Multiedge-type low-density parity-check codes for continuous-variable quantum key distribution

Hossein Mani,[1] Tobias Gehring,[1,*] Philipp Grabenweger,[2] Bernhard Ömer,[2] Christoph Pacher,[2] and Ulrik Lund Andersen[1]

[1]*Center for Macroscopic Quantum States (bigQ), Department of Physics, Technical University of Denmark, Fysikvej, 2800 Kongens Lyngby, Denmark*

[2]*Center for Digital Safety & Security, AIT Austrian Institute of Technology GmbH, Giefinggasse 4, 1210 Vienna, Austria*

(Received 22 July 2020; accepted 15 June 2021; published 25 June 2021)

Continuous-variable quantum key distribution (QKD) utilizes an ensemble of coherent states of light to distribute secret encryption keys between two parties. An essential ingredient of the QKD protocol is highly efficient information reconciliation. To achieve highly efficient reconciliation, error-correcting codes with a low channel coding rate are inevitable in the most common schemes of multilevel coding and multistage decoding (MLC-MSD) and multidimensional reconciliation. Multiedge-type (MET) low-density parity-check (LDPC) codes are well suited for highly efficient reconciliation at low rates. Here, we calculate the optimal channel coding rates in the MLC-MSD scheme for reverse reconciliation, introduce the concept of generalized extrinsic information transfer charts for MET-LDPC codes, which constitute a simple and fast asymptotic analysis tool, and present a set of MET-LDPC codes with asymptotic efficiency >97% for channel coding rates 0.1, 0.05, 0.02, and 0.01. We believe that our codes will find wide application in implementations of continuous-variable quantum key distribution based on Gaussian modulation.

DOI: 10.1103/PhysRevA.103.062419

I. INTRODUCTION

The security of today's asymmetric cryptography, e.g. the Rivest-Shamir-Adleman (RSA) protocol and the Diffie-Hellman key-exchange protocol, is based on mathematical complexity assumptions of basic problems such as the discrete log problem and the factorization of large numbers [1]. The advent of the quantum computer or even an unexpected algorithmic innovation can compromise their security with drastic consequences for the internet.

One possible solution is quantum key distribution (QKD) [2], which provides information theoretical secure cryptographic key exchange for two parties, Alice and Bob, based on the properties of quantum mechanics. In continuous-variable QKD (CV-QKD) [2–4] the transmitter, Alice, modulates (weak) coherent states, and the receiver, Bob, measures the amplitude and phase quadratures of the electromagnetic light field. The communication distance and the key generation rate are thereby limited by the performance of information reconciliation, which is an important part of every QKD protocol to ensure that both parties generate the same cryptographic key. Reverse reconciliation is usually applied; that is, Alice has to reconcile with Bob's measurement results. The main challenge here is the design of capacity-approaching error correction codes within a large range of signal-to-noise ratios (SNRs) required to cover a large range of transmission distances. For instance, in Ref. [5] an SNR of −15.37 dB was reported for a transmission distance of 80 km, and in Ref. [6] an SNR of −16.198 dB was reported for 100 km. At shorter distances the SNR is larger, e.g., on the order of 10 dB at 1–2 km [7].

In general, two methods have mostly been considered previously: Multidimensional (MD) reconciliation [8] and slice reconciliation using multilevel coding and multistage decoding (MLC-MSD) [9,10]. The first is usually employed for low SNRs, i.e., below 0 dB, and is well studied in the literature [5,8]. The second method has in principle the ability to extract more than 1 bit of information per symbol and is thus usually employed for SNRs higher than 0 dB. In Ref. [9] the reconciliation efficiency and optimum channel code rates for direct reconciliation and an SNR $\geqslant -3$ dB have been calculated. Slice reconciliation can also be used with nonbinary low-density parity-check (LDPC) codes [11]; however, nonbinary LDPC is computationally very complex and therefore only a solution to niche applications [7].

In both cases, MD reconciliation and MLC-MSD, LDPC codes with low channel coding rate are required. Multiedge-type LDPC (MET-LDPC) codes are highly suitable for this task; however, only a few codes have been published so far [5,12], and more emphasis has been put on increasing the throughput of the decoder [13–15]. More specifically, in Ref. [5] a MET-LDPC code with channel coding rate 0.02 was introduced for long-distance reconciliation with asymptotic efficiency of 98.1%, and in Ref. [12], two degree distributions for channel coding rates 0.05 and 0.1 were presented with corresponding asymptotic efficiencies of 96.6 and 91.8 %. In addition, Ref. [15] introduced two spatially coupled LDPC codes with channel coding rates 1/3 and 1/4 for high-speed reconciliation of CV-QKD. At a frame error rate of 0.5 the codes have efficiencies of 93.5% and below 91%, respectively.

*tobias.gehring@fysik.dtu.dk

Here, we first extend the analysis of Ref. [9] on MLC-MSD for CV-QKD to reverse reconciliation, show optimal channel coding rates for a wide range of SNRs from −20 to 10 dB, and compare the performance with MD reconciliation. We then introduce the concept of generalized extrinsic information transfer (G-EXIT) charts for MET-LDPC codes, which are a useful visualization tool for code performance evaluation. Finally, we introduce highly efficient MET-LDPC codes with channel coding rates 0.01, 0.02, 0.05, and 0.1, suitable for both MLC-MSD and MD reconciliation, and we investigate their performance. We find asymptotic efficiencies larger than 97% and determine their performance with finite block length. We expect our codes to find wide application in reconciliation of Gaussian-modulated CV-QKD protocols.

EXIT charts have been used in the context of CV-QKD before [10], but only for the design of irregular LDPC codes. Only moderate efficiencies have been achieved due to the lack of codes with low channel coding rate since low-rate codes with high efficiency cannot be achieved with irregular codes. Here, we fill this gap by considering EXIT charts for MET-LDPC codes.

The organization of the remainder of this paper is as follows. First a short background is presented in Sec. II. In Sec. III, we introduce the MLC-MSD scheme and calculate the designed capacity rate for each level for a given input distribution. Finally, we compare the reconciliation efficiency of the MLC-MSD scheme with MD reconciliation for the same code efficiency used for error correction. In Sec. IV we introduce the concept of G-EXIT charts for MET-LDPC codes. MET-LDPC codes for multiple channel coding rates are presented in Sec. V, where we demonstrate their asymptotic and finite-size performance. Finally, Sec. VI concludes the paper.

II. BACKGROUND

Information reconciliation is a method by which two parties, each possessing a sequence of numbers, agree on a *common* sequence of bits by exchanging one or more messages. Mathematically speaking, in CV-QKD with Gaussian modulation the two sequences of numbers are joint instances of a bivariate random variable that follows a bivariate normal distribution. Physically, these sequences are obtained by one party generating coherent states in the quadrature phase space and the other party measuring them. In other words, in QKD, two parties share correlated random variables and wish to agree on a common bit sequence. However, imperfect correlations introduced by the inherent shot noise of coherent states and noise in the quantum channel and the receiver give rise to discrepancies in the two sequences of numbers which have to be corrected by exchanging additional information.

In reverse reconciliation, which is the focus of this work, Alice reconciles her modulated symbols to match Bob's measurement outcomes. The reconciliation process can be fully described as a conventional communication theory problem, which was first addressed in Ref. [16] as source coding with side information: Let Alice and Bob have access to two correlated information sources X_A and X_B which follow a joint probability distribution $p_{X_A X_B}(x_A, x_B)$. The two parties wish to distill a common binary string from blocks of length n,

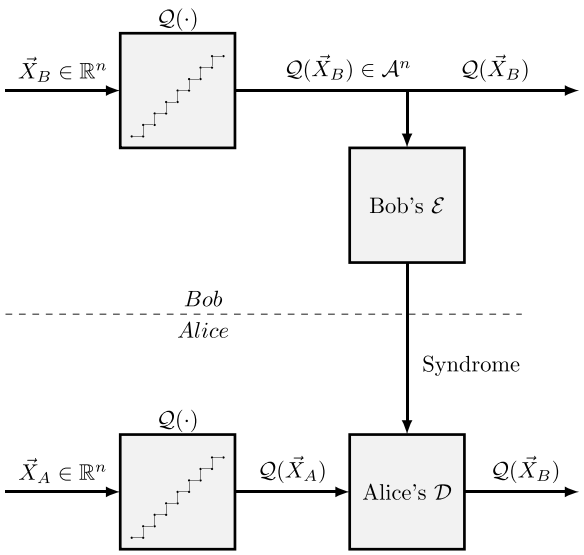

FIG. 1. Correlated source coding configuration. Correlated information sequences $\vec{X}_B = (X_{B,0}, X_{B,1}, \ldots, X_{B,n-1})$ and $\vec{X}_A = (X_{A,0}, X_{A,1}, \ldots, X_{A,n-1})$ are generated by a pair of continuous random variables X_A, X_B from a given bivariate distribution $p_{X_A X_B}(x_A, x_B)$.

