

Security-rate trade-off in quantum key distribution: From individual links to large-scale networks

Stefano Pirandola^{1,*}, Cillian Harney,² and Gaetana Spedalieri²¹*Department of Computer Science, University of York, York YO10 5GH, United Kingdom*²*nodeQ, 71–75 Shelton Street, Covent Garden, London WC2H 9JQ, United Kingdom*

(Received 31 December 2025; accepted 23 June 2026; published 24 August 2026)

We explore the interplay between security and key rate of a quantum key distribution (QKD) protocol. This fundamental trade-off is investigated both at the link layer, i.e., for a QKD protocol implemented on a channel, and at the network layer, i.e., within the context of a multiuser multihop communication network. In the former scenario, we consider both discrete- and continuous-variable protocols in the composable finite-size framework, showing how their epsilon security restricts the key rate achievable on a single fiber link under various assumptions for the process of parameter estimation. In the latter scenario, we analyze the impact of composable security in the network setting, studying the interplay between end-to-end (ETE) epsilon security and ETE key rate for arbitrarily chosen remote end points. Starting from basic topologies to more complex ones, such as perturbed lattices, we analyze this trade-off under different types of routing strategies. In multipath routing, a greater number of paths can improve key rates, but this enhancement is accompanied by a reduction in the ETE security level. In the setting of single-path routing, we introduce a modified Dijkstra algorithm, which we call “lambda protocol,” able to optimize route construction depending on the joint ETE demand for security and rate. Our results are first derived for specific network instances (i.e., fixed topologies within a given network class) and subsequently extended to ensemble averages over entire network classes, where phase transitions are revealed with respect to node connectivity.

DOI: [10.1103/7839-3xhy](https://doi.org/10.1103/7839-3xhy)

I. INTRODUCTION

It is known that the performance of quantum key distribution (QKD) [1–4] is limited by a fundamental rate-loss scaling, exactly identified by the Pirandola-Laurenza-Ottaviani-Banchi (PLOB) repeaterless bound [5]. More precisely, the PLOB bound represents the secret-key capacity of the bosonic lossy channel, i.e., the maximum amount of secret bits per channel use that can be distributed over a lossy communication line, such as an optical fiber: a result later extended to free-space channel [6] and a satellite connection [7]. It is known that the introduction of intermediate relays or repeaters (e.g., trusted nodes, twin-field nodes [8], and other types of nodes [9–12]) can overcome this bound and extend the distance of QKD. For this reason, it is crucial to develop multihop QKD networks, where the distance limitation is mitigated by a more gentle rate-loss scaling that depends not only on the number of repeaters in a chain between two end points but also on the type of routing considered. See Ref. [13] for the maximum QKD rates (capacities) that are achievable for end-to-end (ETE) communications in an arbitrary network topology.

Besides the rate-distance trade-off, another fundamental trade-off is between rate and security. This appears naturally in the context of composable secure QKD, where secret key

rates are guaranteed only up to an error probability, called “epsilon security” ε . The latter represents the overall probability of failure of the QKD protocol due to practical imperfections affecting parameter estimation (PE), error correction (EC), and privacy amplification (PA). Each of these processes has an associated error probability: ε_{PE} for PE, the “correctness” ε_{cor} for EC, and the “secrecy” ε_{sec} for PA. The epsilon security is the sum of these values $\varepsilon \simeq \varepsilon_{\text{PE}} + \varepsilon_{\text{cor}} + \varepsilon_{\text{sec}}$ and bounds the overall probability that Alice’s and Bob’s keys fail to be equal and/or private. It is widely accepted as the overall security level of the protocol.

The influence of epsilon security on the rate of a QKD protocol is widely acknowledged within the community. However, a clear and comprehensive analysis of its impact on both individual links and larger networks remains absent. In this work, we investigate these issues in a range of scenarios, encompassing discrete-variable (DV) and continuous-variable (CV) protocols, from single fiber links to multihop communication networks.

For a QKD protocol implemented over a single link, we begin by analyzing the direct impact of epsilon security on the key rate, stemming from the composable terms that appear in the standard block-based QKD rate expressions. We then extend this analysis to session-based QKD, where multiple blocks are exchanged over a stable channel and parameter estimation is performed on the entire set of session points. This extension highlights the trade-off between the increased key rate enabled by improved estimators and the reduced security resulting from the accumulation of errors across multiple blocks, demonstrating that data processing plays a nontrivial role in shaping the balance between rate and security.

*Contact author: stefano.pirandola@york.ac.uk

TABLE I. Composable security can be expressed by the epsilon security ε or the equivalent security level Σ , which extracts the information about the exponent.

Security	ε	Σ
Low	10^{-5}	5
Medium	10^{-10}	10
High	10^{-15}	15

Most interestingly, we analyze the impact of composable security in QKD networks, from basic topologies to more complex ones, such as Waxman random meshes and perturbed lattices. In particular, we investigate how the choice of the routing strategy can heavily shift the balance between ETE security level and rate performance. In the setting of multipath routing, it is crucial to evaluate the benefit of a generally increased key rate versus a reduced security level induced by the epsilon composition over the greater number of links associated with the paths. In the context of single-path routing, the trade-off is played at the level of individual path construction, which may be optimized over a joint security-rate cost function. For this purpose, we introduce a suitably modified Dijkstra algorithm, where a lambda parameter can be set to continuously shift the priority between security and rate. The fundamental trade-off between security and rate is first illustrated through examples of network topologies, and then analyzed across entire classes of networks using ensemble averaging. Phase transitions are examined in relation to node connectivity.

The paper is organized as follows. In Sec. II, we discuss the security-rate trade-off for QKD protocols over a single link. In Sec. III, we analyze the trade-off in the context of basic QKD networks, extending the investigation to more complex topologies in Sec. IV. In Sec. V, we extend our analysis to authentication. Section VI is for conclusions.

II. SECURITY-RATE TRADE-OFF IN QKD LINKS

As already discussed, the composable security of QKD protocols is based on the definition of an ε parameter. The value of this parameter must be very low, typically on the order of 10^{-10} . The most significant information about this parameter is its order of magnitude or exponent. For this reason, and for the sake of simplifying our treatment in the next sections, we introduce the notion of “security exponent,” or simply “security level,” as given by the (modulus of the) exponent of ε according to

$$\Sigma = -\log_{10} \varepsilon. \quad (1)$$

We have the simple mapping of values shown in Table I.

