathbf{v} \end{pmatrix} = \sum_{i=1}^{k} m_i \mathbf{h}_i + \sum_{i=1}^{n-k} v_i \mathbf{g}_i, \tag{6.1}$$

where $\mathbf{v} \in \mathrm{GF}(2)^{n-k}$ is chosen uniformly at random. The term $\sum_{i=1}^{n-k} v_i \mathbf{g}_i$ corresponds to the choice of a codeword uniformly at random in $\mathcal{C}_0$; therefore, the operation (6.1) is equivalent to coset coding if we can prove that two different messages $\mathbf{m}$ and $\mathbf{m}'$ generate sequences in different cosets of $\mathcal{C}_0$. Assume that two messages $\mathbf{m}$ and $\mathbf{m}'$ are encoded as sequences $\mathbf{x}$ and $\mathbf{x}'$ in the same coset with coset leader $\mathbf{e} \in \mathrm{GF}(2)^n$. Then, there exist codewords $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}_0$ such that

$$\mathbf{x} = \mathbf{e} + \mathbf{c}_1 \quad \text{and} \quad \mathbf{x}' = \mathbf{e} + \mathbf{c}_2.$$

Consequently, $\mathbf{x} + \mathbf{x}' = \mathbf{c}_1 + \mathbf{c}_2 \in \mathcal{C}_0$, which is impossible unless $\mathbf{m} = \mathbf{m}'$.

It remains to prove that there exists a parity-check matrix $\mathbf{H}$ such that $\mathbf{Hx} = \mathbf{m}$. Let $\mathbf{H}'$ be an arbitrary parity-check matrix of $\mathcal{C}_0$. In general, if $\mathbf{x}$ is obtained from $\mathbf{m}$ according to (6.1), then $\mathbf{H}'\mathbf{x} \neq \mathbf{m}$. However, the application $\mathbf{m} \mapsto \mathbf{H}'\mathbf{x}$ is injective (and hence bijective); hence, there exists an invertible matrix $\mathbf{A} \in \mathrm{GF}(2)^{k \times k}$ such that

$$\mathbf{AH}'\mathbf{x} = \mathbf{m}.$$

Note that $\mathbf{H} = \mathbf{AH}'$ is another parity-check matrix for $\mathcal{C}_0$ and is such that $\mathbf{Hx} = \mathbf{m}$. □

6.3.1 Algebraic secrecy criterion

Since coset coding can be defined in terms of the parity-check matrix and generator matrix of a linear code $\mathcal{C}_0$, the algebraic structure of the linear code is likely to have a critical effect on secrecy. Hence, to clarify this connection and simplify the analysis, it is convenient to develop an *algebraic secrecy criterion* equivalent to the original information-theoretic secrecy criterion for coset coding.

Consider an eavesdropper's observation $\mathbf{z}$ with μ unerased bits in positions $(i_1, \ldots, i_\mu)$. If a sequence $\mathbf{x} \in \mathrm{GF}(2)^n$ is such that

$$(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_\mu}) = (\mathbf{z}_{i_1}, \ldots, \mathbf{z}_{i_\mu}),$$

then the sequence is said to be *consistent* with $\mathbf{z}$. If a coset of $\mathcal{C}_0$ contains at least one sequence $\mathbf{x}$ that is consistent with $\mathbf{z}$, then the coset itself is said to be consistent with $\mathbf{z}$. The total number of cosets of $\mathcal{C}_0$ consistent with $\mathbf{z}$ is denoted by $N(\mathcal{C}_0, \mathbf{z})$.

Lemma 6.1. *Let $\mathbf{z}$ be an eavesdropper's observation at the output of the BEC. Then all cosets of $\mathcal{C}_0$ that are consistent with $\mathbf{z}$ contain the same number of sequences consistent with $\mathbf{z}$.*

Proof. Let $(i_1, \ldots, i_\mu)$ denote the set of unerased positions of $\mathbf{z}$. Let $\mathbf{G} \in \mathrm{GF}(2)^{n-k \times n}$ be a generator matrix of $\mathcal{C}_0$, and let $\mathbf{g}_i$ denote the ith column of $\mathbf{G}$. We define the matrix $\mathbf{G}_\mu$ as

$$\mathbf{G}_\mu \triangleq \left(\mathbf{g}_{i_1} \; \mathbf{g}_{i_2} \ldots \mathbf{g}_{i_\mu}\right).$$

Let $\mathcal{C}_1$ be a coset of $\mathcal{C}_0$ consistent with $\mathbf{z}$. Define $\mathfrak{C}_1(\mathbf{z}) \triangleq \{\mathbf{x} \in \mathcal{C}_1 : \mathbf{x} \text{ consistent with } \mathbf{z}\}$ and let $\mathbf{x}_1 \in \mathfrak{C}_1(\mathbf{z})$. We show that $|\mathfrak{C}_1(\mathbf{z})| = \left|\text{Ker}(\mathbf{G}_\mu)\right|$.

- For any $\mathbf{m} \in \text{Ker}(\mathbf{G}_\mu)$, the vector $\mathbf{G}^\intercal\mathbf{m}$ is in $\mathcal{C}_0$ and contains zeros in positions $(i_1, \ldots, i_\mu)$. Therefore, for any $\mathbf{m} \in \text{Ker}(\mathbf{G}_\mu)$, $\mathbf{x}_1 + \mathbf{G}^\intercal\mathbf{m} \in \mathfrak{C}_1(\mathbf{z})$ and
$$\left\{\mathbf{x}_1 + \mathbf{G}^\intercal\mathbf{m} : \mathbf{m} \in \text{Ker}(\mathbf{G}_\mu)\right\} \subseteq \mathfrak{C}_1(\mathbf{z}).$$
- Now, assume $\mathbf{x}_1' \in \mathfrak{C}_1(\mathbf{z})$. Then $\mathbf{x}_1' + \mathbf{x}_1 \in \mathcal{C}_0$ and contains zeros in positions $(i_1, \ldots, i_\mu)$. Hence, there exists $\mathbf{m} \in \text{Ker}(\mathbf{G}_\mu)$ such that $\mathbf{x}_1' + \mathbf{x}_1 = \mathbf{G}^\intercal\mathbf{m}$. Therefore, $\mathbf{x}_1' \in \left\{\mathbf{x}_1 + \mathbf{G}^\intercal\mathbf{m} : \mathbf{m} \in \text{Ker}(\mathbf{G}_\mu)\right\}$ and
$$\mathfrak{C}_1(\mathbf{z}) \subseteq \left\{\mathbf{x}_1 + \mathbf{G}^\intercal\mathbf{m} : \mathbf{m} \in \text{Ker}(\mathbf{G}_\mu)\right\}.$$

Hence, $\mathfrak{C}_1(\mathbf{z}) = \left\{\mathbf{x}_1 + \mathbf{G}^\intercal\mathbf{m} : \mathbf{m} \in \text{Ker}(\mathbf{G}_\mu)\right\}$ and $|\mathfrak{C}_1(\mathbf{z})| = \left|\text{Ker}(\mathbf{G}_\mu)\right|$. Therefore, any coset of $\mathcal{C}_0$ consistent with $\mathbf{z}$ contains exactly $\left|\text{Ker}(\mathbf{G}_\mu)\right|$ sequences consistent with $\mathbf{z}$. □

Proposition 6.2. *Let* $\mathbf{z}$ *be an eavesdropper's observation at the output of the BEC. Then the eavesdropper's uncertainty about* M *given his observation* $\mathbf{z}$ *is*
$$\mathbb{H}(\mathsf{M}|\mathbf{z}) = \log N(\mathcal{C}_0, \mathbf{z}).$$

Proof. Let X^n be the random variable that represents the sequence sent over the channel. Then,
$$\begin{aligned}\mathbb{H}(\mathsf{M}|\mathbf{z}) &= \mathbb{H}(\mathsf{M}\mathsf{X}^n|\mathbf{z}) - \mathbb{H}(\mathsf{X}^n|\mathsf{M}\mathbf{z}) \\ &= \mathbb{H}(\mathsf{X}^n|\mathbf{z}) - \mathbb{H}(\mathsf{X}^n|\mathsf{M}\mathbf{z}).\end{aligned}$$
The term $\mathbb{H}(\mathsf{X}^n|\mathbf{z})$ is the uncertainty in the codeword that was sent given the observation $\mathbf{z}$. By virtue of the definition of coset coding, all codewords are used with equal probability; therefore, $\mathbb{H}(\mathsf{X}^n|\mathbf{z}) = \log N$, where N is the number of sequences that are consistent with $\mathbf{z}$. Now,
$$\mathbb{H}(\mathsf{X}^n|\mathsf{M}\mathbf{z}) = \sum_m \mathbb{H}(\mathsf{X}^n|\mathsf{M} = m, \mathbf{z}) p_{\mathsf{M}|\mathsf{Z}}(m|\mathbf{z}),$$
and the term $\mathbb{H}(\mathsf{X}^n|\mathsf{M} = m, \mathbf{z})$ is the uncertainty in the sequence that was sent given $\mathbf{z}$ and knowing the coset m that was used. By definition, all codewords are used with equal probability and, by Lemma 6.1, all cosets consistent with $\mathbf{z}$ contain the same number of sequences consistent with $\mathbf{z}$; hence, $\mathbb{H}(\mathsf{X}^n|\mathsf{M} = m, \mathbf{z}) = \log N_\text{c}$, where N_c is the number of sequences consistent with $\mathbf{z}$ in a coset consistent with $\mathbf{z}$. Therefore,
$$\mathbb{H}(\mathsf{M}|\mathbf{z}) = \log N - \log N_\text{c} = \log\left(\frac{N}{N_\text{c}}\right) = \log N(\mathcal{C}_0, \mathbf{z}).$$ □

The number of cosets is 2^k, therefore $N(\mathcal{C}_0, \mathbf{z}) \leqslant 2^k$. If $N(\mathcal{C}_0, \mathbf{z}) = 2^k$, then all cosets are consistent with $\mathbf{z}$ and we say that $\mathbf{z}$ is *secured* by the code $\mathcal{C}_0$. The following theorem provides a necessary and sufficient condition for an observation $\mathbf{z}$ to be secured by $\mathcal{C}_0$.