$\vec{X}_A = (X_{A,i})_{i=0}^{n-1}$, $\vec{X}_B = (X_{B,i})_{i=0}^{n-1}$, by exchanging information. Since it is convenient to generate the reconciliation messages as syndromes using linear codes [10,17], the parity-check matrix of the error correction code is often used to generate the syndrome for the reconciliation problem. Thus an equivalent channel coding problem can be solved instead of the above-mentioned source coding with side information.

As depicted in Fig. 1 we describe reconciliation with error correction codes in two steps. The first step is *discretization*, which transforms the continuous Gaussian source X_B into an m-bit source $\mathcal{Q}(X_B) \in \mathcal{A}$ with its binary representation vectors $(X_B^{m-1}, \ldots, X_B^1, X_B^0)$, where X_B^{m-1} is the most significant bit and X_B^0 is the least significant bit. We note that there is an inherent information loss due to the discretization process of the source. The second step is *source coding with side information*. In reverse reconciliation as considered here, Bob sends an encoding (compressed version) of $\mathcal{Q}(\vec{X}_B)$ to Alice, such that she can infer $\mathcal{Q}(\vec{X}_B)$ with high probability using her own source $\vec{X}_A$ as side information.

The efficiency β of this process is defined as

$$\beta = \frac{H(\mathcal{Q}(X_B)) - R^s}{I(X_B; X_A)}, \tag{1}$$

where $I(X_B; X_A)$ is the mutual information and $H(\mathcal{Q}(X_B)) - R^s$ is the net shared information per symbol between the two parties [9,18], with $H(\cdot)$ being the Shannon entropy and R^s being the source coding rate. Thus the practical efficiency of the reconciliation depends on the ability to design very good discretizers and highly efficient compression codes with minimum possible source coding rate. Note that Slepian and Wolf [16] have shown that $H(Y|Z)$ is the lower bound to the source coding rate when decoding Y given side information Z. Therefore $R^s \geqslant H(\mathcal{Q}(X_B)|X_A)$.

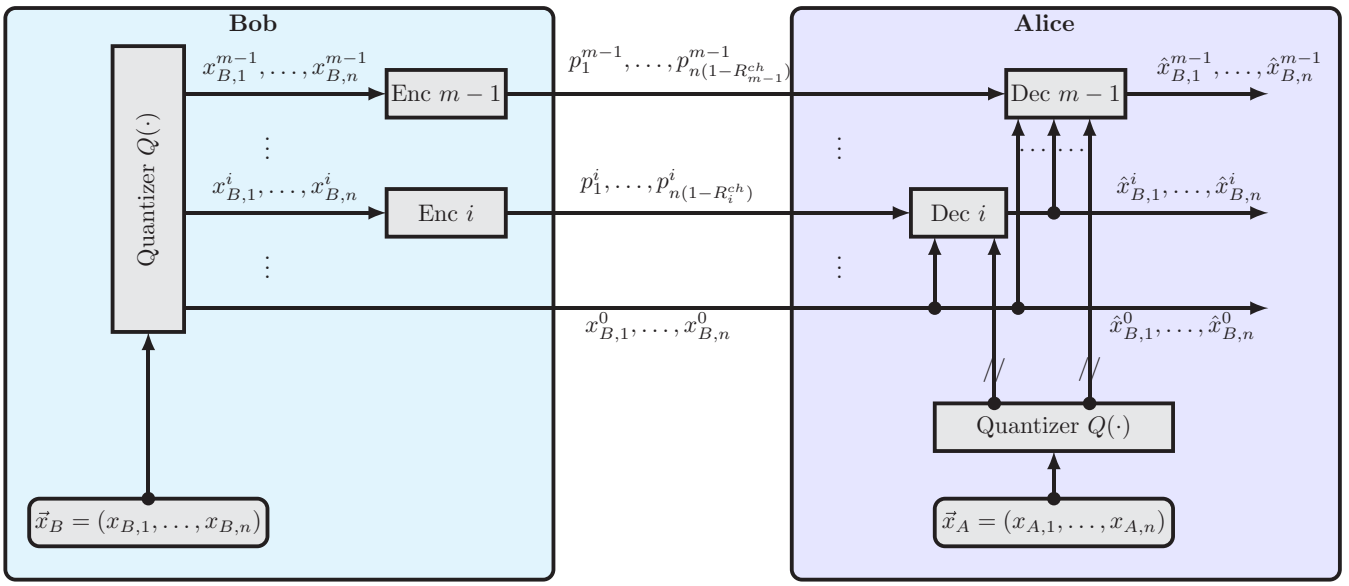

FIG. 2. The MLC-MSD scenario for the reverse reconciliation. First the input source is quantized into an m-bit source. Then each of the m sources are encoded and sent to Alice. The decoder has the side information from its own source and with the m encoded sources produces an estimate of the quantized source. Usually, we transmit the least significant bits (LSBs) directly to the channel.

III. SLICE RECONCILIATION BASED ON MLC-MSD

For the second step we use an MLC-MSD scheme, which is presented in Fig. 2 using m levels corresponding to the number of bits used for discretization. Bob encodes level i with source coding rate R_i^s. Using the Slepian-Wolf theorem, the source coding rate R_i^s is lower bounded by the conditional entropy of the ith bit of X_B, given side information X_A and all the lower bits of X_B:

$$R_i^s \geqslant H\left(X_B^i | X_A, X_B^{i-1}, \ldots, X_B^0\right) = R_i^{s^*}. \tag{2}$$

The total source coding rate is given by summing over the individual source code rates R_i^s:

$$R^s = \sum_{i=0}^{m-1} R_i^s.$$

Using the parity of error correction codes to generate syndromes, the corresponding channel coding rate for level i is given by

$$R_i^{ch} = 1 - R_i^s \leqslant 1 - H\left(X_B^i | X_A, X_B^{i-1}, \ldots, X_B^0\right). \tag{3}$$

Thus the reconciliation efficiency of the system in terms of the channel coding rates reads

$$\beta = \frac{H(\mathcal{Q}(X_B)) - m + \sum_{i=0}^{m-1} R_i^{ch}}{I(X_B; X_A)}. \tag{4}$$

By optimizing the individual codes for each level the overall efficiency of reconciliation can be maximized. The channel coding rate R_i^{ch} is bounded by

$$1 - H\left(X_B^i | X_B^{i-1}, \ldots, X_B^0\right) \leqslant R_i^{ch} \leqslant 1 - R_i^{s^*}, \tag{5}$$

where the upper bound is given by Eq. (3). This implies that

$$0 \leqslant \beta \leqslant \frac{H(\mathcal{Q}(X_B)) - m + \sum_{i=0}^{m-1} R_i^{\mathrm{ch}^*}}{I(X_B; X_A)}, \tag{6}$$

where $R_i^{\mathrm{ch}^*} = 1 - R_i^{\mathrm{s}^*}$.

We will now discuss how to calculate the source coding rates $R_i^{s^*}$ for the individual levels. Using the chain rule, we can describe the total mutual information as a summation of the conditional mutual information of the individual levels

$$\begin{aligned} I(X_A; Q(X_B)) &= I\left(X_A; X_B^{m-1}, \ldots, X_B^0\right) \\ &= I\left(X_A; X_B^0\right) + I\left(X_A; X_B^1 | X_B^0\right) + \cdots \\ &\quad + I\left(X_A; X_B^i | X_B^{i-1}, \ldots, X_B^0\right) + \cdots \\ &\quad + I\left(X_A; X_B^{m-1} | X_B^{m-2}, \ldots, X_B^0\right). \end{aligned} \tag{7}$$

Thus we define the conditional mutual information for level i as

$$\begin{aligned} I_i &:= I\left(X_A; X_B^i | X_B^{i-1}, \ldots, X_B^0\right) \\ &= H\left(X_B^i | X_B^{i-1}, \ldots, X_B^0\right) - H\left(X_B^i | X_A, X_B^{i-1}, \ldots, X_B^0\right). \end{aligned} \tag{8}$$

Using Eq. (2), we obtain an analytical way to calculate the source coding rate for level i,

$$R_i^s \geqslant H\left(X_B^i | X_B^{i-1}, \ldots, X_B^0\right) - I_i. \tag{9}$$

The conditional mutual information I_i can thereby be calculated as follows [19]:

$$\begin{aligned} I_i &= I\left(X_A; X_B^i | X_B^{i-1}, \ldots, X_B^0\right) \\ &= I\left(X_A; X_B^{m-1}, \ldots, X_B^i | X_B^{i-1}, \ldots, X_B^0\right) \\ &\quad - I\left(X_A; X_B^{m-1}, \ldots, X_B^{i+1} | X_B^i, \ldots, X_B^0\right). \end{aligned} \tag{10}$$