A. DV-QKD link

Let us consider one-photon BB84 with composable security. First, we describe the specific features of this protocol and then analyze the associated security-rate trade-off. We consider a symmetric protocol where both bases Z and X are used for key generation (KG) and PE. The bases are chosen with probabilities q_z and q_x , so the sifting probability

is $p_{\text{sift}} = q_z^2 + q_x^2$. They are used for PE according to a ratio r_{PE} so that a percentage r_{PE} of the states are sacrificed, while $1 - r_{\text{PE}}$ are used for KG (see Appendix A for a more detailed, step-by-step, description of the protocol).

We assume that Alice has a single-photon source, sending N states with a repetition rate r_s . We then consider Bob with a setup efficiency η_{eff} and equipped with a detector with dark count probability p_{dc} , misalignment error e_{det} , and dead time τ_{dt} . Alice and Bob are separated by a channel with transmissivity η_{ch} . In a fiber, this is related to the distance d (in km) and the loss rate α_L (typically, 0.2 dB/km) by the relation $\eta_{\text{ch}} = 10^{-\alpha_L d/10}$.

In this overall setup, Bob’s detection probability is

$$Q_1 = \eta + (1 - \eta)p_{\text{dc}}, \quad (2)$$

where $\eta := \eta_{\text{eff}}\eta_{\text{ch}}$. Given N initial states, the number of sifted states is

$$M = p_{\text{sift}}Q_1N, \quad (3)$$

whose $m = r_{\text{PE}}M$ are for PE and $n = M - m$ for KG. Note that the raw key rate $\kappa := n/N$ is equal to

$$\kappa = (1 - r_{\text{PE}})p_{\text{sift}}Q_1. \quad (4)$$

The quantum bit error rate (QBER) of the channel has an expected value

$$E^* = \frac{p_{\text{dc}}/2 + e_{\text{det}}\eta}{\eta + p_{\text{dc}}}. \quad (5)$$

In practice, Alice and Bob will build an unbiased estimator $\hat{E} \simeq E^*$ and the worst-case value

$$\bar{E} = \hat{E} + \sqrt{(2/m)\ln[(m+1)/\varepsilon_{\text{PE}}]}, \quad (6)$$

the latter bounding E^* up to an error probability ε_{PE} .

Starting from $\bar{E}$, Alice and Bob choose the level of correction needed for their EC step. For example, assuming an EC strategy based on low-density parity-check (LDPC) codes, they will select a suitable code rate R_{code} able to achieve a good probability of success p_{EC} and a good reconciliation efficiency

$$f_{\text{EC}} = \frac{1 - R_{\text{code}}}{h(\bar{E})}, \quad (7)$$

with $h(\cdot)$ being the binary Shannon entropy [14]. During the correction step, Bob will transmit a syndrome to Alice, leaking $nf_{\text{EC}}h(\bar{E})$ bits of information. Then, during the following verification step, Bob will send Alice a hash of length $\lceil \log_2(1/\varepsilon_{\text{cor}}) \rceil$ for some agreed value of ε_{cor} . If Bob’s hash agrees with Alice’s, then EC is successful, with associated probability p_{EC} . The “correctness” parameter ε_{cor} bounds the joint probability that EC is successful and the strings are different. Overall, the process of EC involves the following leakage of bits:

$$\text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}} = nf_{\text{EC}}h(\bar{E}) + \lceil \log_2(1/\varepsilon_{\text{cor}}) \rceil. \quad (8)$$

An important consideration must be made for the two EC-related parameters p_{EC} and f_{EC} . Their exact values are only known when the EC process is performed in practice. Once their values are established for some value of $\bar{\varepsilon}_{\text{cor}}$, these values are still valid for any $\varepsilon_{\text{cor}} \geq \bar{\varepsilon}_{\text{cor}}$. Increasing the value of the

correctness corresponds to verifications with shorter hashes, meaning that the EC success probability p_{EC} can only increase (so keeping p_{EC} unchanged corresponds to imposing a lower bound on the actual performance).

After EC, Alice and Bob perform (PA), compressing their corrected and verified raw keys (of length n) into the final secret key of length $l = rn$, where r is the secret fraction. The error associated with PA, the “epsilon secrecy” ε_{sec} , represents how distant Alice and Bob are from an ideal key completely decoupled from the eavesdropper. As shown in Appendix A, we can write

$$r = 1 - h(\tilde{E}) + \frac{2 \log_2(\varepsilon_{\text{sec}}) - \text{leak}_{\text{EC}}^{\text{cor}}}{n} - \frac{7 \sqrt{\log_2(4/\varepsilon_{\text{sec}})}}{\sqrt{n}}. \quad (9)$$

In terms of bits/use, the secret key rate R is given by $R = \kappa r$. In terms of bits per second, we then have [15]

$$R_{\text{block}} = r_s c_{\text{dt}} R, \quad (10)$$

where c_{dt} is the dead time rate [Eq. (10) of Ref. [16]]

$$c_{\text{dt}} = \frac{1}{1 + (r_s \tau_{\text{dt}} - 1) Q_1}. \quad (11)$$

Overall, the protocol succeeds with probability p_{EC} and is secure up to an epsilon security [17]

$$\varepsilon_{\text{block}} = \varepsilon_{\text{PE}} + \varepsilon_{\text{cor}} + \varepsilon_{\text{sec}}. \quad (12)$$

Let us assume that n_{bks} blocks are exchanged, each of size N . Also, assume that the channel is stable, so the statistics are maintained block by block. We can consider a session with an average key rate [15]

$$R_{\text{ses}} = p_{\text{EC}} R_{\text{block}} \quad (13)$$

and total epsilon security [18]

$$\varepsilon_{\text{ses}} = n_{\text{bks}} \varepsilon_{\text{block}}. \quad (14)$$

We can now show how the security-rate trade-off is crucial in DV-QKD. Our first analysis involves the fundamental trade-off between the block rate R_{block} and the block epsilon security $\varepsilon_{\text{block}}$. More precisely, we set

$$\varepsilon_{\text{PE}} = \varepsilon_{\text{cor}} = \varepsilon_{\text{sec}} = \varepsilon_{\text{block}}/3 \quad (15)$$

and adopt Σ_{block} following the definition in Eq. (1). We assume realistic protocol parameters that are valid for strong security $\Sigma_{\text{block}} = 20$, and we analyze the behavior of the key rate in Eq. (10) as we decrease Σ_{block} down to low security. As we see in the numerical example of Fig. 1, the key rate R_{block} increases by decreasing Σ_{block} . The effect is highly nontrivial, since the rate becomes positive for $\Sigma_{\text{block}} < 15$ and reaches reasonably high values of 2 kbit/s at low security ($\Sigma_{\text{block}} = 5$).