Proposition 6.3 (Ozarow and Wyner)**.** *Let* $\mathcal{C}_0$ *be an* $(n, n-k)$ *binary linear code with generator matrix* $\mathbf{G}$*, and let* $\mathbf{g}_i$ *denote the* i*th column of* $\mathbf{G}$*. Let* $\mathbf{z}$ *be an observation of*

the eavesdropper with μ unerased bits in positions $(i_1, \ldots, i_\mu)$. Then $\mathbf{z}$ is secured by $\mathcal{C}_0$ if and only if the matrix

$$\mathbf{G}_\mu \triangleq (\mathbf{g}_{i_1}\ \mathbf{g}_{i_2} \ldots \mathbf{g}_{i_\mu})$$

has rank μ.

Proof. Assume that $\mathbf{G}_\mu$ has rank μ. Then, by definition, the code $\mathcal{C}_0$ has codewords with all possible sequences of $\mathrm{GF}(2)^\mu$ in the μ unerased positions. Since cosets are obtained by translating $\mathcal{C}_0$, all cosets also have codewords with all possible binary sequences in the μ unerased positions. Therefore, $N(\mathcal{C}_0, \mathbf{z}) = 2^k$.

Now, assume that $\mathbf{G}_\mu$ has rank strictly less than μ. Then, there exists at least one sequence of μ bits $\mathbf{c}_\mu$ that does not appear in any codeword of $\mathcal{C}_0$ in the μ unerased positions. For any sequence $\mathbf{x}' \in \mathrm{GF}(2)^n$, we let $\mathbf{x}'_\mu$ denote the bits of $\mathbf{x}'$ in the μ unerased positions. Since the cosets form a partition of $\mathrm{GF}(2)^n$, there exists a sequence $\mathbf{x}'$ in a coset $\mathcal{C}'$ such that $\mathbf{x}'_\mu \oplus \mathbf{c}_\mu$ has the same value as $\mathbf{z}$ in the unerased positions. Since $\mathbf{c}_\mu$ does not appear in any codeword of $\mathcal{C}_0$, the coset $\mathcal{C}'$ is not consistent with $\mathbf{z}$; therefore $N(\mathcal{C}_0, \mathbf{z}) < 2^k$. □

As an immediate consequence of Proposition 6.3, we obtain a necessary and sufficient condition for communication in perfect secrecy with respect to an eavesdropper who observes any set of μ unerased bits.

Corollary 6.1. *Let $\mathcal{C}_0$ be an $(n, n-k)$ binary linear code with generator matrix $\mathbf{G}$. Coset coding with $\mathcal{C}_0$ guarantees perfect secrecy against an eavesdropper who observes* any *set of μ unerased bits if and only if all submatrices of $\mathbf{G}$ with μ columns have rank μ.*

Proof. If all submatrices of $\mathbf{G}$ with μ columns have rank μ, then any observation $\mathbf{z}$ with μ unerased positions is secured by $\mathcal{C}_0$ and $\mathbb{H}(\mathsf{M}|\mathbf{z}) = k$; therefore,

$$\begin{aligned}\mathbb{I}(\mathsf{M}; \mathsf{Z}^n) &= \mathbb{H}(\mathsf{M}) - \mathbb{H}(\mathsf{M}|\mathsf{Z}^n) \\ &= k - \sum_{\mathbf{z}} p_{\mathsf{Z}^n}(\mathbf{z})\mathbb{H}(\mathsf{M}|\mathbf{z}) \\ &= 0.\end{aligned}$$

Conversely, if there exists a submatrix of $\mathbf{G}$ with μ columns and rank less than μ, then there exists an observation $\mathbf{z}'$ that is not secured by $\mathcal{C}_0$ and such that $\mathbb{H}(\mathsf{M}|\mathbf{z}') < k$. Therefore,

$$\begin{aligned}\mathbb{I}(\mathsf{M}; \mathsf{Z}^n) &= \mathbb{H}(\mathsf{M}) - \mathbb{H}(\mathsf{M}|\mathsf{Z}^n) \\ &= k - \sum_{\mathbf{z}} p_{\mathsf{Z}^n}(\mathbf{z})\mathbb{H}(\mathsf{M}|\mathbf{z}) \\ &> 0.\end{aligned}$$

□

Remark 6.2. *A binary wiretap channel model in which the eavesdropper is known to access no more than μ of n transmitted bits is called a* wiretap channel of type II.

This model differs from the binary erasure wiretap channel of Figure 6.3 in that the eavesdropper can in principle choose which μ bits are observed. This model was extensively studied by Ozarow and Wyner, and we discuss it in the context of network coding in Chapter 9.

6.3.2 Coset coding with dual of LDPC codes

We now go back to the binary erasure wiretap channel of Figure 6.3. In general, we cannot guarantee that the eavesdropper's observation contains a fixed number E of unerased symbols with probability one; however, by Chebyshev's inequality,

$$\forall \beta > 0 \quad \mathbb{P}\left[\left|\frac{\mathsf{E}}{n} - (1-\epsilon)\right| > \beta\right] \leqslant \frac{\operatorname{Var}\mathsf{E}}{n^2\beta^2} = \frac{\epsilon(1-\epsilon)}{n\beta^2}.$$

In other words, the fraction of unerased positions is arbitrarily close to $(1-\epsilon)$ with high probability as n becomes large. Consequently, we will be able to leverage the results of Section 6.3.1 if we can find a suitable matrix $\mathbf{G}$ for coset coding such that the observations of the eavesdropper are secured with high probability. It turns out that *parity-check* matrices of LDPC codes satisfy the desired condition. In fact, the threshold property of decoding under message-passing can be interpreted as follows.

Lemma 6.2. *Let $\mathbf{H}$ be the parity-check matrix of a length-n LDPC code selected uniformly at random in an ensemble whose block error probability threshold under belief-propagation decoding for the erasure channel is α^*. Form a submatrix $\mathbf{H}'$ of $\mathbf{H}$ by selecting each column of $\mathbf{H}$ with probability $\alpha < \alpha^*$. Then,*

$$\mathbb{P}\left[\operatorname{rk}(\mathbf{H}') = \alpha n\right] = 1 - \delta(n).$$

Proof. By Theorem 6.3 and Theorem 6.4, with probability $1-\delta(n)$, $\mathbf{H}$ is such that the block error probability under message-passing decoding vanishes as n goes to infinity if $\alpha < \alpha^*$. In other words, if the erased bits in a given observation $\mathbf{z}$ are treated as unknown, the equation $\mathbf{Hz} = \mathbf{0}$ has a unique solution. Without loss of generality, we assume that the first bits of $\mathbf{z}$ are erased and rewrite the equation $\mathbf{Hz} = \mathbf{0}$ as $(\mathbf{H}'\ \mathbf{H}'')\mathbf{z} = \mathbf{0}$, where $\mathbf{H}'$ corresponds to the erased position of $\mathbf{z}$. This equation has a unique solution if and only if $\mathbf{H}'$ has full column rank αn. □

Theorem 6.5 (Thangaraj *et al.*)**.** *Let $\mathcal{C}_0$ be an $(n, n-k)$ binary LDPC code selected uniformly at random in an ensemble with erasure threshold α^* and let $\mathcal{C}_0^\perp$ be its dual. Then, as n goes to infinity, coset coding with $\mathcal{C}_0^\perp$ and its cosets guarantees (weak) secrecy at rate $R = k/n$ over any binary erasure wiretap channel with erasure probability $\epsilon > 1 - \alpha^*$.*

Proof. By definition, a generator matrix $\mathbf{G}$ of $\mathcal{C}_0^\perp$ is a parity-check matrix $\mathbf{H}$ of $\mathcal{C}_0$. Therefore, by Lemma 6.2, if $1-\epsilon < \alpha^*$, any submatrix formed by selecting columns of $\mathbf{G}$ with probability $1-\epsilon$ has full column rank with probability $1-\delta(n)$; hence, by Proposition 6.3, any observation Z^n of the eavesdropper is secured by $\mathcal{C}_0^\perp$ with probability