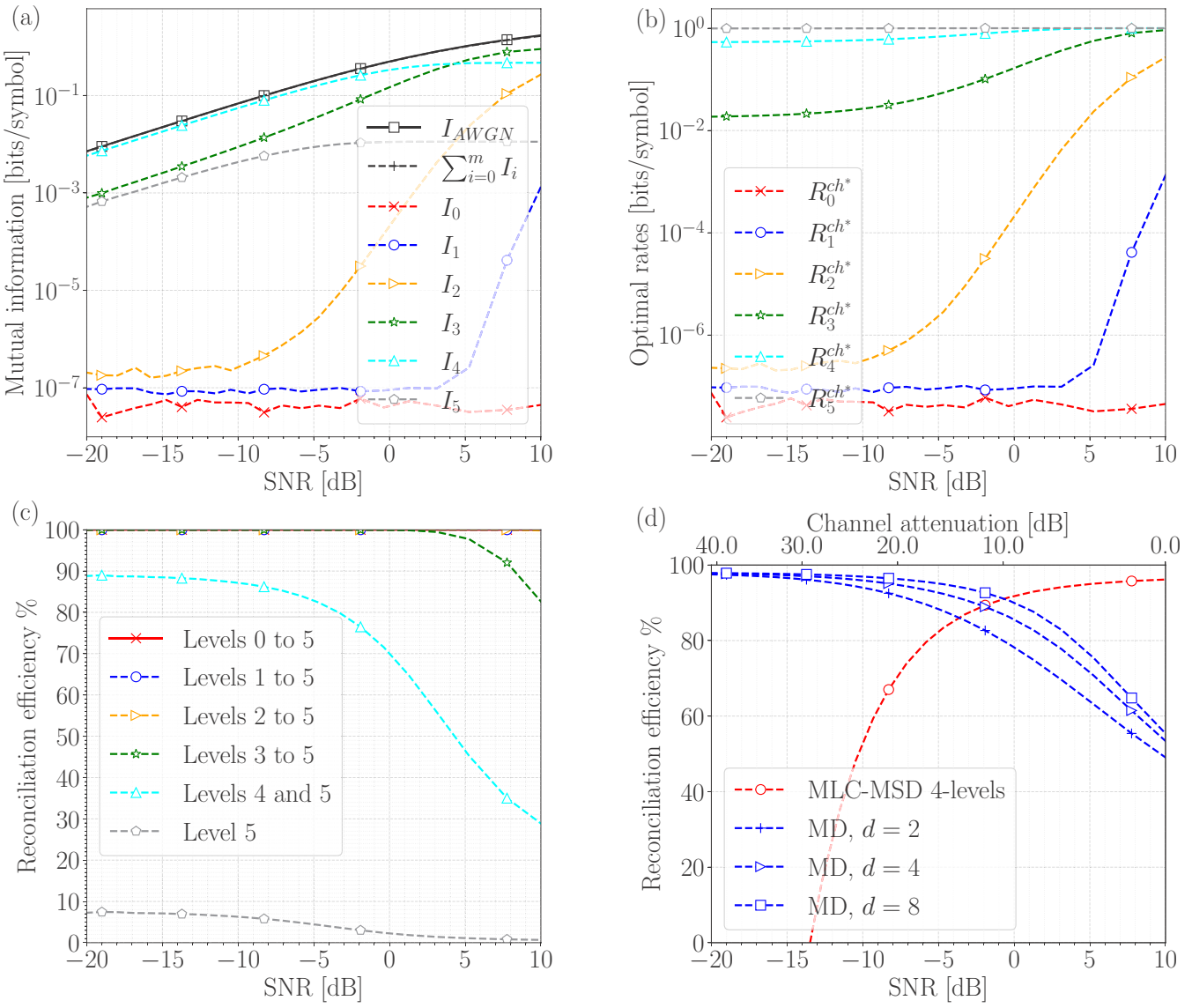

FIG. 3. Simulation of (a) the individual conditional mutual information, (b) the equivalent optimum channel coding rates $R_i^{\text{ch}^*} = 1 - R_i^{\text{s}^*}$ minimizing the leaked information about the raw key, and (c) the corresponding theoretical reconciliation efficiency depending on which levels are encoded and which are transmitted without encoding. The labels of the traces indicate which levels are encoded. (d) Practical reconciliation efficiencies for the MD and the MLC-MSD reconciliation schemes where code efficiency $\beta^{\text{c}} = 0.98$. For the simulation we normalized Alice's and Bob's Gaussian data with zero mean to a standard deviation of 1 and used a digitizer with 6-bit resolution and a range of six standard deviations.

Each term on the right-hand side can be computed separately by

$$I\left(X_A; X_B^{m-1}, \ldots, X_B^i | X_B^{i-1}, \ldots, X_B^0\right) = \mathbb{E}_{x_B^{i-1}, \ldots, x_B^0 \in \{0,1\}^i}\left\{I\left(X_A; X_B^{m-1}, \ldots, X_B^i | x_B^{i-1}, \ldots, x_B^0\right)\right\},$$

where we average over all possible combinations of $x_B^{i-1}, \ldots, x_B^0$. The full characterization of I_i requires a set of probability density functions (PDFs) $f_{X_A|X_B^i}(x_A|x_B^i)$, which are defined as

$$\left\{f_{X_A|X_B^i,\ldots,X_B^0}\left(x_A|x_B^i, x_B^{i-1}, \ldots, x_B^0\right) | \left(x_B^{i-1}, \ldots, x_B^0\right) \in \{0,1\}^i\right\},$$

where

$$f_{X_A|X_B^i,\ldots,X_B^0}\left(x_A|x_B^i, x_B^{i-1}, \ldots, x_B^0\right) = \mathbb{E}_{b \in \mathbf{S}(x_B^i,\ldots,x_B^0)}\{f_{X_A|X_B}(x_A|b)\}.$$

Here, the signal point b is taken from the subset $\mathbf{S}(x_B^0 \cdots x_B^i)$. A detailed description of the set partitioning can be found in Appendix A. $f_{X_A|X_B}(x_A|x_b)$ is the conditional probability distribution describing Alice's outcome conditioned on Bob's, which is described in Appendix B.

A. Simulation results

Figure 3 shows simulation results for a CV-QKD protocol using Gaussian modulation with 6-bit discretization and a

range of the discretizer of six standard deviations. Current security proofs for CV-QKD require discretizers with a high resolution and large range [20]. In the figure the mutual information, the optimal channel coding rates, and the reconciliation efficiency are plotted versus the signal-to-noise ratio (SNR). The SNR is determined by the modulation variance of the coherent state ensemble at the transmitter (which is usually optimized for a certain channel to obtain the maximum secret key rate), the optical loss of the channel, and the noise in the channel.

The conditional mutual information of each level and the total and the expected mutual information of the additive white Gaussian noise (AWGN) channel are depicted in Fig. 3(a). For SNRs below 0 dB the 3 least significant bits (I_0, I_1, I_2) do not contribute (much) to the total mutual information; only the most significant bits do. Note that in the figure the conditional mutual information flattens out towards lower SNRs due to the numerical accuracy of the calculation. For the binary mapping used in this simulation the second- and third-most-significant bits contain the highest amount of conditional mutual information, while the most significant bit has a lower but still considerable amount. This order can be explained by the fact that the probability distribution conditioned on Bob's measurement result is a Gaussian distribution centered at Bob's result and that therefore the sign in the binary representation does not contain most of the information.

The corresponding channel coding rates for each level are shown in Fig. 3(b). As expected from the mutual information the channel coding rates for the 3 least significant bits (R_0^{ch}, R_1^{ch}, R_2^{ch}) are very close to 0 for low SNR. Channel code rates below 0.01 are impractical as such rates would yield a very low throughput of the decoder. Therefore those bits can be transmitted without encoding instead of using a code, and only the 3 most significant bits have to be reconciled. The channel coding rate for the most significant bit (R_5^{ch}) is very close to 1, and we can recover it by using a Bose, Chaudhuri, and Hocquenghem (BCH) code.

Figure 3(c) shows the reconciliation efficiency depending on which levels are encoded. The other remaining bits are simply transmitted without encoding. For the SNR range considered here, high efficiency can be achieved for transmitting a maximum of 3 least significant bits without encoding (encode levels 3–5). If a fourth bit is transmitted without encoding (encode only levels 4 and 5), the efficiency drops by 10–20% at SNRs smaller than −5 dB.

Due to finite-size effects the efficiency of the actual implemented code is always lower than asymptotically possible. Using codes with efficiency β^c instead of the optimal efficiency, the overall reconciliation efficiency of the MLC-MSD scheme drops significantly at low SNRs. In Figure 3(d) we compare the reconciliation efficiency of MLC-MSD with MD reconciliation [8] for $\beta^c = 98\%$. The figure shows that MD reconciliation is the method of choice for SNRs below about 0 dB. For SNRs higher than 0 dB the MLC-MSD scheme has better performance. In the plot we also indicate to which channel attenuation the SNR may correspond. Assuming an (channel input related) excess noise of 0.0015 shot noise units [21], we optimized the secret key rate for each channel attenuation value to obtain the optimal modulation variance. We note that in practice the chosen modulation variance and thus the SNR may be different, for instance, due to the presence of phase noise. In both schemes, the MLC-MSD and the MD, MET-LDPC codes with low channel coding rates are required to get very high efficiency.