Our second analysis is about the trade-off between the session rate R_{ses} and the session epsilon security ε_{ses} or, equivalently, the session security level $\Sigma_{\text{ses}} = -\log_{10} \varepsilon_{\text{ses}}$. This trade-off is induced when a different, more powerful approach to PE is considered. When we consider a QKD session of n_{bks} blocks in stable conditions, there are two possibilities for building the worst-case value $\tilde{E}$ of Eq. (6):

(1) From each block, we use m points to derive the value of $\tilde{E}$ for that block. This value will determine the block key

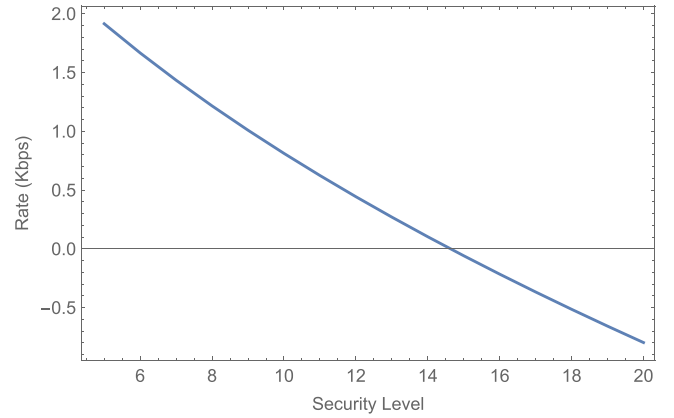


FIG. 1. Trade-off for DV-QKD (BB84 protocol). We plot the block key rate R_{block} (kbit/s) vs the block security level Σ_{block} . By decreasing the security level from 20 down to 5, we gradually increase the key rate, as expected. In this example, high security ($\Sigma_{\text{block}} \geq 15$) does not seem possible. Parameters are $N = 10^8$, $r_s = 10^8$, $q_z = q_x = 0.5$, $e_{\text{det}} = 10^{-3}$, $p_{\text{dc}} = 10^{-5}$, $\tau_{\text{dt}} = 20 \mu\text{s}$, $d = 160 \text{ km}$, $\alpha_L = 0.2 \text{ dB/km}$, $\eta_{\text{eff}} = 0.8$, $r_{\text{PE}} = 0.4$, and $f_{\text{EC}} = 1.22$.

rate R_{block} and, therefore, the amount of PA compression to apply to that block to deliver the final key.

(2) From the entire set of PE points $m' := n_{\text{bks}} m$, we build a refined value of $\tilde{E}$. We use this value uniformly for each block. This means that the process of PA will be delayed until the end of the session, at which point each block undergoes an independent PA compression based on an improved R_{block} . Each block-provided key will then be concatenated to form a session-provided key.

Here, we consider the second approach. As mentioned above, it increases the block rate R_{block} and, therefore, the session rate R_{ses} . The cost to pay is that the session epsilon security will worsen with the number of blocks n_{bks} according to Eq. (14). The entire QKD session becomes a superblock with better key rate but lower security level.

For our numerical investigation, we have chosen the same parameters as in Fig. 1 with the additional value $p_{\text{EC}} = 0.9$. We have then imposed the strong value $\Sigma_{\text{block}} = 15$ for the block security level. In Fig. 2, we show how the worst-case value of the QBER $\tilde{E}$ improves in the number of blocks n_{bks} so that its values decrease to approach the value of the estimator $\hat{E}$. Simultaneously, the session security level Σ_{ses} decreases in n_{bks} from the single-block value of 15 down to 13. More interestingly, in Fig. 3, we show the direct trade-off between the session key rate R_{ses} and the session security level Σ_{ses} . At the level $\Sigma_{\text{ses}} = 15$ (for a single-block session), there is no rate. The rate increases by decreasing the session level (i.e., increasing the number of blocks in the session), with values of the rate rising to 1–2 kbit/s.

B. CV-QKD link

We consider a Gaussian-modulated coherent-state protocol that can be implemented with homodyne [19] or heterodyne [20] detector. See Appendix B for more details about the steps of the protocol and its security.

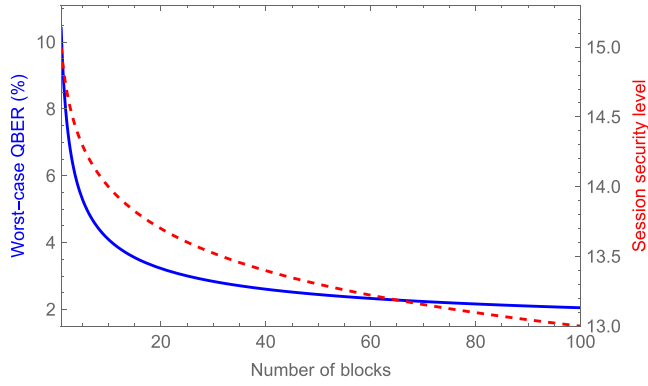


FIG. 2. We plot the worst-case QBER $\tilde{E}$ (in percentage) and the session security level Σ_{ses} as functions of the number of blocks in the session n_{bks} . We set $\Sigma_{\text{block}} = 15$ for the security level of each single block. Parameters are chosen as in Fig. 1 with the additional value $p_{\text{EC}} = 0.9$.

We assume that Alice prepares coherent states and modulates their amplitude according to a bivariate Gaussian distribution with variance σ_x^2 . A block of N states is prepared and sent through the channel with a repetition rate r_s . We then consider Bob equipped with a high-bandwidth detector ($> r_s^{-1}$) with associated noise variance V_0 , which can be either homodyne ($V_0 = 1$) or heterodyne ($V_0 = 2$). Bob's detector has calibrated efficiency η_{eff} and electronic noise v_{el} (considered to be trusted quantities as typical in CV implementations). The two parties are separated by an optical fiber with transmissivity $\eta_{\text{ch}} := T$ (related to the distance d in km by the loss rate $\alpha_L = 0.2$ dB/km) and excess noise ξ .