$1-\delta(n)$. Equivalently, if we let Q be the random variable defined as

$$\mathrm{Q} \triangleq \begin{cases} 1 & \text{if } Z^n \text{ is secured by } \mathcal{C}_0^{\perp}, \\ 0 & \text{else,} \end{cases}$$

then $\mathbb{P}[\mathrm{Q}=1] \geqslant 1-\delta(n)$. Consequently,

$$\begin{aligned} \frac{1}{n}\mathbb{I}(\mathrm{M};Z^n) &\leqslant \frac{1}{n}\mathbb{I}(\mathrm{M};Z^n\mathrm{Q}) \\ &\leqslant \frac{1}{n}\left(\mathbb{H}(\mathrm{M}) - \mathbb{H}(\mathrm{M}|Z^n\mathrm{Q})\right) \\ &= \frac{1}{n}\left(\mathbb{H}(\mathrm{M}) - \mathbb{H}(\mathrm{M}|Z^n\mathrm{Q}=1)\mathbb{P}[\mathrm{Q}=1] - \mathbb{H}(\mathrm{M}|Z^n\mathrm{Q}=0)\mathbb{P}[\mathrm{Q}=0]\right) \\ &\leqslant \frac{1}{n}\left(\mathbb{H}(\mathrm{M}) - \mathbb{H}(\mathrm{M})\left(1-\delta(n)\right)\right) \\ &\leqslant \frac{\mathbb{H}(\mathrm{M})}{n}\delta(n) \\ &\leqslant \delta(n). \end{aligned}$$

In other words, coset coding with $\mathcal{C}_0^{\perp}$ guarantees weak secrecy at rate k/n. □

Note that the analysis of coset coding with the dual of LDPC codes does not rely on any capacity-achieving property; our proof relies solely on the concentration and threshold properties established by Theorem 6.3 and Theorem 6.4. Nevertheless, as illustrated by the following examples, the price paid is that we cannot achieve rates arbitrarily close to the secrecy capacity.

Example 6.4. A rate-$\frac{1}{2}$ $(3, 6)$ regular LDPC code, with erasure threshold $\alpha^* \approx 0.42$, can be used for secure communication over any binary erasure wiretap channel with erasure probability $\epsilon > 1-\alpha^* = 0.58$. The communication rate is 0.5, which is at most 86% of the secrecy capacity.

6.3.3 Degrading erasure channels

As shown in Proposition 3.3, a wiretap code designed for a specific eavesdropper's channel $\left(\mathcal{X}, p_{\mathrm{Z}|\mathrm{X}}, \mathcal{Z}\right)$ can be used over any other eavesdropper's channel that is stochastically degraded with respect to $\left(\mathcal{X}, p_{\mathrm{Z}|\mathrm{X}}, \mathcal{Z}\right)$. Codes designed for an erasure eavesdropper's channel are therefore useful over a much broader class of channels. Nevertheless, rates are then bounded strictly below the secrecy capacity since the full characteristics of the eavesdropper's channel are not necessarily exploited. The following proposition shows that all binary-input channels are stochastically degraded with respect to some binary erasure channels.

Proposition 6.4. *A memoryless binary-input channel* $(\{0,1\}, p_{\mathrm{Y}|\mathrm{X}}, \mathcal{Y})$ *(the alphabet* $\mathcal{Y}$ *may be continuous or finite) is stochastically degraded with respect to an erasure*

channel with erasure probability

$$\epsilon = \int_{\mathcal{Y}} \left(\min_{u \in \{0,1\}} p_{Y|X}(y|u) \right) \mathrm{d}y.$$

Proof. Since $\int_{\mathcal{Y}} p_{Y|X}(y|x)\mathrm{d}y = 1$ for any x and $p_{Y|X}(y|x) \geqslant 0$ for any x and y, it is clear that $\epsilon \in [0, 1]$. Let $(\{0, 1\}, p_{Z|X}, \{0, 1, ?\})$ be an erasure channel with erasure probability ϵ defined as above, that is

$$\forall x \in \{0, 1\} \qquad p_{Z|X}(?|x) = \epsilon \quad \text{and} \quad p_{Z|X}(x|x) = 1 - \epsilon.$$

Consider the channel $(\{0, 1, ?\}, \mathcal{Y}, p_{Y|Z}(y|z))$ such that

$$p_{Y|Z}(y|?) = \frac{1}{\epsilon} \min_{u \in \{0,1\}} p_{Y|X}(y|u),$$

$$p_{Y|Z}(y|z) = \frac{1}{1-\epsilon} \left(p_{Y|X}(y|z) - \min_{u \in \{0,1\}} p_{Y|X}(y|u) \right) \quad \text{if } z \in \{0, 1\}.$$

One can check that these are valid transition probabilities with the value of ϵ above. In addition, for any $(x, y) \in \{0, 1\} \times \mathcal{Y}$,

$$\begin{aligned} \sum_{z \in \{0,1,?\}} p_{Y|Z}(y|z) p_{Z|X}(z|x) &= p_{Y|Z}(y|?)\,\epsilon + \sum_{z \in \{0,1\}} p_{Y|Z}(y|z)\,(1-\epsilon)\mathbb{1}(z - x) \\ &= \min_{u \in \{0,1\}} p_{Y|X}(y|u) + p_{Y|X}(y|x) - \min_{u \in \{0,1\}} p_{Y|X}(y|u) \\ &= p_{Y|X}(y|x). \end{aligned}$$

Therefore, the channel $\big(\{0, 1\}, p_{Y|X}, \mathcal{Y}\big)$ is stochastically degraded with respect to a binary erasure channel with probability ϵ. □

The following examples are direct applications of Proposition 6.4 and Proposition 3.3.

Example 6.5. Consider a binary symmetric wiretap channel, in which the main channel is noiseless and the eavesdropper's channel is binary symmetric with cross-over probability $p < \frac{1}{2}$. A code designed for an erasure wiretap channel with erasure probability $\epsilon^* = 2p$ could achieve a secure rate ϵ^*. Figure 6.4 shows the secrecy capacity $C_{\mathrm{s}} = \mathbb{H}_{\mathrm{b}}(p)$ and the achievable rates as a function of the cross-over probability p. For $p = 0.29$, $\epsilon^* = 0.58$, and we can use the irregular code of Example 6.1 and its cosets for secure communication. Note that the rate of secure communication is 0.5 bits per channel use, compared with the secrecy capacity $C_{\mathrm{s}} \approx 0.86$ bits per channel use.

Example 6.6. Consider a Gaussian wiretap channel in which the main channel is noiseless and the eavesdropper's channel is an AWGN channel with noise variance σ^2. Let us restrict our attention to binary inputs $x \in \{-1; +1\}$. A code designed for an erasure wiretap channel with erasure probability

$$\epsilon^* = \operatorname{erfc}\left(\frac{1}{\sqrt{2\sigma^2}} \right)$$

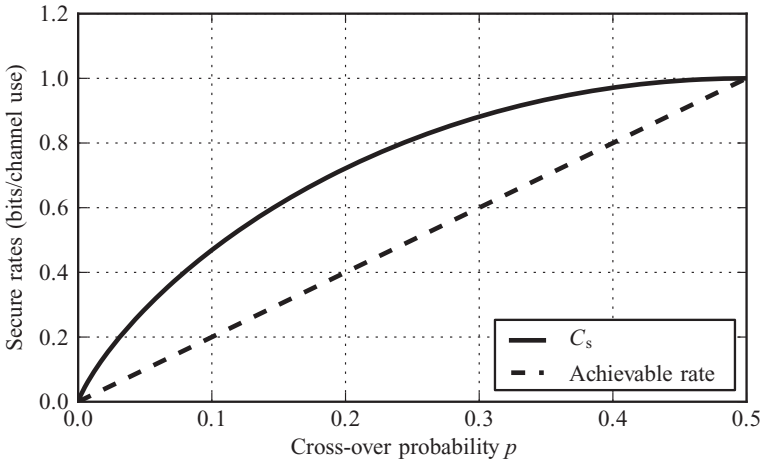

Figure 6.4 Secrecy capacity and achievable rates for a WTC with noiseless main channel and BSP(p) eavesdropper's channel.

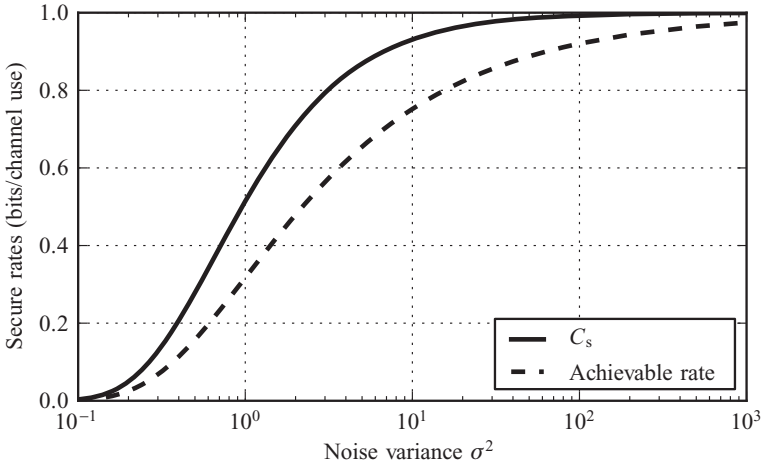

Figure 6.5 Secrecy capacity and achievable rates for a WTC with noiseless main channel and binary-input Gaussian eavesdropper's channel.

will achieve a secure rate ϵ^*. Figure 6.5 shows the secrecy capacity of the binary-input Gaussian wiretap channel and the achievable rates as a function of the eavesdropper's channel variance σ^2. For $\sigma^2 = 3.28$, $\epsilon^* \approx 0.58$, and we can again use the irregular code of Example 6.1. The communication rate is 0.5 bits per channel use, compared with the secrecy capacity $C_s \approx 0.81$ bits per channel use.