IV. GENERALIZED EXTRINSIC INFORMATION TRANSFER CHART

Density evolution (DE) is the main tool for analyzing the average asymptotic behavior of the belief propagation decoder for MET-LDPC code ensembles with infinite block length and infinite number of iterations. The density evolution analysis is in general simplified by the all-one-code-word assumption, by the channel symmetry, and by working in the log-likelihood ratio domain [22–24]. Density evolution for MET-LDPC codes is described by the recursion

$$\begin{aligned}\mathbf{P}^{l+1} &= \lambda(\mathbf{R}, \rho(\mathbf{P}^l)),\\ \mathbf{Q}^{l+1} &= \rho(\mathbf{P}^l),\end{aligned} \tag{11}$$

where $\rho(\mathbf{x})$ and $\lambda(\mathbf{r}, \mathbf{x})$ are the edge-perspective representations of the MET-LDPC code. They are described in Appendix C. $\mathbf{P}^l = (P_1^l, \ldots, P_{n_e}^l)$ denotes the vector of messages passed from variable nodes to check nodes in iteration l assuming that $\mathbf{P}^0 = \mathbf{\Delta_0}$. Here, $\mathbf{\Delta_0}$ describes a vector of densities where each density is given by δ_0, the Dirac delta function at point zero. Similarly, $\mathbf{R}$ denotes the received distributions. The density evolution for MET-LDPC codes is a generalization of the density evolution for irregular LDPC codes with n_e dimensions. For irregular codes, Eq. (11) reduces to a one-dimensional recursion.

The concepts of EXIT and G-EXIT charts were first introduced for irregular LDPC codes [25], for which they provide a graphical representation of the convergence behavior. Running the density evolution for irregular LDPC codes, one can monitor the intermediate densities at the output of variable nodes and check nodes. Given two families of L densities $\{c_{\epsilon_i}\}$ and $\{a_\epsilon\}$ related to P and Q and parametrized by ϵ, the G-EXIT function is

$$G(c_{\epsilon_i}, a_\epsilon) = \frac{\int_z \int_\omega a_\epsilon(z) \frac{\partial c_{\epsilon_i}(\omega)}{\partial \epsilon} \log_2(1 + e^{-z-\omega}) d\omega dz}{\int_\omega \frac{\partial c_{\epsilon_i}(\omega)}{\partial \epsilon} \log_2(1 + e^{-\omega}) d\omega}. \tag{12}$$

The G-EXIT curve is then given in parametric form by $\{H(c_{\epsilon_i}), G(c_{\epsilon_i}, a_\epsilon)\}$, where

$$H(c_{\epsilon_i}) = \int_{-\infty}^{\infty} c_{\epsilon_i}(\omega) \log_2(1 + e^{-\omega}) d\omega.$$

In a G-EXIT chart we plot the G-EXIT curve of the variable nodes and the inverse of the dual G-EXIT curve of the check nodes. The dual G-EXIT curve is defined in parametric form as $\{G(a_\epsilon, c_{\epsilon_i}), H(a_\epsilon)\}$ [25]. Note that for a binary linear code and transmission over *binary memoryless symmetric* channels the G-EXIT and the dual G-EXIT curves have equal area [25].

For irregular LDPC codes the corresponding G-EXIT chart is a two-dimensional plot. The extension of the G-EXIT charts for MET-LDPC codes is not straightforward as for $n_e = 2$ edge types the G-EXIT chart would be three dimensional and for MET-LDPC codes with higher edge types, i.e., $n_e > 2$,

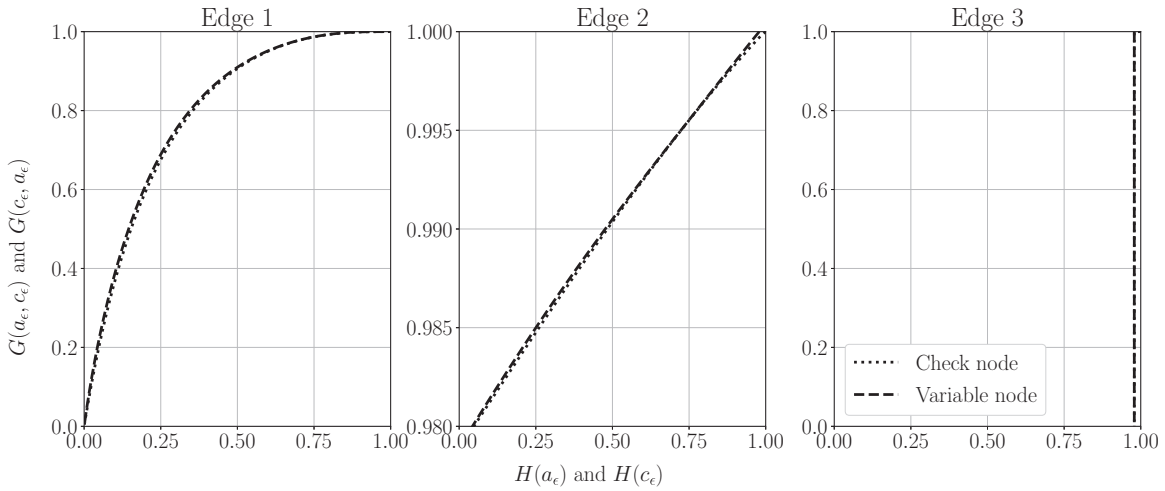

FIG. 4. G-EXIT charts for all edge types of a MET-LDPC code with channel coding rate 0.02.

the G-EXIT chart would be even higher dimensional, which renders a graphical representation impossible.

Here, we propose to solve this by plotting n_e two-dimensional G-EXIT charts by monitoring the densities along each edge type separately. To plot the two-dimensional G-EXIT charts, we apply the G-EXIT functions of Eq. (12) to the density families at the output of variable nodes and check nodes along each edge type, i.e., for each component of the vectors in Eq. (11). The combination rules of the densities for the ith component are given by

$$c_{\epsilon_i} = \sum_j \lambda_j^i \, \mathbf{R}^{\otimes \mathbf{b}} \otimes [\mathbf{Q}^{l+1}]^{\otimes \mathbf{d}}, \tag{13}$$

$$a_\epsilon = \sum_j \rho_j^i [\mathbf{P}^l]^{\boxtimes \mathbf{d}}, \tag{14}$$

where $\mathbf{Q}^{\otimes \mathbf{d}} = \bigotimes_{k=1}^{n_e} Q_k^{\otimes d_k}$, $\mathbf{R}^{\otimes \mathbf{b}} = \bigotimes_{k=0}^{n_r} R_k^{\otimes b_k}$. Here, $\otimes$ is the convolution in variable nodes. Similarly, $\mathbf{P}^{\boxtimes \mathbf{d}} = \boxtimes_{k=1}^{n_e} P_k^{\boxtimes d_k}$, where $\boxtimes$ is the convolution of check nodes [26].

In Fig. 4 we show the G-EXIT chart for a MET-LDPC code with channel code rate of 0.02. Since the code has $n_e = 3$ edge types, the chart consists of three plots, each representing the G-EXIT chart for one edge type.

From the n_e two-dimensional G-EXIT curves we can determine the convergence behavior of the MET-LDPC code. If the curves cross each other in any of the graphs, the MET-LDPC code does not converge. In fact, in our simulations, either the curves for all edge types showed a crossover or the curves for none of them did (see, e.g., Appendix C, Fig. 8). In addition, the gap between the two curves is minimal in all graphs for the SNR corresponding to the threshold of the curve. Highly efficient codes have a very small gap. This can be seen in the G-EXIT charts for the codes presented in Sec. V.

V. MET-LDPC CODES FOR USE IN CV-QKD IMPLEMENTATIONS

We will now present a set of highly efficient MET-LDPC codes which are optimized for the binary input (BI) AWGN channel. These codes can be used in MD reconciliation or MLC-MSD for lower levels. For example, for MD reconciliation a channel code of rate 0.02 is required at an SNR of -15.4 dB, and a channel code of rate 0.01 is required for an SNR of -18.4 dB. For MLC-MSD the low channel code rates might be used at the lower levels. As shown in Fig. 3(b) a channel code of rate 0.02 should be used at an SNR of 5 dB for the fourth level. The exact channel code rates depend, however, on the number of levels and the range of the digitizer. For example, in Ref. [10] a channel code of rate 0.01 can be used at an SNR of 3 dB, and a channel code of rate 0.02 can be used at an SNR of 7 dB.

The design of MET-LDPC code involves solving an optimization problem [27,28]. The optimization algorithm searches between all the valid degree distributions to find a code close to the threshold. Here, we reduced the search space by focusing on codes in the cascade structure (see Appendix C). The cascade structure allows us to find a MET-LDPC code by first designing an irregular LDPC code and then growing the MET-LDPC code on top.