Alice and Bob agree on a ratio r_{PE} of states to be used for PE, so $m = r_{\text{PE}}N$ are sacrificed while $n = N - m$ proceed to the next steps for KG. Using the m states, they build unbiased estimators, $\hat{T} \simeq T$ and $\hat{\xi} \simeq \xi$, together with worst-case values $\tilde{T}$ and $\tilde{\xi}$, up to a total error probability $2\varepsilon_{\text{PE}}$ (see Appendix B 1 for detailed expressions). The remaining states are subject to analog-to-digital conversion (ADC), so that each of Alice's input values and Bob's output measurements are converted

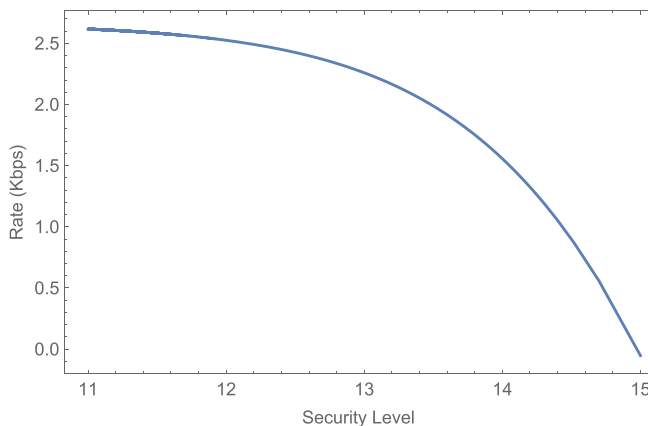


FIG. 3. We plot the session key rate R_{ses} (kbit/s) vs the session security level Σ_{ses} for the BB84 protocol. We set $\Sigma_{\text{block}} = 15$ for the security level of each single block. Parameters are chosen as in Fig. 1 and we set $p_{\text{EC}} = 0.9$.

into binary strings of length V_0D , where D is the digitalization per quadrature (e.g., $D = 7$ bits). After conversion, Bob will use the n KG points to compute an estimator $\hat{H}$ of the Shannon entropy H of its local string, up to an error probability $\varepsilon_{\text{ent}} = \varepsilon_{\text{PE}}$. This computation does not involve any disclosure of information, since it is done completely on Bob's side (see Appendix B 1 for mathematical details).

During EC, Alice and Bob perform the usual steps of correction and verification, with a probability of success p_{EC} and correctness ε_{cor} . The efficiency of the EC process can be quantified by the reconciliation parameter $\beta \leq 1$. Finally, the parties perform PA, compressing their error-corrected and verified strings into a secret key, which is completely detached from an eavesdropper up to an error probability ε_{sec} . Overall, the protocol has epsilon security $\varepsilon_{\text{block}} = \varepsilon_{\text{cor}} + \varepsilon_{\text{sec}} + 3\varepsilon_{\text{PE}}$ and its key rate is given by (see Appendix B 1)

$$R_{\text{block}} \leq \frac{r_s}{N} (nR_m^{\text{PE}} - F), \quad (16)$$

where R_m^{PE} is equal to the asymptotic key rate but computed over estimators and worst-case values, while F accounts for finite-size terms as follows:

$$F = (\sqrt{n} \log_2 n) \sqrt{2 \ln(2/\varepsilon_{\text{PE}})} + 4\sqrt{n} \log_2(\sqrt{2V_0D} + 2) \sqrt{\log_2(8/\varepsilon_{\text{sec}}^2) - \log_2(\varepsilon_{\text{sec}}^2 \varepsilon_{\text{cor}}/2)}. \quad (17)$$

In a typical fiber implementation, channel and setup parameters are relatively stable, and we can consider a CV-QKD session of n_{bks} blocks, each of size N [21]. A fraction p_{EC} of the initial blocks will survive EC. Data processing is done block by block, with the final keys being concatenated into a long session key. As before for DV-QKD, the average session key rate takes the form $R_{\text{ses}} = p_{\text{EC}} R_{\text{block}}$, as in Eq. (13), and the session epsilon security is $\varepsilon_{\text{ses}} = n_{\text{bks}} \varepsilon_{\text{block}}$, as in Eq. (14) [15,18].

Similarly to DV-QKD, we now discuss the security-rate trade-off in CV-QKD, starting from the interplay between the block rate R_{block} and the block epsilon security $\varepsilon_{\text{block}}$. To simplify our analysis, let us assume

$$\varepsilon_{\text{cor}} = \varepsilon_{\text{sec}} = 3\varepsilon_{\text{PE}} = \varepsilon_{\text{block}}/3, \quad (18)$$

and we adopt Σ_{block} following Eq. (1). Assuming parameters that are valid for strong security (here we start from $\Sigma_{\text{block}} = 20$), we then analyze the key rate in Eq. (16) as we decrease Σ_{block} down to lower security levels. The fundamental trade-off is confirmed by the numerical results shown in Fig. 4.

In our second analysis, we consider the trade-off between session rate R_{ses} and session security level $\Sigma_{\text{ses}} = -\log_{10} \varepsilon_{\text{ses}}$, when we adopt a session-wide PE over $m' := n_{\text{bks}}m$ points for computing the estimators and worst-case values of the channel parameters, T and ξ . We expect that the improvement of the session key rate R_{ses} is counterbalanced by the decrease in the session security Σ_{ses} as we increase the number of blocks. This is confirmed by the numerical investigation shown in Fig. 5.