6.4 Reconciliation of binary memoryless sources

As seen in Section 4.3, one can generate secret keys from a DMS by performing information reconciliation followed by privacy amplification. Hash functions for privacy

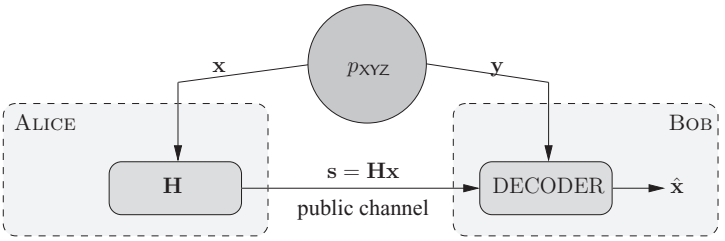

Figure 6.6 Source coding with side information with syndromes of linear codes.

amplification are already known (see for instance Example 4.5 and Example 4.6), and the only missing piece for implementing a complete key-distillation strategy is an efficient information-reconciliation protocol. For clarity, this section focuses solely on binary memoryless sources and the extension to arbitrary discrete sources and continuous sources is relegated to Section 6.5.

Since Proposition 4.5 shows that, without loss of optimality, the reconciliation of discrete random variables can be treated as a problem of source coding with side information, we need only design an encoder for a binary memoryless source X so that a receiver, who has access to a correlated binary memoryless source Y, retrieves X with arbitrarily small probability of error. The Slepian–Wolf theorem (Theorem 2.10) guarantees the existence of codes compressing at a rate arbitrarily close to $\mathbb{H}(\mathsf{X}|\mathsf{Y})$ coded bits, but does not provide explicit constructions. As illustrated in Figure 6.6, one can build source encoders from well-chosen linear codes. Given the parity-check matrix $\mathbf{H} \in \mathrm{GF}(2)^{k\times n}$ of a linear code, a vector of n observations $\mathbf{x}$ is encoded by computing the syndrome $\mathbf{s} = \mathbf{H}\mathbf{x}$. Upon reception of $\mathbf{s}$, the decoder can minimize its probability of error by looking for the sequence $\mathbf{x}$ that maximizes the a-posteriori probability

$$\mathbb{P}[\mathbf{y}|\mathbf{x}, \mathbf{s} = \mathbf{H}\mathbf{x}].$$

This procedure is equivalent to maximum a-posteriori (MAP) estimation of $\mathbf{x}$ within the coset code with syndrome $\mathbf{s}$. One can show that there exist good linear codes such that the probability of error can be made as small as desired, provided that the number of syndrome bits k is at least $n\mathbb{H}(\mathsf{X}|\mathsf{Y})$. Note that the code rate of the linear code used for syndrome coding is $1 - k/n$, while the compression rate is k/n; in the remainder of this chapter, we explicitly specify which rate is being considered in order to avoid confusion.

In practice, the computational complexity required to perform MAP decoding is prohibitive, but we can use LDPC codes and adapt the message-passing algorithm described in Table 6.1 to obtain a suboptimal yet efficient algorithm. Essentially, given a syndrome $\mathbf{s}$ and a sequence of observations $\mathbf{y}$, the key idea is to slightly modify the intrinsic LLRs to account for the value of the syndrome and the a-priori distribution of X. To describe the modified message-passing algorithm, we introduce the following notation. The n-bit vector observed by the encoder is denoted $\mathbf{x} = (x_1 \dots x_n)^\intercal$, while the n-bit vector available to the decoder as side information is denoted $\mathbf{y} = (y_1 \dots y_n)^\intercal$. The decoder receives the syndrome vector $\mathbf{s} = (s_1 \dots s_k)^\intercal$, whose entries correspond to the values of check nodes $\{c_1, \dots, c_k\}$ in the Tanner graph of the LDPC code. The set of

Table 6.2 Belief-propagation algorithm for source coding with side information

☐ **Initialization**.

► For each $i \in [\![1, n]\!]$ and for each $j \in \mathcal{N}(i)$

$$v_{ij}^{(0)} = u_{ji}^{(0)} = 0.$$

► For each $i \in [\![1, n]\!]$

$$\lambda_i^{\mathrm{INT}} = \log\left(\frac{p_{\mathsf{XY}}(0, y_i)}{p_{\mathsf{XY}}(1, y_i)}\right).$$

☐ **Iterations**. For each iteration $l \in [\![1, l_{\max}]\!]$

► For each $i \in [\![1, n]\!]$ and for each $j \in \mathcal{N}(i)$

$$v_{ij}^{(l)} = \lambda_i^{\mathrm{INT}} + \sum_{m \in \mathcal{N}(i)\setminus j} u_{mj}^{(l-1)}.$$

► For each $j \in [\![1, k]\!]$ and $i \in \mathcal{M}(j)$

$$u_{ji}^{(l)} = 2\tanh^{-1}\left((1-2s_j)\prod_{m \in \mathcal{M}(j)\setminus i}\tanh\left(\frac{v_{mj}^{(l)}}{2}\right)\right).$$

☐ **Extrinsic information**. For all $i \in [\![1, n]\!]$

$$\lambda_i^{\mathrm{EXT}} = \sum_{m \in \mathcal{N}(i)} u_{mi}^{(l_{\max})}.$$

☐ **Hard decisions.** For all $i \in [\![1, n]\!]$

$$\hat{x}_i = \frac{1}{2}\left(1 - \operatorname{sign}\left(\lambda_i^{\mathrm{INT}} + \lambda_i^{\mathrm{EXT}}\right)\right).$$

check-node indices connected to a variable node x_i is denoted by $\mathcal{N}(i)$, while the set of variable-node indices connected to a check node c_j is denoted by $\mathcal{M}(j)$. The algorithm is described in Table 6.2.

Compared with the algorithm in Table 6.1, note that the initialization of the LLRs is now based on the joint distribution $p_{\mathsf{XY}}(x, y)$ and that the jth syndrome value s_j affects the sign of the messages $u_{ji}^{(l)}$.

Example 6.7. To illustrate the performance of the algorithm, we consider a uniform binary memoryless source X, and Y is obtained by sending X through a binary symmetric channel with cross-over probability p. In principle, one can reconstruct the source X^n, provided that it is compressed at a rate of at least $\mathbb{H}_{\mathrm{b}}(p)$. For a code of rate $\frac{1}{2}$, the efficiency is $\beta = 1/(2(1 - \mathbb{H}_{\mathrm{b}}(p)))$. Figure 6.7 shows the bit-error-rate versus efficiency performance of syndrome coding using LDPC codes of length 10^4 with the degree distributions given in Example 6.1 and Example 6.2. These non-optimized degree distributions already provide over 80% efficiency at an error rate of 10^{-5}. Longer codes with optimized degree distributions can easily achieve over 90% efficiency.

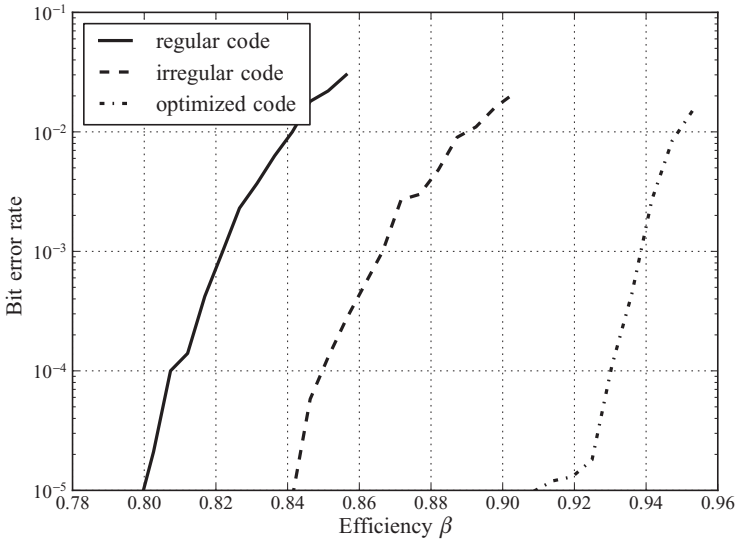

Figure 6.7 Error rate versus efficiency for reconciliation of binary random variables based on LDPC codes. The regular and irregular codes are rate-$\frac{1}{2}$ codes with degree distributions given in Example 6.1 and Example 6.2 and length 10^4. The optimized code is the rate-$\frac{1}{2}$ code of [58] with length 5×10^5.