The codes are presented in Table I. We show the degree distributions of MET-LDPC codes for channel coding rates 0.01, 0.02, 0.05, and 0.1, while for comparison we also include previously published channel codes of rate 0.02 and 0.5. The asymptotic threshold σ^*_{DE} of these codes was obtained by running the density evolution for each code until getting an error probability of less than 10^{-10}. The required number of iterations to find the threshold of each code and the corresponding SNR are also specified in the table. The asymptotic efficiency of each code is defined by

$$\beta_{\mathrm{DE}} = \frac{R^{ch}}{C(\sigma^*_{\mathrm{DE}})}, \tag{15}$$

where $C(\sigma^*_{\mathrm{DE}})$ denotes the Shannon capacity at σ^*_{DE}.

A visual representation of the convergence behavior of the channel code with rate 0.02 is presented in the form of G-EXIT charts for the separate edge types in Fig. 4. The high efficiency of the code results in the two curves in each chart being very close. For edge type 3, which belongs to variable nodes of degree 1, the corresponding G-EXIT curves are two completely matching vertical lines at $x = 0.9799$. The value

TABLE I. List of MET-LDPC codes with different channel coding rates R^{ch}. We present the degree distributions and compare their convergence threshold σ^*_{DE} using density evolution for the BI AWGN channel with the threshold at Shannon capacity σ^*_{Sh}. The SNR is specified at the threshold. β_{DE} is the asymptotic efficiency of the code. The maximum number of iterations for all the codes was 1000 except for the proposed code with rate 0.02, for which a maximum of 2000 iterations was allowed.

R^{ch}	Degree distribution	σ^*_{DE}	SNR (dB)	σ^*_{Sh}	β_{DE} (%)
0.01	$\nu(\mathbf{r},\mathbf{x}) = 0.01125\, r_1\, x_1^2\, x_2^{111} + 0.00875\, r_1\, x_1^3\, x_2^{118} + 0.98\, r_1\, x_3,$ $\mu(\mathbf{x}) = 0.0053125\, x_1^3 + 0.0046875\, x_1^7 + 0.65875\, x_2^2\, x_3^1 + 0.32125\, x_2^3\, x_3^1$	8.37	−18.45	8.46	97.8
0.02	$\nu(\mathbf{r},\mathbf{x}) = 0.0225\, r_1 x_1^2 x_2^{52} + 0.0175\, r_1 x_1^3 x_2^{57} + 0.96\, r_1 x_3^1,$ $\mu(\mathbf{x}) = 0.0165\, x_1^4 + 0.0035\, x_1^9 + 0.2475\, x_2^3 x_3^1 + 0.7125\, x_2^2 x_3^1$	5.93	−15.46	5.96	98.8
0.02 (Appendix A of Ref. [5])	$\nu(\mathbf{r},\mathbf{x}) = 0.0225\, r_1 x_1^2 x_2^{57} + 0.0175\, r_1 x_1^3 x_2^{57} + 0.96\, r_1 x_3^1,$ $\mu(\mathbf{x}) = 0.010625\, x_1^3 + 0.009375\, x_1^7 + 0.6\, x_2^2 x_3^1 + 0.36\, x_2^3 x_3^1$	5.91	−15.43	5.96	98.1
0.05	$\nu(\mathbf{r},\mathbf{x}) = 0.05625\, r_1\, x_1^2\, x_2^{20} + 0.04375\, r_1\, x_1^3\, x_2^{25} + 0.90\, r_1\, x_3,$ $\mu(\mathbf{x}) = 0.0265625\, x_1^3 + 0.0234375\, x_1^7 + 0.48125\, x_2^2\, x_3^1 + 0.41875\, x_2^3\, x_3^1$	3.69	−11.34	3.73	97.8
0.10	$\nu(\mathbf{r},\mathbf{x}) = 0.075\, r_1 x_1^2 x_2^{21} + 0.05\, r_1 x_1^3 x_2^{20} + 0.875\, r_1 x_3^1,$ $\mu(\mathbf{x}) = 0.025\, x_1^{12} + 0.825\, x_2^3 x_3^1 + 0.050\, x_2^2 x_3^1$	2.56	−8.16	2.59	97.5
0.50 (Ref. [26])	$\nu(\mathbf{r},\mathbf{x}) = 0.20\, r_0\, x_2^3\, x_3^3 + r_1\,(0.50\, x_1^2 + 0.30\, x_1^3 + 0.2\, x_4),$ $\mu(\mathbf{x}) = 0.10\, x_1^3 x_2^2 + 0.4\, x_1^4\, x_2^1 + 0.2\, x_3^3\, x_4^1$	0.965	0.305	0.9787	98.2

corresponds to the entropy of the channel $H(\sigma^*_{\mathrm{DE}})$. G-EXIT charts of all other codes can be found in Appendix C 2.

To evaluate the finite-block-length performance of the channel code with rate 0.02, Fig. 5(a) shows the simulated frame error rate (FER). We compare our results with the simulated FER of the code of Ref. [5] with channel code rate 0.02, which has been used in many implementations. In the figure we plot the FER for block lengths of 1.024×10^6, while for the code of Ref. [5] block lengths of 8.192×10^5 and 1.6384×10^6 were used. For both codes a progressive edge growth (PEG) algorithm [29] was used to construct quasicyclic MET-LDPC codes with $Z\times Z$ circulant permutation matrices. Our designed channel code outperforms both variants of the channel code of Ref. [5]. More specifically, compared with code of length 8.192×10^5, our code provides 0.1 dB additional gain at FER = 0.01. In addition, our code always provides lower FER in spite of its shorter block length in comparison with the other code of length 1.6384×10^6. The vertical dashed lines in the figure display the Shannon asymptotic threshold for the channel code rate 0.02 on a BI AWGN channel and the asymptotic threshold obtained by density evolution for our code and the code of Ref. [5], respectively.

The efficiencies of the two codes versus the FER are compared in Fig. 5(b). It can be observed that for all FERs the efficiency of our code is higher than that of the code of Ref. [5] even with longer block length. For instance, for an efficiency

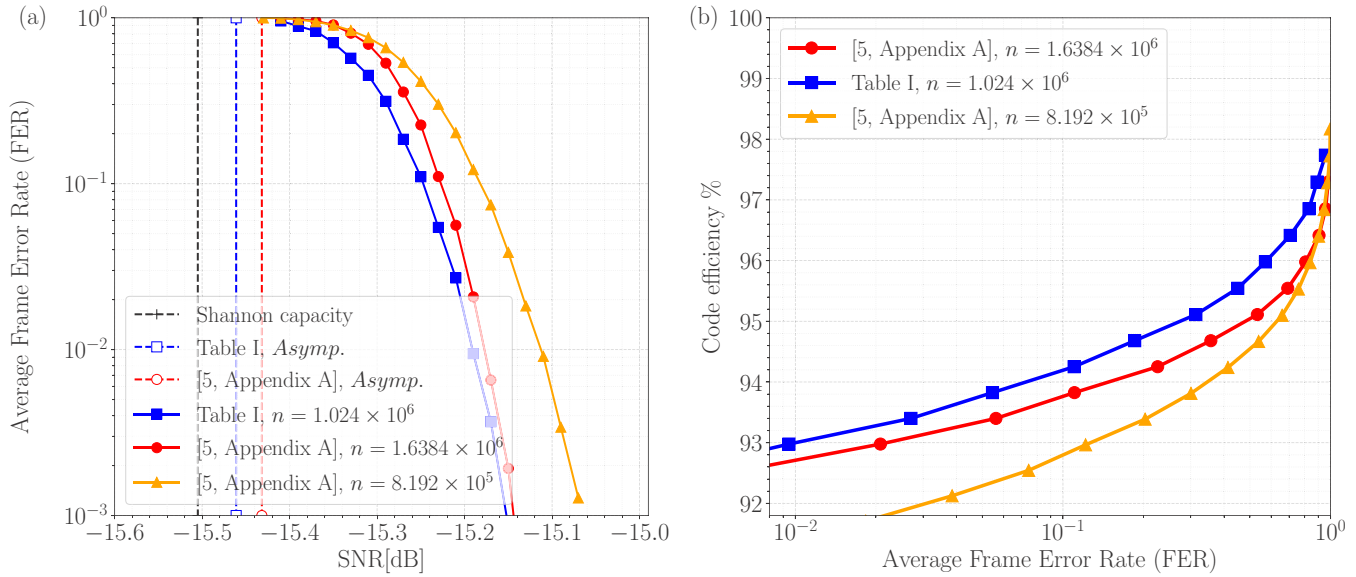

FIG. 5. Simulation of (a) the frame error rate vs SNR for MET-LDPC codes with channel coding rate 0.02 and (b) the efficiency vs frame error rate for MET-LDPC codes with channel coding rate 0.02. To plot the FER curves, we set the maximum number of iterations to 500, and for each point, 100 frames of errors are counted. We used $Z = 256$ for the solid orange curve and $Z = 1024$ for the solid red and blue curves.