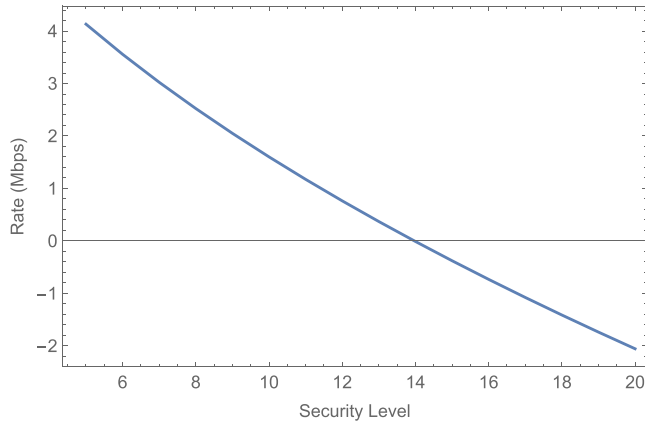


FIG. 4. Trade-off for CV-QKD (homodyne-based protocol). We plot the block key rate R_{block} (Mbit/s) vs the block security level Σ_{block} . By decreasing the security level from 20 to lower values, we see how the rate gradually increases and becomes positive. Parameters are $r_s = 10^8$, $N = 10^7$, $\sigma_x^2 = 40$, $d = 15$ km, $\alpha_L = 0.2$ dB/km, $\xi = 0.01$, $\eta_{\text{eff}} = 0.8$, $v_{\text{el}} = 0.01$, $r_{\text{PE}} = 0.3$, $D = 7$, and $\beta = 0.9$.

III. SECURITY-RATE TRADE-OFF IN QKD NETWORKS

Here, we investigate the interplay between security and rate for an ETE connection (unicast) in a static trusted-node QKD network, i.e., a network with a stable topology, where the properties of the links and the trusted nodes are constant over time. Before we proceed with the analysis, we clarify our notation and assumptions.

A. Notation

In general, a network $\mathcal{N} = (P, E)$ is a pair composed of a set of points P and a set of undirected edges E . To denote an edge or link, one can consider a link-index notation, so a link is identified by a single index such as e . We then define an ETE path as a collection of links between Alice and Bob. This is $\omega = \{e_a, \dots, e_b\}$, where e_a is the link out-bounding from Alice and e_b is the link in-bounding to Bob. Two paths are called “disjoint” if they do not have a link in common,

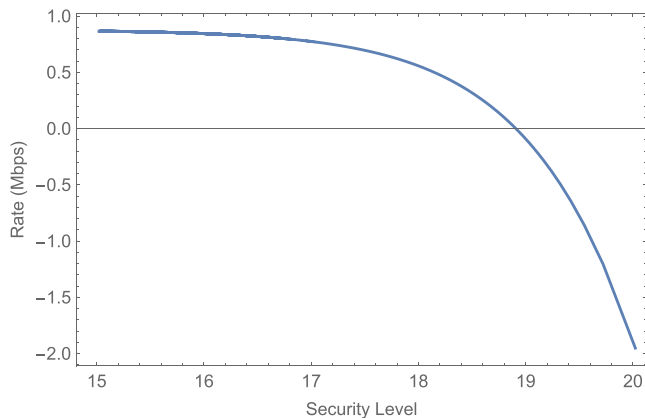


FIG. 5. We plot the session key rate R_{ses} (Mbit/s) vs the session security level Σ_{ses} for the homodyne-based protocol. We set $\Sigma_{\text{block}} = 20$ for the security level of each single block. Parameters as in Fig. 4 and $p_{\text{EC}} = 0.95$.

and we denote by Ω the set of all ETE paths between two end points. Besides the link-index notation just described, we also consider a node-index notation where we represent a link by its two end nodes. Thus, we may write $e = (i, j)$ for a link $e \in E$ that connects nodes $i \in P$ and $j \in P$.

Given a network described as a collection of nodes $P = \{i_k\}_k$ that are interconnected by a collection of edges $E = \{(i_k, j_l)\}_{k,l}$, resulting in a finite, undirected graph $\mathcal{N} = (P, E)$, we consider ETE protocols that employ point-to-point compositably secure QKD along each network link in accordance with some global routing strategy. As a result, a QKD network of this form is primarily characterized according to the single-link rate and ε -security distributions:

$$\mathcal{R} := \{R_{ij}\}_{(i,j) \in E}, \quad \mathcal{E} := \{\varepsilon_{ij}\}_{(i,j) \in E}, \quad (19)$$

where R_{ij} is the secret-key rate of the link $e = (i, j)$ and ε_{ij} is its associated epsilon security. Note that the key rate of the link corresponds to the average session key rate R_{ses} of the protocol that is implemented over that link. Correspondingly, the epsilon security associated with the link corresponds to the session value ε_{ses} .

We define the ETE rate R_E as the total rate per network use achievable by two remote end points. For a specific routing protocol $\mathcal{P}$ used in the network $\mathcal{N}$, we write $R_E(\mathcal{N}|\mathcal{P})$. The scenario considered is that of a trusted-node QKD network. This means that individual and independent QKD sessions are implemented over the links belonging to a route (edge set $\Omega_{\mathcal{P}}$), generating an ensemble of link keys. These link keys are then swapped via one-time pad sessions and composed to create an ETE key for the end points of the route. In conditions of stationary traffic, each link $e = (i, j)$ has a stable key rate R_{ij} and stable epsilon security ε_{ij} , as represented in Eq. (19). Thus, in a scenario of key composition, the ETE rate R_E will be affected by bottleneck rates along the route, while the ETE epsilon security ε_E will increase with the number of links in the route. See Appendix C for more details about the calculation of the ETE rate starting from the link rates.

If a single link is ε_{ij} secure, this means that there is at most a ε_{ij} probability that information will be leaked to an adversary. In this context, given the security distribution of a network $\{\varepsilon_{ij}\}_{(i,j) \in E}$ and the ETE protocol $\mathcal{P}$ that makes use of an edge set $\Omega_{\mathcal{P}} = \{(i, j)_k \in E\}_k$, we can define the ETE epsilon security ε_E affecting the overall key distribution between the two remote end points. This is provided by the compositions of the epsilons along the edge set and given by

$$\varepsilon_E(\mathcal{N}|\mathcal{P}) := 1 - \prod_{(i,j) \in \Omega_{\mathcal{P}}} (1 - \varepsilon_{ij}) \simeq \sum_{(i,j) \in \Omega_{\mathcal{P}}} \varepsilon_{ij}. \quad (20)$$

Analogously, we can compute the ETE security level as

$$\Sigma_E(\mathcal{N}|\mathcal{P}) := -\log_{10} [\varepsilon_E(\mathcal{N}|\mathcal{P})]. \quad (21)$$

In our investigation, the total number of links satisfies $|E| \ll \varepsilon^{-1}$, so composability can be easily treated mathematically and the ETE communications are guaranteed to have sufficiently low ε securities. Then, for simplicity, we reduce the scenario to a single epsilon value for the various links. This “uniform link security” occurs when the epsilon security is fixed as an input parameter for all the QKD protocols on the