Remark 6.3. *To ensure the generation of identical keys with privacy amplification, note that the sequence* $\mathbf{x}$ *should be reconstructed* exactly*, and even a bit error rate as low as* 10^{-5} *is not acceptable. The presence of errors is usually well detected by the message-passing decoder, and the key-distillation process could simply be aborted in such cases. However, discarding an entire sequence that contains a small fraction of errors incurs a significant efficiency loss. A more efficient technique consists of concatenating a high-rate outer code, such as a BCH code, to correct the few remaining errors.*

6.5 Reconciliation of general memoryless sources

We now turn our attention to a general memoryless source $(\mathcal{XY}, p_{XY})$, which might be discrete or continuous. If X is a general discrete random variable, Proposition 4.5 applies once again, and reconciliation can be treated as a Slepian–Wolf coding problem. If X is a continuous random variable, its lossless reconstruction would require infinitely many bits, and the traditional approach is to consider the approximate reconstruction of X under a distortion constraint. However, the objective of reconciliation is slightly different because we want to extract a common (binary) sequence from observations of the components X and Y so that privacy amplification can be used later on. Therefore, a pragmatic approach consists of quantizing X into a discrete random variable X' to revert back to the discrete case. In principle, the source X' can be compressed at a rate

R arbitrarily close to $\mathbb{H}(X'|Y)$ and the resulting reconciliation efficiency is

$$\beta = \frac{\mathbb{H}(X') - \mathbb{H}(X'|Y)}{\mathbb{I}(X;Y)} = \frac{\mathbb{I}(X';Y)}{\mathbb{I}(X;Y)}. \tag{6.2}$$

In general, $\beta < 1$, but the penalty inflicted by quantization can be made as small as desired by choosing a fine enough quantizer.

Remark 6.4. *A scalar quantizer is sufficient to obtain near-optimal performance as long as there is no rate constraint on public communication. However, for the same amount of information exchanged over the public channel, a vector quantizer will have a better performance than a scalar quantizer.*

6.5.1 Multilevel reconciliation

In this section, we describe a generic protocol for the reconciliation of a memoryless source $(\mathcal{X}\mathcal{Y}, p_{XY})$ for which $|\mathcal{X}| < \infty$. As discussed above, this protocol is also useful for continuous random variables, and we study the case of Gaussian random variables in Section 6.5.2.

In principle, we could design a Slepian–Wolf code that operates on symbols in the alphabet $\mathcal{X}$ directly, but it is more convenient to use binary symbols only. Letting $\ell = \lceil \log|\mathcal{X}| \rceil$, every symbol $x \in \mathcal{X}$ can be assigned a unique ℓ-bit binary label denoted by the vector $(g_1(x) \dots g_\ell(x))^\intercal$, where

$$\forall i \in [\![1, \ell]\!] \qquad g_i : \mathcal{X} \to \mathrm{GF}(2).$$

The source X is then equivalent to a binary vector source $\mathrm{B}^\ell \triangleq (g_1(\mathrm{X}) \dots g_\ell(\mathrm{X}))^\intercal$. For $i \in [\![1, \ell]\!]$, we call the component source $\mathrm{B}_i \triangleq g_i(\mathrm{X})$ the ith *binary level*, since it corresponds to the ith "level" in the binary representation of X. If we were to try to encode and decode the ℓ binary levels independently with ideal Slepian–Wolf codes, we would use a compression rate R_i arbitrarily close to $\mathbb{H}(\mathrm{B}_i|\mathrm{Y})$ for each level $i \in [\![1, \ell]\!]$, and the overall compression rate would be

$$\sum_{i=1}^{\ell} \mathbb{H}(\mathrm{B}_i|\mathrm{Y}) \geqslant \sum_{i=1}^{\ell} \mathbb{H}\big(\mathrm{B}_i|\mathrm{B}^{i-1}\mathrm{Y}\big) = \mathbb{H}\big(\mathrm{B}^\ell|\mathrm{Y}\big) = \mathbb{H}(\mathrm{X}|\mathrm{Y}),$$

with equality if and only if B_i is independent of B^{i-1} given Y for all $i \in [\![1, \ell]\!]$. Therefore, in general, encoding and decoding the levels separately is suboptimal.

It is possible to achieve better compression by considering the separate encoding/joint-decoding procedure illustrated in Figure 6.8. We start by considering the binary source B_1 and let $\mathcal{E}_1$ be the event representing the occurrence of a decoding error. For any $\gamma, \epsilon > 0$, the Slepian–Wolf theorem guarantees the existence of a code compressing B_1 at rate $R_1 = \mathbb{H}(\mathrm{B}_1|\mathrm{Y}) + \gamma$ with probability of error $\mathbb{P}[\mathcal{E}_1] \leqslant \epsilon$. Assuming that level B_1 is successfully decoded, we can now treat it as additional side information available to the receiver. Hence, we encode B_2 using a code with compression rate $R_2 = \mathbb{H}(\mathrm{B}_2|\mathrm{B}_1\mathrm{Y}) + \gamma$ that guarantees a probability of error of $\mathbb{P}\left[\mathcal{E}_2|\mathcal{E}_1^{\mathrm{c}}\right] \leqslant \epsilon$. By repeating this procedure for

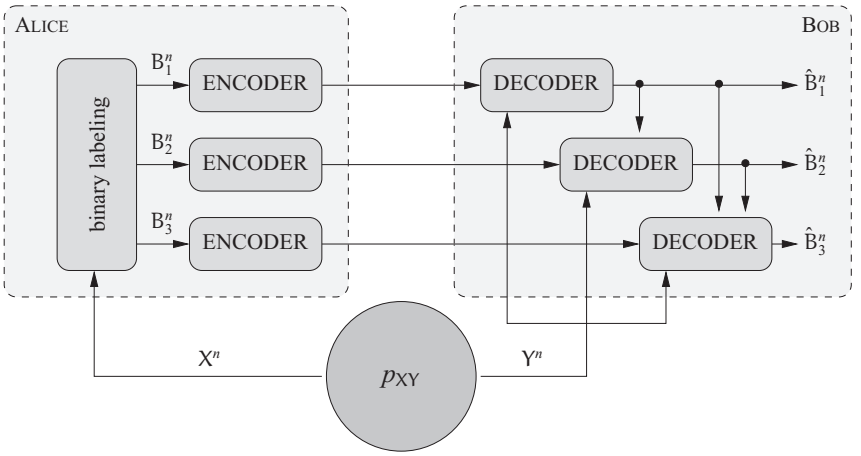

Figure 6.8 Example of a multilevel reconciliation protocol for $\ell = 3$. Each level B_i is decoded using Y^n and previously decoded levels as side information.

each level $i \in [\![1, \ell]\!]$, B_i is encoded *independently* at a rate $R_i = \mathbb{H}\big(\mathrm{B}_i|\mathrm{B}^{i-1}\mathrm{Y}\big) + \gamma$ with a code ensuring a probability of decoding error of $\mathbb{P}\big[\mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j^{\mathrm{c}}\big] \leqslant \epsilon$. However, the levels must be decoded *successively*, using the previously decoded levels as side information. The overall probability of error is

$$\begin{aligned}
\mathbb{P}\left[\bigcup_{i=1}^{\ell} \mathcal{E}_i\right] &= \mathbb{P}\left[\bigcup_{i=1}^{\ell} \left(\mathcal{E}_i \cup \bigcap_{j=1}^{i-1} \mathcal{E}_i^{\mathrm{c}}\right)\right] \\
&\leqslant \sum_{i=1}^{\ell} \mathbb{P}\left[\mathcal{E}_i \cup \bigcap_{j=1}^{i-1} \mathcal{E}_i^{\mathrm{c}}\right] \\
&\leqslant \sum_{i=1}^{\ell} \mathbb{P}\left[\mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_i^{\mathrm{c}}\right] \\
&\leqslant \ell\epsilon,
\end{aligned}$$

which can be made as small as desired with a large enough blocklength n since the number of levels ℓ is fixed. In addition, the overall compression rate is then

$$R_{\mathrm{tot}} = \sum_{i=1}^{\ell} \mathbb{H}\big(\mathrm{B}_i|\mathrm{B}^{i-1}\mathrm{Y}\big) + \ell\gamma = \mathbb{H}\big(\mathrm{B}^{\ell}|\mathrm{Y}\big) + \ell\gamma = \mathbb{H}(\mathrm{X}|\mathrm{Y}) + \ell\gamma,$$

which can be made as close as desired to the optimal compression rate $\mathbb{H}(\mathrm{X}|\mathrm{Y})$. This optimal encoding/decoding scheme is called *multilevel reconciliation* because of its similarity to multilevel channel coding with multistage decoding.

Remark 6.5. *The optimality of multilevel reconciliation does not depend on the labeling* $\{g_i\}_\ell$ *used to transform* X *into the vector of binary sources* $(\mathrm{B}_1 \ldots \mathrm{B}_\ell)^\intercal$. *However, different*

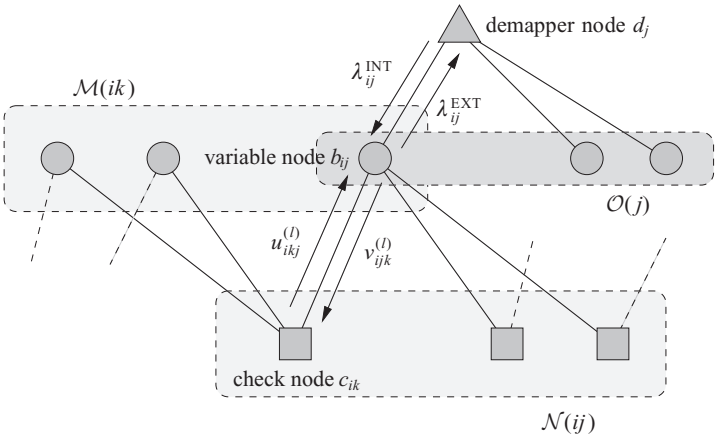

Figure 6.9 Illustration of a message-passing algorithm for multilevel reconciliation. The set of check nodes connected to a variable note b_{ij} is denoted $\mathcal{N}(ij)$. The set of variable nodes connected to a check node c_{ik} is denoted $\mathcal{M}(ik)$. The set of variable nodes connected to a demapper node d_j is denoted $\mathcal{O}(j)$.