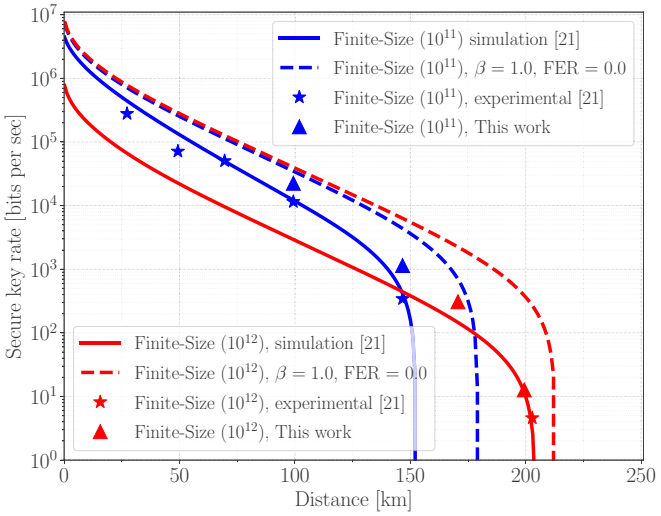

FIG. 6. Numerical simulation of secret key rate comparing the performance of our codes with previous codes. The experimental points and the simulation parameters are taken from Table I of Ref. [21]: The repetition rate is 5 MHz, the fraction of symbols for parameter estimation is $\nu = 0.1$, the modulation variance V_A has been optimized, and the fiber attenuation is $\alpha = 0.16$ dB/km. The (input related) excess noise is 0.0086 shot noise units for the blue curves and 0.0081 shot noise units for the red ones. The electronic noise is 0.2717 and 0.1523 shot noise units, respectively, and the trusted receiver efficiency is 61.34%.

of 95% the two variants of code in Ref. [5] have a FER of 0.6 and 0.53, respectively, but our code is able to provide the same efficiency with a FER as low as 0.26. This improvement corresponds to a 1.85 and 1.57 times higher secret key rate in terms of bits per second for the same channel. In addition, for an accepted FER = 0.5, the efficiency of our code is close to 96%, while the efficiencies of the other two variants are less than or equal to 95%.

To show the performance of our MET-LDPC codes on CV-QKD, we simulated the achievable secret key rate versus distance as shown in Fig. 6. Taking finite-size effects into account, the secret key rate for reverse reconciliation on CV-QKD is [30]

$$K_{\text{finite}} = f(1-\text{FER})(1-\nu)[\beta I_{AB} - \mathcal{X}_{EB} - \Delta(n_{\text{privacy}})], \tag{16}$$

where f is the repetition rate, FER is the frame error rate of the reconciliation, ν is the fraction of symbols used for parameter estimation, β is the reconciliation efficiency, I_{AB} is the classical mutual information between Alice and Bob, $\mathcal{X}_{EB}$ bounds Eve's Holevo information on Bob's variable, and $\Delta(n_{\text{privacy}})$ is the finite-size penalty. For the simulation we use the parameters from Ref. [21] and compare different error reconciliation codes. In particular, we use $f = 5$ MHz and assume that the length of the data for the privacy amplification is $n_{\text{privacy}} = 10^{11}$ (blue symbols in Fig. 6) or 10^{12} (red symbols). The number of the quantum symbols is $n_{\text{quantum}} = 10 \times n_{\text{privacy}}$ ($\nu = 0.1$).

In Fig. 6 the solid curves show the simulated secret key rate fitting the experimental points obtained in Ref. [21], which are marked with a star. Using our codes for information reconciliation, higher secret key rates, marked with the triangles, could be achieved instead. For comparison we also plot the maximum achievable secret key rate assuming an information reconciliation efficiency of 1 and a FER of 0. From the set of codes presented in Table I we use the code with channel code rate 0.1 for a distance of 99.31 km (15.89-dB losses), the code with rate 0.02 for 146.62 km (23.46-dB losses), and the code with rate 0.01 for 170.62 km (28.65-dB losses). Additionally, we use the repetition technique [6] with factor 3 for channel code rate 0.01 to extend the distance up to 200 km (32.10-dB losses). Due to the higher efficiency and lower FER the achievable secret key rates are about a factor of 2 higher than what is achievable with the previous codes.

VI. CONCLUSION

In summary, we presented a practical approach to calculate the optimal channel code rates of a MLC-MSD reconciliation scheme suitable for reverse reconciliation in Gaussian modulation CV-QKD. The optimal channel code rates in this scheme depend on the number of individual levels which is given by the resolution of the Gaussian-distributed coherent states and thus determined by the QKD system's implementation, and on the SNR. Using a 6-bit discretization and a range of six standard deviations, for SNRs below 0 dB the reduced code efficiency for each level due to finite block size of the error-correcting codes is detrimental for the overall efficiency, and thus one has to resort to MD reconciliation.

Furthermore, we introduced the powerful tool and concept of G-EXIT charts for MET-LDPC codes to visualize their performance and convergence behavior, and we presented a set of highly efficient MET-LDPC codes with channel coding rates 0.1, 0.05, 0.02, and 0.01 with asymptotic efficiencies higher than 97%. The finite-block-length performance of the 0.02 code was evaluated, and a lower FER was found than for previously available codes. Specifically, the MET-LDPC code with channel coding rate 0.01 can be used in conjunction with MD reconciliation to achieve distances longer than 170 km [21], a regime that is only possible to achieve with information reconciliation performing closely at the theoretical optimum. We therefore believe that our codes will find wide applications in CV-QKD implementations.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 820466 (CiViQ). The authors thank the Quantum Innovation Center Qubiz funded by the Innovation Fund Denmark for support. H.M., T.G., and U.L.A. acknowledge support from the Danish National Research Foundation, Center for Macroscopic Quantum States (bigQ, DNRF142), and from Innovation Fund Denmark (CryptQ, 0175-00018B).

APPENDIX A: SET PARTITIONING

We consider a quantization scheme with $M = 2^m$, $m > 1$, signal points in a D-dimensional real signal space, with signal points taken from the signal set $\mathbf{S} = \{\vec{a}_0, \vec{a}_1, \ldots, \vec{a}_{M-1}\}$ with probabilities $\Pr\{\vec{a}_k\}$. Each signal point has its equiva-

lent binary form defined by a bijective mapping $\vec{a} = \mathcal{M}(\vec{x})$ of binary representation vectors $\vec{x} = (x_B^{m-1}, \ldots, x_B^0)$ to signal points $\vec{a} \in \mathbf{S}$. Two well-defined mappings are binary and Gray mapping. As an example for $m = 3$ levels, in one-dimensional signal space ($D = 1$), the $M = 2^3$ signal points are taken from $\mathbf{S} = \{-7, -5, -3, -1, +1, +3, +5, +7\}$. Fixing the values of coordinates i to 0, i.e., $x_B^i, \ldots, x_B^0$, we obtain subsets of the signal set $\mathbf{S}$:

$$\begin{aligned}&\mathbf{S}\big(x_B^i, \ldots, x_B^0\big)\\ &\quad = \{\vec{a} = \mathcal{M}(\vec{x}) \,|\, \vec{x} = \big(b^{m-1}, \ldots, b^{i+1}, x_B^i, \ldots, x_B^0\big),\\ &b^j \in \{0, 1\},\ j = i+1, \ldots, m-1\}. \end{aligned} \tag{A1}$$

For more details about set partitioning and mapping, see Ref. [19]. For example, for the above-mentioned constellation points with $M = 8$ with binary partitioning we have

$$\begin{aligned}\mathbf{S}\big(x_B^0 = 0\big) &= \{\vec{a} = \mathcal{M}(\vec{x}) | \vec{x} = \{000, 010, 100, 110\}\}\\ &= \{-7, -3, +1, +5\},\\ \mathbf{S}\big(x_B^1 x_B^0 = 10\big) &= \{\vec{a} = \mathcal{M}(\vec{x}) | \vec{x} = \{010, 110\}\}\\ &= \{-3, +5\},\\ \mathbf{S}\big(x_B^2 x_B^1 x_B^0 = 010\big) &= \{\vec{a} = \mathcal{M}(\vec{x}) | \vec{x} = \{010\}\} = \{-3\}.\end{aligned}$$