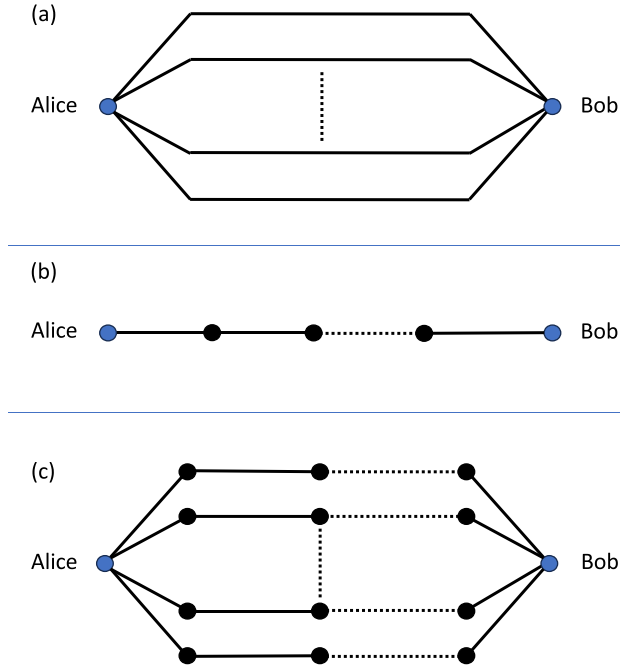


FIG. 6. Basic topologies with vertical and horizontal depths. We depict a multilink in panel (a) displaying vertical depth, a single repeater chain in panel (b) displaying horizontal depth, and a parallel mesh in panel (c) displaying both depths. All these configurations connect two end points, Alice and Bob, whose aim is to distribute a key at the ETE rate R_E with ETE epsilon security ε_E .

various links. More generally, this condition can be enforced as a “uniformly bounded link security,” meaning that the actual epsilon securities of the various links are actually different, but they can be upper bounded by a maximum acceptable value; i.e., we can set $\varepsilon_e \leq \varepsilon$ for any e . Any link that violates this bound is deemed insecure and removed from the topology. Treating the upper bound ε as the effective common security parameter is then fully consistent with standard QKD security analysis, which adopts a worst-case perspective.

B. Basic topologies

We start by considering simple network topologies: a multilink with vertical depth M (number of parallel links) and a single repeater chain with horizontal depth N (number of sequential repeaters). We then gradually extend the topology in both dimensions, addressing meshes of increasing complexity.

1. Multilink

Consider Alice and Bob connected by a multilink of depth M as in Fig. 6(a). The generic link e is a channel with rate R_e and epsilon security ε , i.e., rate-security pair $\{R_e, \varepsilon\}$. Let us call p the number of simultaneous paths they access, with $p = 1$ corresponding to single path and $p = M$ to flooding [22–24]. The best single-path strategy is given by the widest path between the two end points, which is just the link with the highest rate, so we have $R_E = \max_e R_e$ and $\varepsilon_E = \varepsilon$. In the case of p -path routing, we need to find the p links with higher rates and sum them up. In other words, we introduce an ordering $\{R_{e_1}, \dots, R_{e_M}\}$ with $R_{e_k} \geq R_{e_{k+1}}$ and then we select the first p

entries, so we have

$$R_E = \sum_{k=1}^p R_{e_k}, \quad \varepsilon_E = p\varepsilon. \quad (22)$$

Note that, for a uniform multilink, where the channels have all the same rate, the scenario becomes very simple. In this case, there is no difference in ETE rate and security if Alice and Bob use single-path routing. No matter how they choose their path, this will be a single link with the same properties. For multipath routing, we have the simple additive formulas

$$R_E = pR, \quad \varepsilon_E = p\varepsilon. \quad (23)$$

If we increase p , we certainly increase the ETE rate (better performance), but we proportionally increase the ETE epsilon parameter (lower security). Multipath routing naturally introduces a rate-security trade-off [25,26].

2. Repeater chain

Consider now a single repeater chain, where Alice and Bob are connected by N middle nodes, and the $N + 1$ intermediate links have rate-security pairs $\{R_e, \varepsilon\}$ [see Fig. 6(b)]. In this case, the ETE rate is simply given by the lowest rate in the chain (the bottleneck) while the epsilon parameters sum up. In other words, we have

$$R_E = \min_e R_e, \quad \varepsilon_E = (N + 1)\varepsilon. \quad (24)$$

The ETE rate is equal to the bottleneck rate, but the epsilon parameter grows with the depth of the chain.

3. Parallel mesh

The previous analyses can be easily extended to a parallel mesh given by M parallel N -repeater-long disjoint paths [see Fig. 6(c)]. Let us call ω a generic path between Alice and Bob. We denote by $e \in \omega$ a link belonging to path ω . The generic link e has rate-security pair $\{R_e, \varepsilon\}$.

A path has an associated rate

$$R_E(\omega) = \min_{e \in \omega} R_e \quad (25)$$

and epsilon security $(N + 1)\varepsilon$. If we consider p paths $\Omega_p = \{\omega_1, \dots, \omega_p\}$ between Alice and Bob, then the associated ETE rate is equal to

$$R_E(\Omega_p) = \sum_{\omega \in \Omega_p} R_E(\omega) \quad (26)$$

and the associated epsilon security is simply $p(N + 1)\varepsilon$. To get the best performance, we introduce an ordering $\{\omega_1, \dots, \omega_M\}$ such that $R_E(\omega_j) \geq R_E(\omega_{j+1})$. Then, we select Ω_p considering the first p paths from this ordering.

Consider the simple case of a parallel mesh with uniform link rate, i.e., $R_e = R$ for all e . For a multipath routing strategy with p paths, the previous formulas simplify to the following expressions:

$$R_E = pR, \quad \varepsilon_E = p(N + 1)\varepsilon. \quad (27)$$

We easily see the joint effect of what was discussed for the previous two topologies: the ETE rate is increased by the number of paths p , but the ETE epsilon is increased by both p

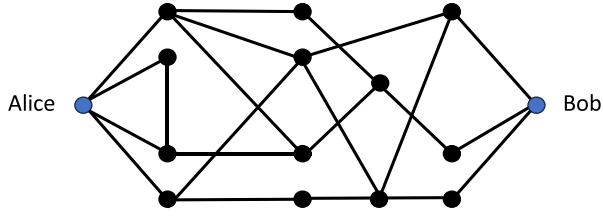


FIG. 7. Example of generic mesh between Alice and Bob.

and N . Note that we can also write

$$\frac{\varepsilon_E}{R_E} = (N + 1) \frac{\varepsilon}{R}. \quad (28)$$

Because the ratio $\frac{\varepsilon_E}{R_E}$ should be kept as low as possible, we see that the horizontal depth of a mesh is always deleterious for the security level, for any value of p [27].