choices of labeling result in different values for the compression rates $R_i = \mathbb{H}\big(\mathrm{B}_i|\mathrm{B}^{i-1}\mathrm{Y}\big)$ *for* $i \in [\![1, \ell]\!]$*, which can in turn have an effect on the complexity of the code design. For instance, it can be quite difficult to design high-rate or low-rate codes with near-optimal performance. We provide heuristic guidelines for the choice of labeling in Section 6.5.2.*

As in Section 6.4, we can implement multilevel reconciliation efficiently using syndrome coding with LDPC codes for each of the ℓ levels. Specifically, given a vector of realizations $\mathbf{x} = (x_1 \dots x_n)$ of the source X, the labeling creates ℓ binary vectors $\mathbf{b}_i = g_i(\mathbf{x})$ for $i \in [\![1, \ell]\!]$, which can be represented in matrix form as

$$\begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_i \\ \vdots \\ \mathbf{b}_\ell \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ b_{i1} & b_{i2} & \dots & b_{in} \\ \vdots & \vdots & \vdots & \vdots \\ b_{\ell 1} & b_{\ell 2} & \dots & b_{\ell n} \end{pmatrix} \begin{matrix} \text{level } 1 \\ \vdots \\ \text{level } i \\ \vdots \\ \text{level } \ell \end{matrix}$$

For each level $i \in [\![1, \ell]\!]$, the encoder computes the syndromes $\mathbf{s}_i = \mathbf{H}_i \mathbf{b}_i$ with the parity-check matrix $\mathbf{H}_i \in \mathrm{GF}(2)^{k_i \times n}$ of an $(n, n - k_i)$ linear code. Note that, for a fixed $j \in [\![1, n]\!]$, the bits $\{\mathrm{B}_{1j}, \mathrm{B}_{2j}, \dots \mathrm{B}_{\ell j}\}$ are correlated because they stem from the same symbol X_j. Consequently, the intrinsic LLR of every bit b_{ij} with $i \in [\![1, \ell]\!]$ depends not only on the side information y_j but also on the estimation of the bits $\{b_{mj}\}_\ell \setminus \{b_{ij}\}$. Therefore, the message-passing algorithm described in Section 6.4 must be modified to take into account the result of previously decoded levels.

To describe the full message-passing algorithm, we introduce the following notation, which is also illustrated in Figure 6.9 for convenience. For each level $i \in [\![1, \ell]\!]$, the encoder computes the vector $\mathbf{b}_i = g_i(\mathbf{x}) = (b_{i1} \dots b_{in})^\intercal$, where each b_{ij} corresponds

to the ith bit in the binary description of symbol x_j. The receiver, who obtains $\mathbf{y} = (y_1 \dots y_n)^\intercal$ as side observation, receives the syndrome vector $\mathbf{s}_i = \mathbf{H}_i \mathbf{b}_i$ of k_i bits, whose entries correspond to the values of check node $\{c_{i1}, \dots, c_{ik_i}\}$ in the Tanner graph of the LDPC code for level i. The set of check-node indices connected to a variable node b_{ij} is denoted by $\mathcal{N}(ij)$. The set of variable-node indices connected to a check node c_{ik} is denoted by $\mathcal{M}(ik)$. Finally, the update of the intrinsic LLRs is represented by *n demapper nodes* $\{d_j\}_n$ connecting the Tanner graphs of individual levels. The set of variable nodes $\{b_{1j}, \dots, b_{\ell j}\}$ stemming from the same symbol x_j is denoted by $\mathcal{O}(j)$. One can show that the intrinsic LLR $\lambda_{ij}^{\mathrm{INT}}$ of bit $b_{i,j}$ should be calculated from symbol y_j and the extrinsic LLRs $\lambda_{mj}^{\mathrm{EXT}}$ of bits $\{b_{mj}\}$ for $m \in \mathcal{O}(j) \setminus i$ as

$$\lambda_{ij}^{\mathrm{INT}} = \log \left(\frac{\displaystyle\sum_{x \in \mathcal{X}: g_i(x)=0} p_{XY}(x, y_j) \exp\left(\sum_{m \in \mathcal{O}(j) \setminus i} (1 - g_m(x)) \lambda_{mj}^{\mathrm{EXT}} \right)}{\displaystyle\sum_{x \in \mathcal{X}: g_i(x)=1} p_{XY}(x, y_j) \exp\left(\sum_{m \in \mathcal{O}(j) \setminus i} (1 - g_m(x)) \lambda_{mj}^{\mathrm{EXT}} \right)} \right). \tag{6.3}$$

Note that the extrinsic LLRs $\lambda_{mj}^{\mathrm{EXT}}$ appear as weighting factors in front of the joint probability terms $p_{XY}(x, y_i)$. As the magnitude of $\lambda_{mj}^{\mathrm{EXT}}$ increases (that is, the estimation of b_{mj} becomes more reliable), some symbols x are given more importance than others in the calculation of $\lambda_{ij}^{\mathrm{INT}}$. For clarity, we denote the operation defined by (6.3) as

$$\lambda_{ij}^{\mathrm{INT}} = \boxplus_{m \in \mathcal{O}(j) \setminus i} \lambda_{mj}^{\mathrm{EXT}}.$$

The entire decoding algorithm is described in Table 6.3.

If the overall graph contains no cycles, the algorithm can be shown to compute exactly the a-posteriori LLRs

$$\log \left(\frac{\mathbb{P}\left[\mathrm{B}_{ij} = 0 | \mathbf{y}, \mathbf{s}\right]}{\mathbb{P}\left[\mathrm{B}_{ij} = 1 | \mathbf{y}, \mathbf{s}\right]} \right).$$

In practice, finite-length LDPC codes contain cycles, but the algorithm still provides reasonable approximations of the real a-posteriori LLRs.

Remark 6.6. *Since the update of λ_{ij}^{INT} is based on extrinsic LLRs, it is possible to modify the scheduling of the previous algorithm and to start decoding level i even if the previous levels $[\![1, i-1]\!]$ have not been entirely decoded. In that case, level i cannot be decoded entirely, but the extrinsic LLRs might be fed back to the message-passing algorithm of the previous levels. Although this feedback is not necessary in principle, it does improve the performance of practical implementations.*

Remark 6.7. *The optimality of multilevel reconciliation requires different compression rates, and therefore different codes, for each level. To simplify the code design, it is possible to use a single code across all levels, albeit with a reduced efficiency. Whereas multilevel reconciliation is similar to multilevel coding, this simplified approach is*

Table 6.3 Belief-propagation algorithm for multilevel reconciliation

☐ **Initialization.**
► For each $i \in [\![1, \ell]\!]$, for each $j \in [\![1, n]\!]$ and for each $k \in [\![1, k_i]\!]$

$$\lambda_{ij}^{\mathrm{INT}} = \lambda_{ij}^{\mathrm{EXT}} = v_{ijk}^{(0)} = u_{ikj}^{(0)} = 0.$$

☐ **Iterations across levels.** For each level $i \in [\![1, \ell]\!]$
► For each $j \in [\![1, n]\!]$

$$\lambda_{ij}^{\mathrm{INT}} = \boxplus_{m \in \mathcal{O}(j) \setminus i} \lambda_{mj}^{\mathrm{EXT}}.$$

► Iteration within level. For each $l \in [\![1, l_{\max}]\!]$
▷ For each $j \in [\![1, n]\!]$ and for each $k \in \mathcal{N}(ij)$

$$v_{ijk}^{(l)} = \lambda_{ij}^{\mathrm{INT}} + \sum_{m \in \mathcal{N}(ij) \setminus k} u_{imj}^{(l-1)}.$$

▷ For each $k \in [\![1, k_i]\!]$ and for each $j \in \mathcal{M}(ik)$

$$u_{ikj}^{(l)} = 2 \tanh^{-1} \left((1 - 2s_{ik}) \prod_{m \in \mathcal{M}(ik) \setminus j} \tanh \left(\frac{v_{imk}^{(l)}}{2} \right) \right).$$

► **Extrinsic information.** For each $j \in [\![1, n]\!]$

$$\lambda_{ij}^{\mathrm{EXT}} = \sum_{m \in \mathcal{N}(ij)} u_{imj}^{(l_{\max})}.$$

☐ **Hard decisions.** For each $i \in [\![1, \ell]\!]$ and for each $j \in [\![1, n]\!]$

$$\hat{b}_{ij} = \frac{1}{2} \left(1 - \mathrm{sign} \left(\lambda_{ij}^{\mathrm{INT}} + \lambda_{ij}^{\mathrm{EXT}} \right) \right).$$

similar to bit-interleaved coded modulation. The decoding algorithm described in Table 6.3 is easily adapted to syndrome coding with a single LDPC code.

6.5.2 Multilevel reconciliation of Gaussian sources

In this section, we detail the construction of a multilevel reconciliation scheme for a memoryless Gaussian source. We assume

$$\mathsf{X} \sim \mathcal{N}(0, 1) \quad \text{and} \quad \mathsf{Y} = \mathsf{X} + \mathsf{N} \ \text{ with } \ \mathsf{N} \sim \mathcal{N}(0, \sigma^2),$$

and we define the signal-to-noise ratio of this source as $\mathrm{SNR} \triangleq 1/\sigma^2$. Notice that this type of source could be simulated by transmitting Gaussian noise over a Gaussian channel. Before we design a set of LDPC codes for multilevel reconciliation, we first need to construct a scalar quantizer, define a labeling, and compute the compression rate required at each level.