APPENDIX B: CLASSICAL STATISTICAL REPRESENTATION

In the following we assume that the symbols of Alice, X_A, and Bob, X_B, are jointly distributed according to a bivariate normal distribution with zero mean. The bivariate normal distribution can be described by

$$f_{X_B,X_A}(x_B, x_A) = \frac{\exp\left(-\frac{(x_A, x_B)\Sigma^{-1}(x_A, x_B)^T}{2}\right)}{2\pi\sqrt{|\Sigma|}} \tag{B1}$$

with the covariance matrix

$$\Sigma = \begin{bmatrix} \sigma_A^2 & \rho\,\sigma_A\,\sigma_B \\ \rho\,\sigma_A\,\sigma_B & \sigma_B^2 \end{bmatrix}, \tag{B2}$$

where

$$\rho = \frac{\mathbb{E}\{X_A X_B\}}{\sigma_A \sigma_B} \tag{B3}$$

is the correlation coefficient between X_A and X_B and σ_A and σ_B denote their standard deviations, respectively. We empirically estimate the covariance matrix during the parameter estimation phase of the quantum key distribution protocol. We then normalize Alice's and Bob's data by dividing by their respective standard deviation, i.e.,

$$\begin{aligned} x_A^j &\to x_A^j/\sigma_A,\\ x_B^j &\to x_B^j/\sigma_B, \end{aligned}$$

such that

$$\Sigma \to \Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}. \tag{B4}$$

The conditional probability distribution describing Alice's outcome conditioned on Bob's is given by

$$f_{X_A|X_B}(x_A|x_B) = \mathcal{N}(\rho x_B, (1 - \rho^2)). \tag{B5}$$

TABLE II. Degree structure of a MET-LDPC code with channel coding rate 0.02. $\sigma_{Sh}^* = 5.96$, $\sigma_{\mathrm{DE}}^* = 5.93$, and $\beta_{\mathrm{DE}} = 98.8\%$.

$\nu_{\mathbf{bd}}$	**b**		**d**			$\mu_{\mathbf{d}}$	**d**		
0.0225	0	1	2	52	0	0.0165	4	0	0
0.0175	0	1	3	57	0	0.0035	9	0	0
0.96	0	1	0	0	1	0.2475	0	3	1
						0.7125	0	2	1

As we discussed in Sec. II, let us denote the quantized version of X_B by $Q(X_B)$ with its binary equivalent vector $(X_B^{m-1}, \ldots, X_B^1, X_B^0)$. Considering a fixed step size δ for discretization, the entropy of the quantized source can be approximated by

$$H(Q(X_B)) \approx h(X_B) - \log_2 \delta,$$

where $h(X_B)$ is the differential entropy defined for the continuous variable X_B. A similar quantization can be applied on Alice's side to get $Q(X_A)$. This also holds for the conditional entropy; that is,

$$H(Q(X_B)|Q(X_A)) \approx h(X_B|X_A) - \log_2 \delta.$$

If m is large enough, and thus δ is small, we can approximate

$$I(Q(X_B); Q(X_A)) \approx I(Q(X_B); X_A) \approx I(X_B; X_A),$$

where the equality holds when $\delta \to 0$.

APPENDIX C: MET-LDPC CODES

1. MET-LDPC code ensemble

Multiedge-type LDPC (MET-LDPC) codes are a generalization of the concept of irregular LDPC codes [26,31]. These codes provide improvements in performance and complexity by giving more flexibility over different edge types. In this structure, each node is characterized by the number of connections (sockets) to edges of each edge type. It is noteworthy to mention that an irregular LDPC code is a single-edge-type LDPC (SET-LDPC) code. Using MET-LDPC codes, we are able to design capacity-achieving codes without using variable nodes with very high degree, which provides a less complex implementation. Also it exploits the advantage of using variable nodes of degree 1, which are very useful for designing LDPC codes at low channel coding rate and low SNR [26]. It is important to recall that in the case of SET-LDPC codes the minimum usable variable node degree is 2.

A graph ensemble is specified through two multivariable polynomials, one associated with variable nodes and the other associated with check nodes. We denote these multivariable polynomials by

$$\nu(\mathbf{r}, \mathbf{x}) = \sum \nu_{\mathbf{bd}} \mathbf{r}^{\mathbf{b}} \mathbf{x}^{\mathbf{d}}, \quad \mu(\mathbf{x}) = \sum \mu_{\mathbf{d}} \mathbf{x}^{\mathbf{d}}, \tag{C1}$$

respectively, where in Eq. (C1) we define the vectors $\mathbf{b}$, $\mathbf{d}$, $\mathbf{r}$, $\mathbf{x}$ and the coefficients $\nu_{\mathbf{bd}}$ and $\mu_{\mathbf{d}}$ as follows. Let n_e denote the number of edge types and n_r denote the number of different channels over which the code-word bits can be transmitted. To represent the structure of the graph, we introduce the following *node-perspective* multivariable-polynomial representation. We thereby interpret degrees as exponents. Let $\mathbf{d} := (d_1, \ldots, d_{n_e})$ be a multiedge degree, and let $\mathbf{x} := (x_1, \ldots, x_{n_e})$

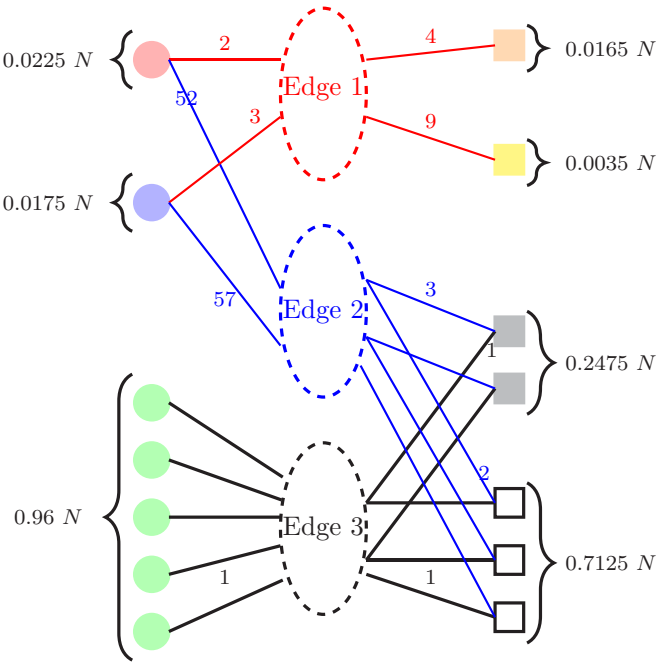

FIG. 7. Graphical representation of the three-edge-type LDPC code presented in Table II, where circles represent the variable nodes and squares represent the check nodes. The numbers of nodes for different edge types are shown as fractions of the code length N, where N is the number of transmitted code-word bits.

denote (vector) variables. We write $\mathbf{x}^{\mathbf{d}}$ for $\prod_{i=1}^{n_e} x_i^{d_i}$. Similarly, let $\mathbf{b} := (b_0, \ldots, b_{n_r})$ be a received degree, and let $\mathbf{r} := (r_0, \ldots, r_{n_r})$ denote variables corresponding to received distributions. By $\mathbf{r}^{\mathbf{b}}$ we mean $\prod_{i=0}^{n_r} r_i^{b_i}$. In this paper we use r_1 for the transmission channel and r_0 for punctured bits (with no transmission channel). Typically, vectors $\mathbf{b}$ will have one entry set to 1 and the rest set to 0. Finally, the coefficients $\nu_{\mathbf{bd}}$ and $\mu_{\mathbf{d}}$ are non-negative real values corresponding to the fraction of variable nodes of type (**bd**) and the fraction of constraint nodes of type **d** in the graph.

For example, let N be the length of the code word; then for each constraint node degree type **d** the quantity $\mu_{\mathbf{d}}N$ is the number of constraint nodes of type **d** in the graph. Similarly, the quantity $\nu_{\mathbf{bd}}N$ is the number of variable nodes of type (**bd**) in the graph. We store this information in a table to describe the structure of the graph. For instance, a full description of a MET-LDPC code ensemble with channel coding rate 0.02 we designed with the following structure is presented in Table II and Fig. 7.

$$\begin{aligned}\nu(\mathbf{r},\mathbf{x}) &= 0.0225\, r_1 x_1^2 x_2^{52} + 0.0175\, r_1 x_1^3 x_2^{57} + 0.96\, r_1 x_3,\\ \mu(\mathbf{x}) &= 0.0165\, x_1^4 + 0.0035\, x_1^9 + 0.2475\, x_2^3 x_3^1\\ &\quad + 0.7125\, x_2^2 x_3^1.\end{aligned}$$