4. Generic mesh

For a generic mesh between Alice and Bob (see Fig. 7 for an example), the computation of the optimal ETE rate R_E follows the specific p -path routing strategy adopted. Given a routing algorithm that identifies p disjoint paths $\Omega_p = \{\omega_1, \dots, \omega_p\}$, each path ω with associated rate $R_E(\omega)$, the total ETE rate of Alice and Bob $R_E = R_E(\Omega_p)$ is given by the formula in Eq. (26). We then compute the associated link consumption L defined as the total number of links involved in the routing solution. This will give the total ETE epsilon security for Alice-Bob communication $\varepsilon_E = L\varepsilon$. With these tools, we can explore the trade-off between ETE rate and security.

According to our previous analysis, the increase in the ETE rate achieved by resorting to multipath routing is counterbalanced by a simultaneous rise in the ETE epsilon parameter. This rate-security trade-off will be examined in more complex scenarios below. In general, the problem can be formulated as a joint optimization task in which both the ETE rate R_E and the ETE security ε_E are incorporated into a single cost function. In the following section, we will summarize variants of routing algorithms that can achieve these rate/security goals.

C. Route optimization algorithms

In the single-unicast setting, one can establish communication between end users via an ETE routing strategy, $\mathcal{P}$. As discussed, we may consider unique strategies based on the network objective: ETE rate or security optimization. In this context, Dijkstra's algorithm (DA) [28,29] embodies an efficient method for determining single-path routing solutions in line with specific pathwise objective. We may consider rate-optimized single-path routing $\mathcal{P}_{sp}^{R^*}$ or security-optimized single-path routing $\mathcal{P}_{sp}^{\Sigma^*}$. Focusing on protocols that operate at the extremes of these objectives, we can also understand hybrid strategies as interpolations. Finally, as discussed above, multipath routing provides a natural way to trade off the ETE key rate against security.

1. Single-path rate optimization

Given a network $\mathcal{N} = (P, E)$ with edgewise rate distribution $\mathcal{R} = \{R_e\}_{e \in E}$, we define an undirected weighted graph

$G_{\mathcal{R}} = (P, E, \mathcal{R})$. Then, given two endnodes Alice $a \in P$ and Bob $b \in P \setminus \{a\}$, with set of possible paths $\Omega = \{\omega\}$, a rate-optimized single-path DA $\mathcal{P}_{sp}^{R^*}$ locates the optimal path

$$\omega_R^* = \arg \max_{\omega \in \Omega} \min_{e \in \omega} R_e, \quad (29)$$

in run-time $\mathcal{O}(|E| + |P| \log_2 |P|)$.

The algorithm aims to optimize the global widest-path objective $J_{\text{rate}} = \max_{\omega \in \Omega} \min_{e \in \omega} R_e$. To implement this objective, it assigns to each node $u \in P$ a label T_u . This label represents the best bottleneck value currently known for a route from Alice a to u . When a settled node u is used to inspect one of its neighboring nodes v , the candidate bottleneck value of the extended route is $\min\{T_u, R_{uv}\}$. The label of v is then updated only if this candidate improves the best value currently known. This gives the local relaxation rule

$$T_v \leftarrow j_{\text{rate}}(u, v) := \max[T_v, \min(T_u, R_{uv})]. \quad (30)$$

Alternatively, we can use the local update function

$$j_{\text{rate}}(u, v) := \min[T_v, \max(T_u, R_{uv}^{-1})], \quad (31)$$

which is based on the inverse of the link rate. See Algorithm 1 for the node-label update (with $j = j_{\text{rate}}$).

2. Single-path security optimization

One can readily modify DA to achieve ETE security optimization. The most secure route between two nodes will inherently be single path, since there is no incentive to use additional resources to amplify performance. Given a network $\mathcal{N} = (P, E)$ with edgewise security distributions $\varepsilon = \{\varepsilon_e\}_{e \in E}$, we can define the transformed security distribution

$$\tilde{\varepsilon} := \{-\log_{10}(1 - \varepsilon_e) \mid e \in E\} \quad (32)$$

and use it to define an undirected weighted graph $G_{\tilde{\varepsilon}} = (P, E, \tilde{\varepsilon})$. Given a pair of endnodes, a and b , an application of DA that minimizes the weighted sum of edges used in the graph $G_{\tilde{\varepsilon}}$ will identify an ETE route $\omega_{\tilde{\varepsilon}}^*$ with optimal pathwise security in $\mathcal{N}$. More precisely, we consider a security-optimized single-path DA $\mathcal{P}_{sp}^{\Sigma^*}$ that identifies

$$\begin{aligned} \omega_{\tilde{\varepsilon}}^* &= \arg \min_{\omega \in \Omega} \left[- \sum_{e \in \omega} \log_{10}(1 - \varepsilon_e) \right] \\ &= \arg \max_{\omega \in \Omega} \prod_{e \in \omega} (1 - \varepsilon_e) = \arg \min_{\omega \in \Omega} \varepsilon_E(\omega), \end{aligned} \quad (33)$$

where $\varepsilon_E(\omega)$ is the ETE epsilon security associated with the generic path ω , as defined in Eq. (20). Note that, in the specific case of a uniform security (same ε for all links), optimizing the ETE security ε_E becomes equivalent to finding the shortest path on $G_{\tilde{\varepsilon}}$. In the general case, the local relaxation rule of $\mathcal{P}_{sp}^{\Sigma^*}$ is given by

$$T_v \leftarrow j_{\text{sec}}(u, v) := \min[T_v, T_u - \log_{10}(1 - \varepsilon_{uv})] \quad (34)$$

and processed according to Algorithm 1 (with $j = j_{\text{sec}}$).