By construction, the joint distribution of X and Y satisfies the property

$$\forall (x, y) \in \mathbb{R}^2 \qquad p_{\mathsf{Y}\mathsf{X}}(y, x) = p_{\mathsf{Y}\mathsf{X}}(-y, -x),$$

which it is desirable to preserve when designing the scalar quantizer. Specifically, we restrict ourselves to quantizers $Q : \mathbb{R} \to \mathbb{R}$ such that

$$\forall (x, y)^2 \in \mathbb{R} \qquad p_{YX}(y, Q(x)) = p_{YX}(-y, -Q(x)),$$

and we let $X' = Q(X)$. There is some leeway for choosing the number of quantization intervals, but one should choose them such that $\mathbb{I}(X'; Y)$ is close to $\mathbb{I}(X; Y)$ in order to minimize the efficiency penalty in (6.2). Naturally, the higher the SNR is, the more quantization intervals are needed.

Example 6.8. For simplicity, we consider quantizers with equal-width quantization intervals (except for the two intervals on the boundaries). If SNR = 3, the quantizer with 16 intervals maximizing the mutual information $\mathbb{I}(X'; Y)$ is as given below.

Quantization intervals	
$(-\infty, -2.348]$	$(+0.000, +0.254]$
$(-2.348, -1.808]$	$(+0.254, +0.514]$
$(-1.808 - 1.412]$	$(+0.514, +0.787]$
$(-1.412, -1.081]$	$(+0.787, +1.081]$
$(-1.081, -0.787]$	$(+1.081, +1.412]$
$(-0.787, -0.514]$	$(+1.412, +1.808]$
$(-0.514, -0.254]$	$(+1.808, +2.348]$
$(-0.254 + 0.000]$	$(+2.348, +\infty)$

The quantizer yields $\mathbb{I}(X'; Y) \approx 0.98$ bits, compared with $\mathbb{I}(X; Y) = 1$ bit. The entropy of the quantized source is $\mathbb{H}(X') \approx 3.78$ bits.

We assume that the intervals can be labeled with ℓ bits. Once a labeling $\{g_i\}_\ell$ has been defined, we obtain a binary vector source $\mathrm{B}^\ell \triangleq (\mathrm{B}_1, \ldots, \mathrm{B}_\ell)$ with $\mathrm{B}_i \triangleq g_i(X')$ for $i \in [\![1, \ell]\!]$, which is equivalent to the quantized source X'. The optimal compression rates that would be required for ideal codes are then $R_i = \mathbb{H}\big(\mathrm{B}_i | Y \mathrm{B}^{i-1}\big)$. These rates are easily calculated by writing R_i as

$$R_i = \mathbb{H}\big(\mathrm{B}_i | Y \mathrm{B}^{i-1}\big) = \mathbb{H}\big(\mathrm{B}^i | Y\big) - \mathbb{H}\big(\mathrm{B}^{i-1} | Y\big).$$

The entropy $\mathbb{H}\big(\mathrm{B}^i | Y\big)$ is given explicitly by

$$- \sum_{\mathbf{b} \in \mathrm{GF}(2)^i} \int_{-\infty}^{\infty} \left(\int_{\mathcal{A}_i(\mathbf{b})} p_{X|Y}(x|y) \mathrm{d}x \right) \log \left(\int_{\mathcal{A}_i(\mathbf{b})} p_{XY}(x, y) \mathrm{d}x \right) \mathrm{d}y,$$

with

$$\mathcal{A}_i(\mathbf{b}) \triangleq \{x : (g_1(Q(x)), \ldots, g_i(Q(x))) = \mathbf{b}\}.$$

Note that the optimal compression rates needed at each level depend on the specific choice of labeling considered.

Example 6.9. We consider the quantizer obtained in Example 6.8. The simplest labeling is the *natural labeling* given below, in which the first level consists of the least significant bits.

Natural labeling																
Level 4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Level 3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
Level 2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
Level 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

The optimal rates for SNR $= 3$ are then the following.

Level	Compression rate	Code rate for syndrome coding
4	0.079	0.921
3	0.741	0.259
2	0.984	0.016
1	0.987	0.013

Another simple labeling is the *reverse natural labeling*, which is similar to natural labeling, but for which the first level consists of the most significant bits.

Reverse natural labeling																
Level 4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Level 3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
Level 2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
Level 1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

The optimal rates for SNR $= 3$ are now the following.

Level	Compression rate	Code rate for syndrome coding
4	0.936	0.064
3	0.801	0.199
2	0.547	0.453
1	0.507	0.493

If we could construct ideal codes at all rates, then no labeling would be better than any other. However, in practice, it is difficult to design efficient codes with compression rates close to unity. Rather than designing efficient codes, it is actually easier not to compress at all and to simply disclose the bits of the entire level. This simplification induces a small

penalty, but, for compression rates close to unity, this is more efficient than trying to design a powerful code. For instance, for the natural labeling in Example 6.9, disclosing levels 1 and 2 incurs a negligible efficiency loss. In addition, finite-length LDPC codes do not have the performance of ideal codes; therefore, the code rates at each level must be chosen to be below the ideal ones, which inflicts an efficiency penalty. Nevertheless, one can still achieve high efficiencies, as illustrated by the following example.

Example 6.10. We consider the quantizer of Example 6.8 used in conjunction with natural labeling. Instead of the optimal rates computed in Example 6.9 for SNR $= 3$, we use the following:

Level	Compression rate	Code rate
4	0.14	0.86
3	0.76	0.24
2	1.0	0.0
1	1.0	0.0

Levels 1 and 2 are entirely disclosed so that there are only two codes to design, with rates 0.24 and 0.86, respectively. We choose the rate-0.24 LDPC codes with degree distributions

$$\begin{aligned}\lambda_1(x) = {} & 0.249\,17x + 0.163\,92x^2 + 0.000\,01x^3 + 0.159\,92x^5 + 0.023\,23x^6 \\ & + 0.080\,87x^{12} + 0.019\,58x^{13} + 0.036\,39x^{20} + 0.018\,61x^{26} \\ & + 0.029\,56x^{27} + 0.010\,06x^{30} + 0.048\,15x^{32} + 0.160\,51x^{99},\end{aligned}$$

$$\rho_1(x) = 0.1x^4 + 0.9x^5,$$

and the rate-0.86 LDPC code with degree distributions

$$\lambda_2(x) = 0.227\,34x + 0.131\,33x^2 + 0.641\,33x^4,$$

$$\rho_2(x) = x^{24}.$$

Figure 6.10 shows the error-rate versus efficiency performance obtained with the multi-level reconciliation algorithm of Section 6.5.1 and various choices of blocklength. Note that long codes can achieve a probability of error of 10^{-5} at an efficiency close to 90%.

6.6 Secure communication over wiretap channels

Although wiretap codes are not known for general channels, it is nevertheless possible to construct practical codes that are based on sequential key-distillation strategies with one-way reconciliation. In fact, for a WTC $(\mathcal{X}, p_{YZ|X}, \mathcal{Y}, \mathcal{Z})$, one can implement the following four-stage protocol.

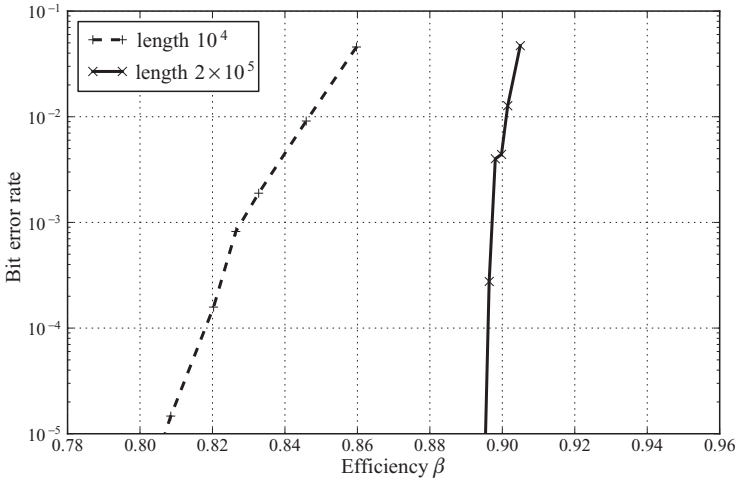

Figure 6.10 Error rate versus efficiency of a finite-length LDPC-based multilevel reconciliation scheme.

1. **Randomness sharing.** Alice generates n realizations of a random variable X with distribution p_X and transmits them through the WTC. Bob and Eve observe n realizations of correlated random variables Y and Z, respectively. The resulting joint distribution factorizes as $p_{XYZ} = p_{YZ|X}p_X$.
2. **Information reconciliation.** Alice computes syndromes using the multilevel reconciliation protocol described in Section 6.5.1, and transmits them over the main channel $(\mathcal{X}, p_{Y|X}, \mathcal{Y})$ using a channel error-correcting code. In principle, the rate of the channel code can be chosen arbitrarily close to the capacity C_m.
3. **Privacy amplification.** Alice chooses a hash function at random in a universal family and transmits this choice over the main channel using a channel error-correcting code. Alice and Bob distill a secret key K.
4. **Secure communication.** Alice uses the key K to encrypt a message M with a one-time pad and transmits the encrypted message over the main channel using a channel error-correcting code.