The edge-perspective degree distribution can be described as a vector of multivariable polynomials, for variable nodes and check nodes, respectively,

$$\begin{aligned}\lambda(\mathbf{r},\mathbf{x}) &= \left(\frac{\nu_{x_1}(\mathbf{r},\mathbf{x})}{\nu_{x_1}(\mathbb{1},\mathbb{1})}, \frac{\nu_{x_2}(\mathbf{r},\mathbf{x})}{\nu_{x_2}(\mathbb{1},\mathbb{1})}, \ldots, \frac{\nu_{x_{n_e}}(\mathbf{r},\mathbf{x})}{\nu_{x_{n_e}}(\mathbb{1},\mathbb{1})}\right),\\ \rho(\mathbf{x}) &= \left(\frac{\mu_{x_1}(\mathbf{x})}{\mu_{x_1}(\mathbb{1})}, \frac{\mu_{x_2}(\mathbf{x})}{\mu_{x_2}(\mathbb{1})}, \ldots, \frac{\mu_{x_{n_e}}(\mathbf{x})}{\mu_{x_{n_e}}(\mathbb{1})}\right),\end{aligned} \tag{C2}$$

where

$$\begin{aligned}\nu_{x_i}(\mathbf{r},\mathbf{x}) &= \frac{\partial}{\partial x_i}\nu(\mathbf{r},\mathbf{x}),\\ \mu_{x_i}(\mathbf{x}) &= \frac{\partial}{\partial x_i}\mu(\mathbf{x}),\end{aligned}$$

and $\mathbb{1}$ denotes a vector of all 1s where the length is being determined by context. The coefficients of ν and μ are constrained to ensure that the number of sockets of each type is the same on both sides (variable and check) of the graph. This gives rise to n_e linear conditions on the coefficients of ν

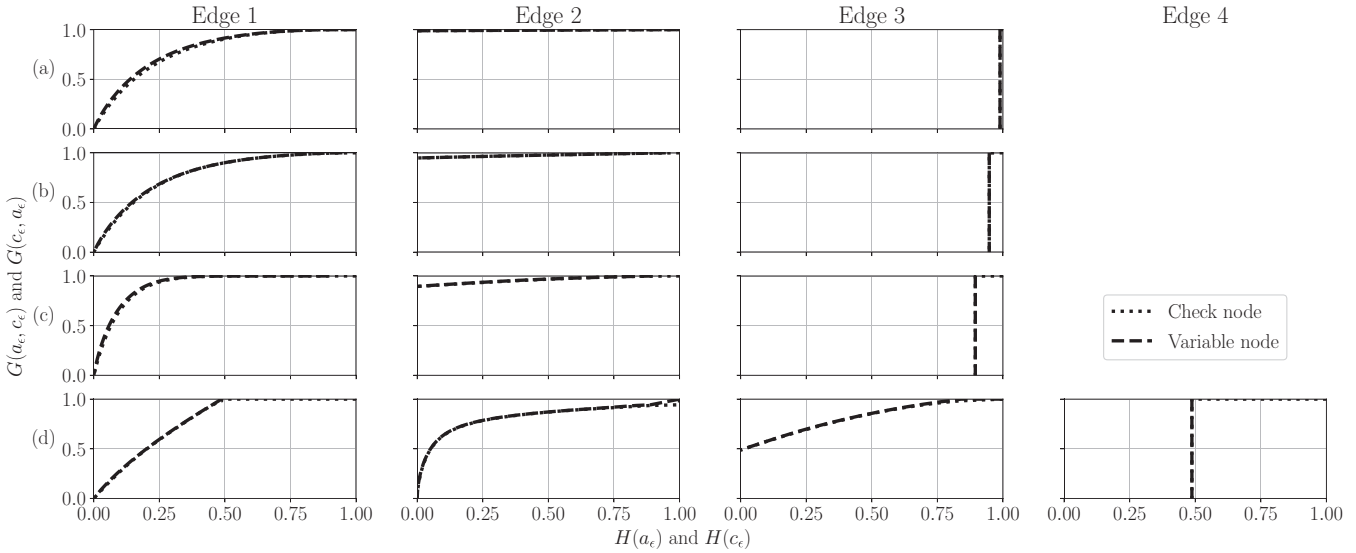

FIG. 8. The G-EXIT charts for separate edge types for the MET-LDPC codes with channel coding rates (a) 0.01, (b) 0.05, (c) 0.10, and (d) 0.50. When the code converges at a specific threshold value, we are able to plot the G-EXIT charts separately by applying the G-EXIT operators for densities at each edge type.

and μ,

$$\nu_{x_i}(\mathbb{1}, \mathbb{1}) = \mu_{x_i}(\mathbb{1}), \quad i = 1, \ldots, n_e.$$

Finally, the nominal channel coding rate for nonpunctured code is given by

$$R^{ch} = \nu(\mathbb{1}, \mathbb{1}) - \mu(\mathbb{1}).$$

2. G-EXIT charts for other codes

The graphical representation of the convergence behavior of the other codes with channel coding rates 0.01, 0.05, 0.10, and 0.50 described in Table I is presented in Fig. 8. The graphs show that the G-EXIT charts can be plotted for a variety of codes from very low channel coding rates of 0.01 to codes with channel coding rate of 0.50. Also we are not limited to codes with three edge types. For example, Fig. 8(d) shows a MET-LDPC code with $n_e = 4$.

[1] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).
[2] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, Rev. Mod. Phys. **81**, 1301 (2009).
[3] E. Diamanti and A. Leverrier, Entropy **17**, 6072 (2015).
[4] F. Laudenbach, C. Pacher, C.-H. F. Fung, A. Poppe, M. Peev, B. Schrenk, M. Hentschel, P. Walther, and H. Hübel, Adv. Quantum Technol. **1**, 1800011 (2018).
[5] P. Jouguet, S. Kunz-Jacques, and A. Leverrier, Phys. Rev. A **84**, 062317 (2011).
[6] D. Huang, P. Huang, D. Lin, and G. Zeng, Sci. Rep. **6**, 19201 (2016).
[7] C. Pacher, J. Martinez-Mateo, J. Duhme, T. Gehring, and F. Furrer, arXiv:1602.09140.
[8] A. Leverrier, R. Alléaume, J. Boutros, G. Zémor, and P. Grangier, Phys. Rev. A **77**, 042325 (2008).
[9] P. Jouguet, D. Elkouss, and S. Kunz-Jacques, Phys. Rev. A **90**, 042329 (2014).
[10] M. Bloch, A. Thangaraj, S. W. McLaughlin, and J. M. Merolla, in *Proceedings of the 2006 IEEE Information Theory Workshop – ITW '06 Punta del Este* (IEEE, Piscataway, NJ, 2006), pp. 116–120.
[11] M. C. Davey and D. MacKay, IEEE Commun. Lett. **2**, 165 (1998).
[12] X. Wang, Y.-C. Zhang, Z. Li, B. Xu, S. Yu, and H. Guo, arXiv:1703.04916.
[13] D. Lin, D. Huang, P. Huang, J. Peng, and G. Zeng, Int. J. Quantum Inf. **13**, 1550010 (2015).
[14] X. Wang, Y. Zhang, S. Yu, and H. Guo, Sci. Rep. **8**, 10543 (2018).
[15] X.-Q. Jiang, S. Yang, P. Huang, and G. Zeng, IEEE Photon. J. **10**, 1 (2018).
[16] D. Slepian and J. Wolf, IEEE Trans. Inf. Theory **19**, 471 (1973).
[17] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, IEEE Trans. Inf. Theory **47**, 619 (2001).
[18] J. Cardinal and G. V. Assche, in *Proceedings of the 2003 IEEE Information Theory Workshop* (IEEE, Piscataway, NJ, 2003), pp. 135–138.
[19] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, IEEE Trans. Inf. Theory **45**, 1361 (1999).
[20] P. Jouguet, S. Kunz-Jacques, E. Diamanti, and A. Leverrier, Phys. Rev. A **86**, 032309 (2012).
[21] Y. Zhang, Z. Chen, S. Pirandola, X. Wang, C. Zhou, B. Chu, Y. Zhao, B. Xu, S. Yu, and H. Guo, Phys. Rev. Lett. **125**, 010502 (2020).
[22] S. Jayasooriya, M. Shirvanimoghaddam, L. Ong, G. Lechner, and S. J. Johnson, IEEE Trans. Commun., **64**, 4044 (2016).
[23] H. Saeedi and A. H. Banihashemi, IEEE Trans. Commun. **58**, 1376 (2010).
[24] V. Rathi and R. Urbanke, IEEE Proc. Commun. **152**, 1069 (2005).
[25] C. Measson, A. Montanari, T. J. Richardson, and R. Urbanke, IEEE Trans. Inf. Theory **55**, 4793 (2009).
[26] T. Richardson and R. Urbanke, in *Workshop Honoring Prof. Bob McEliece on His 60th Birthday* (California Institute of Technology, Pasadena, 2002), pp. 24–26.
[27] S. Jayasooriya, M. Shirvanimoghaddam, L. Ong, and S. J. Johnson, IET Commun. **11**, 61 (2017).
[28] S. Jeong and J. Ha, IEEE Trans. Commun. **67**, 6652 (2019).
[29] B. Ömer, PEG Software for Quasi-cyclic LDPC Codes, AIT Austrian Institute of Technology, Vienna, 2020, http://Bernhard.oemer@ait.ac.at.
[30] A. Leverrier, F. Grosshans, and P. Grangier, Phys. Rev. A **81**, 062343 (2010).
[31] T. Richardson and R. Urbanke, *Modern Coding Theory* (Cambridge University Press, Cambridge, 2008).