In general, this procedure is fairly inefficient because many channel uses are wasted to generate a source and distill a secret key instead of for communicating secure messages. The secure rate R_s achieved by this procedure can be estimated as follows. For a given input distribution p_X, about $n\mathbb{H}(X|Y)$ bits are required for information reconciliation, which can be transmitted with $n\mathbb{H}(X|Y)/C_m$ additional channel uses. The choice of a hash function in the family of hash functions in Example 4.6 requires about $n\mathbb{H}(X)$ bits for privacy amplification, which can be transmitted with another $n\mathbb{H}(X)/C_m$ channel uses. Finally, the secret key K distilled by Alice and Bob contains on the order of $n(\mathbb{I}(X;Y) - \mathbb{I}(X;Z))$ bits, which allows the secure transmission of the same number of message bits and requires $n(\mathbb{I}(X;Y) - \mathbb{I}(X;Z))/C_m$ channel uses. Hence, the secure

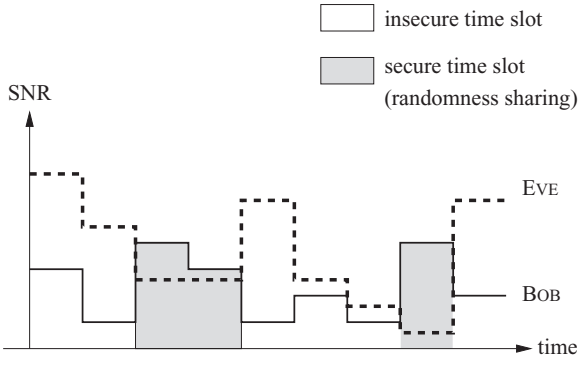

Figure 6.11 Opportunistic use of fading for secure communication.

communication rate is approximately

$$R_{\mathrm{s}} \approx \frac{C_{\mathrm{m}}(\mathbb{I}(\mathsf{X};\mathsf{Y}) - \mathbb{I}(\mathsf{X};\mathsf{Z}))}{C_{\mathrm{m}} + \mathbb{H}(\mathsf{X}|\mathsf{Y}) + \mathbb{H}(\mathsf{X}) + \mathbb{I}(\mathsf{X};\mathsf{Y}) - \mathbb{I}(\mathsf{X};\mathsf{Z})}.$$

Example 6.11. Let us consider a binary WTC in which the main channel and the eavesdropper's channel are both binary symmetric channels with cross-over probabilities p_{m} and p_{e}, respectively ($p_{\mathrm{e}} > p_{\mathrm{m}}$). Assume that the procedure described above is used with $\mathsf{X} \sim \mathcal{B}(\frac{1}{2})$. Then the secure communication rate is at most

$$R_{\mathrm{s}} = \frac{1 - \mathbb{H}_{\mathrm{b}}(p_{\mathrm{m}})}{2 + \mathbb{H}_{\mathrm{b}}(p_{\mathrm{e}}) - \mathbb{H}_{\mathrm{b}}(p_{\mathrm{m}})} C_{\mathrm{s}}.$$

In the extreme case $p_{\mathrm{m}} = 1$ and $p_{\mathrm{e}} = \frac{1}{2} + \epsilon$ for some small $\epsilon > 0$, note that C_{s} is on the order of 1 bit per channel use but the rate achieved R_{s} is only on the order of $\frac{1}{3}$ bits per channel use.

Example 6.12. Let us consider the quasi-static fading WTCs discussed in Section 5.2.3 with full channel state information. As illustrated in Figure 6.11, the four-stage protocol can be implemented opportunistically in such a way that randomness sharing is performed only during *secure time slots* for which the legitimate receiver has a better instantaneous SNR than that of the eavesdropper. Reconciliation, privacy amplification, and secure communication are performed during the remaining time slots. This does not affect the secure rate because the key distilled during the protocol is secure against an eavesdropper who obtains the reconciliation and privacy-amplification messages perfectly.

To avoid sharing randomness at a faster rate than that at which it can be processed, it could be necessary to use a fraction of the secure time slots for reconciliation, privacy amplification, or secure communication; however, if the eavesdropper has a much higher SNR on average, all of the communication required for secret-key distillation is

performed during insecure time slots. In such a case, the protocol achieves secure rates close to the secrecy capacity of the channel.

Remark 6.8. *Although we have assumed that privacy amplification is performed using the hash functions of Example 4.6, there is not much to be gained by choosing another universal family of hash functions. It can be shown that the minimum number of hash functions in a universal family* $\mathcal{H} = \{h : \mathrm{GF}(2)^n \to \mathrm{GF}(2)^k\}$ *is* 2^{n-k} *[67]; in practice, no families with fewer than* 2^n *functions are known. We could in principle do better by performing privacy amplification with extractors.*

6.7 Bibliographical notes

LDPC codes were invented by Gallager in 1963 [110], and experienced a renewed interest with the works of MacKay and others. Density evolution and the threshold property of LDPC codes under message-passing decoding were investigated by Richardson and Urbanke [111]. A Gaussian-approximation version of density evolution leading to a linear optimization problem was analyzed by Chung, Forney, Richardson, and Urbanke [112]. This enabled the design of irregular LDPC codes performing extremely close to the Shannon limit [113, 114, 115]. Although it is not known whether there exist sequences of capacity-achieving LDPC codes over arbitrary channels, it is possible to construct sequences of capacity-achieving LDPC codes for the erasure channel. The threshold for the block error probability of LDPC ensembles under belief-propagation decoding was analyzed by Jin and Richardson [109].

The first wiretap-code constructions were proposed for the wiretap channel of type II by Ozarow and Wyner [116]. Thangaraj, Dihidar, Calderbank, McLaughlin, and Merolla generalized these ideas to other wiretap channels [117] and proposed an explicit coset coding scheme based on LDPC codes for the erasure wiretap channel. Suresh, Subramanian, Thangaraj, Bloch, and McLaughlin later proved that the same construction can be used to guarantee strong secrecy, albeit at lower rates [118, 119]. A similar construction based on two-edge-type LDPC codes was proposed by Rathi, Andersson, Thobaben, Kliewer, and Skoglund [120]. For discrete additive noise channels, Cohen and Zemor showed that any random linear code used in conjunction with coset coding is likely to satisfy a strong secrecy requirement [121]. All of the aforementioned constructions are connected to nested codes, as highlighted by Liu, Liang, Poor, and Spasojević [122]. In a slightly different spirit, Verriest and Hellman analyzed the use of convolutional encoding over wiretap channels with noiseless main channel and binary symmetric eavesdropper's channel [123]. For the wiretap channel of type II, Wei established a direct relation between the equivocation guaranteed by a linear code and its generalized Hamming weights [124]. Recently, there has also been a lot of interest in the design of wiretap codes based on polar codes [125, 126, 127, 128]. The polarization and capacity-achieving properties of polar codes over binary-input symmetric-output

channels allow one to perform a security analysis that closely follows the equivocation calculation used in Chapter 3.

The challenges posed by the design of wiretap codes ensuring information-theoretic security have fostered the development of alternative secrecy metrics. For instance, Klinc, Ha, McLaughlin, Barros, and Kwak proposed the use of punctured LDPC codes to drive the eavesdropper's bit error probability to an arbitrarily high level over the Gaussian wiretap channel [129]. In the same spirit, Belfiore, Oggier, and Solé proposed the use of lattice codes over the Gaussian wiretap channel [130, 131] and, since the error probability of lattices over Gaussian channels can be related to their theta series, they proposed a secrecy criterion based on the theta series of lattices.

The idea of performing source coding with side information by using the syndromes of good channel codes was proposed by Wyner [132]. Liveris, Xiong, and Georghiades applied this idea to binary memoryless sources with LDPC codes and demonstrated that performance close to the optimal limit could be attained [133]. A threshold analysis of the message-passing algorithm has been proposed by Chen, He, and Jamohan [134].

The reconciliation of binary random variables was first considered by Brassard and Salvail in the context of quantum key distribution with the suboptimal yet computationally efficient algorithm CASCADE [56]. An extensive performance analysis of LDPC-based reconciliation for binary random variables was recently reported by Elkouss, Leverrier, Alléaume, and Boutros [58]. Multilevel reconciliation was first considered by Van Assche, Cardinal, and Cerf [59] for the reconciliation of continuous random variables, and implemented with turbo-codes by Nguyen, Van Assche, and Cerf [60]. Bloch, Thangaraj, McLaughlin, and Merolla reformulated the reconciliation of continuous random variables as a coded modulation problem and used LDPC codes to achieve near-optimal performance [61]. A special case of the general algorithm was developed independently by Ye, Reznik, and Shah [62, 78]. A multidimensional reconciliation scheme for continuous random variables based on the algebraic properties of octonions was also proposed by Leverrier, Alléaume, Boutros, Zémor, and Grangier [135], and was proved to be particularly efficient in the low-SNR regime. Conceptually, multilevel reconciliation is the source-coding counterpart of multilevel coding, a good survey of which can be found in [136] (see also [137] for bit-interleaved coded modulation). LDPC-based multilevel coding has been analyzed quite extensively, see for instance [138].