3. Single-path hybrid strategy

We now construct a hybrid routing protocol that can be tuned to apply more weighting to performance (rate) or security. One way to achieve hybrid rate-security optimization

ALGORITHM 1. Generic Dijkstra-type cost update.

```

// Source node has zero initial cost
 $T_a \leftarrow 0$ 
// All other nodes are initially unreachable
for  $u \in P \setminus \{a\}$  do
     $T_u \leftarrow \infty$ 
end for
// Initialize settled and unsettled node sets
 $A \leftarrow \emptyset$ 
 $B \leftarrow P$ 
// Continue while Bob is unsettled and reachable nodes remain
while  $\{u \in B \mid T_u < \infty\} \neq \emptyset \wedge b \in B$  do
    // Select unsettled node with smallest current cost
     $u \leftarrow \arg \min_{w \in B} \{T_w\}$ 
    // Inspect unsettled neighbours of  $u$ 
    for  $v \in \{v \in B \mid \{u, v\} \in E\}$  do
        // Update the cost of reaching  $v$  through  $u$ 
         $T_v \leftarrow j(u, v)$ 
    end for
    // Mark  $u$  as settled
     $A \leftarrow A \cup \{u\}$ 
    // Remove  $u$  from unsettled set
     $B \leftarrow B \setminus \{u\}$ 
end while
    
```

with DA is to utilize a local update function that includes both rate $\mathcal{R}$ and security ϵ distributions. A possible choice is the following “lambda protocol” $\mathcal{P}_\lambda$:

$$T_v \leftarrow j_\lambda(u, v) := \min \{T_v, \max(T_u, R_{uv}^{-1}) + \lambda[T_u - \log_{10}(1 - \epsilon_{uv})]\}. \quad (35)$$

This update contains two distinct contributions. The first term, $\max(T_u, R_{uv}^{-1})$, is the rate-oriented part of the cost. Since the rate is inverted, high-rate links correspond to small costs [cf. with the minimization in Eq. (31)]. The maximum operation propagates the largest inverse-rate value encountered along the partial route, and therefore plays the same role as a bottleneck-rate update. The second contribution, $T_u - \log_{10}(1 - \epsilon_{uv})$, is the security-oriented edge penalty [cf. Eq. (34)]. It is additive because the ETE composable security is obtained by multiplying the link survival probabilities $(1 - \epsilon_{uv})$ along a route, or equivalently by summing their negative logarithms [cf. Eq. (33)]. Thus, adding this term penalizes routes that accumulate larger security degradation.

The parameter $\lambda \geq 0$ controls the relative weight assigned to this security penalty. For small λ , the update is dominated by the inverse bottleneck-rate term and the algorithm behaves more like a rate-optimized widest-path search. For larger λ , the security penalty becomes more influential and the algorithm increasingly favors routes with lower accumulated composable-security cost.

The outer minimization with the current label T_v is the usual Dijkstra-type relaxation step: The label of v is updated only if the candidate route through u provides a smaller hybrid cost than the best one currently known. Minimizing this local update greedily via a Dijkstra-type procedure therefore

provides a tunable hybrid optimization of rate and security. See Algorithm 1 with $j = j_\lambda$.

4. Multipath routing algorithms

Multipath routing algorithms can be employed to achieve rate optimization. Flooding, $\mathcal{P}_\Pi$, is the ETE routing protocol that yields the greatest performance but is also the one that is least secure, because of the extensive use of network resources. The multiple disjoint path algorithm (MDPAlg) is a modified version of DA that enables efficient, edge-disjoint multipath routing [30]. It was recently modified to a version that optimizes routing by minimizing the inverse accumulated rate (IAR) throughout an ETE route [31]. The IAR-MDPAlg was shown to be particularly suitable for the optimization of quantum communication networks. In this work, we explore the rate-security trade-off for multipath routing in a similar way, denoting such protocol as $\mathcal{P}_{\text{mdp}}^{N_p}$, where N_p is the maximum number of disjoint paths permitted in an ETE routing solution. For more details about IAR-MDPAlg, see Ref. [31].

5. Numerical analysis

As a practical example to study the rate-security trade-off, we consider a random network $\mathcal{W} = (P, E)$ generated according to a Waxman model. This is a random network model for which randomly distributed nodes, i and j , are connected with a probability that decays exponentially with respect to their separation $d(i, j)$. More precisely, links are formed with probability $p_{\text{conn}}(i, j) = \beta e^{-d(i, j)/r_0}$, where $\beta, r_0 \in \mathbb{R}_0^+$ are hyperparameters that define the model. We also introduce a pruning rate $\delta \in \mathbb{R}_0^+$, which defines the minimum acceptable single-link rate $R_{ij} \geq \delta$ between any two network nodes $(i, j) \in E$. For the numerical analysis of this section, we set $\delta = 10^{-6}$ bit/s, so that even connections with extremely low rates are kept as valid links.

In Fig. 8(a), we generate an instance of a Waxman network using a model with $\beta = 1$, $r_0 = 5$ in a square region of side length $L = 25$ km with a CV-QKD homodyne protocol over each link (see Table III in Appendix B for its parameters). This resulted in a network $\mathcal{W}$ with $N = 123$ nodes and nodal density $\rho \approx 0.1$ nodes/km².

In Fig. 8(b), we display ETE rate and security distributions formed by taking 1000 sample end-user pairs and computing ETE routes relative to a number of different routing protocols: rate-optimized $\mathcal{P}_{\text{sp}}^{R^*}$ and security-optimized $\mathcal{P}_{\text{sp}}^{S^*}$ single-path, multiple disjoint path routing $\mathcal{P}_{\text{mdp}}^{N_p=5}$, and flooding $\mathcal{P}_\Pi$. Inspecting these distributions summarizes the rate-security trade-off clearly, showing how resource greedier protocols can acquire better rate distributions at the cost of ETE security.

In Fig. 8(c), we demonstrate the ability to interpolate between rate/security optimization via the lambda protocol $\mathcal{P}_\lambda$ defined by the modified DA using the local cost function in Eq. (35). Here, we plot mean R_E and Σ_E values evaluated over 1000 end-user pair samples. Finally, Fig. 8(d) demonstrates the rate-security trade-off for $\mathcal{P}_{\text{mdp}}^{N_p}$ with respect to path number N_p . Here, we plot both mean value and variance for R_E and Σ_E , by sampling 1000 end-user pairs in the network.

In Appendix D, we provide a further study based on a regular 8×8 square-lattice topology with diagonal

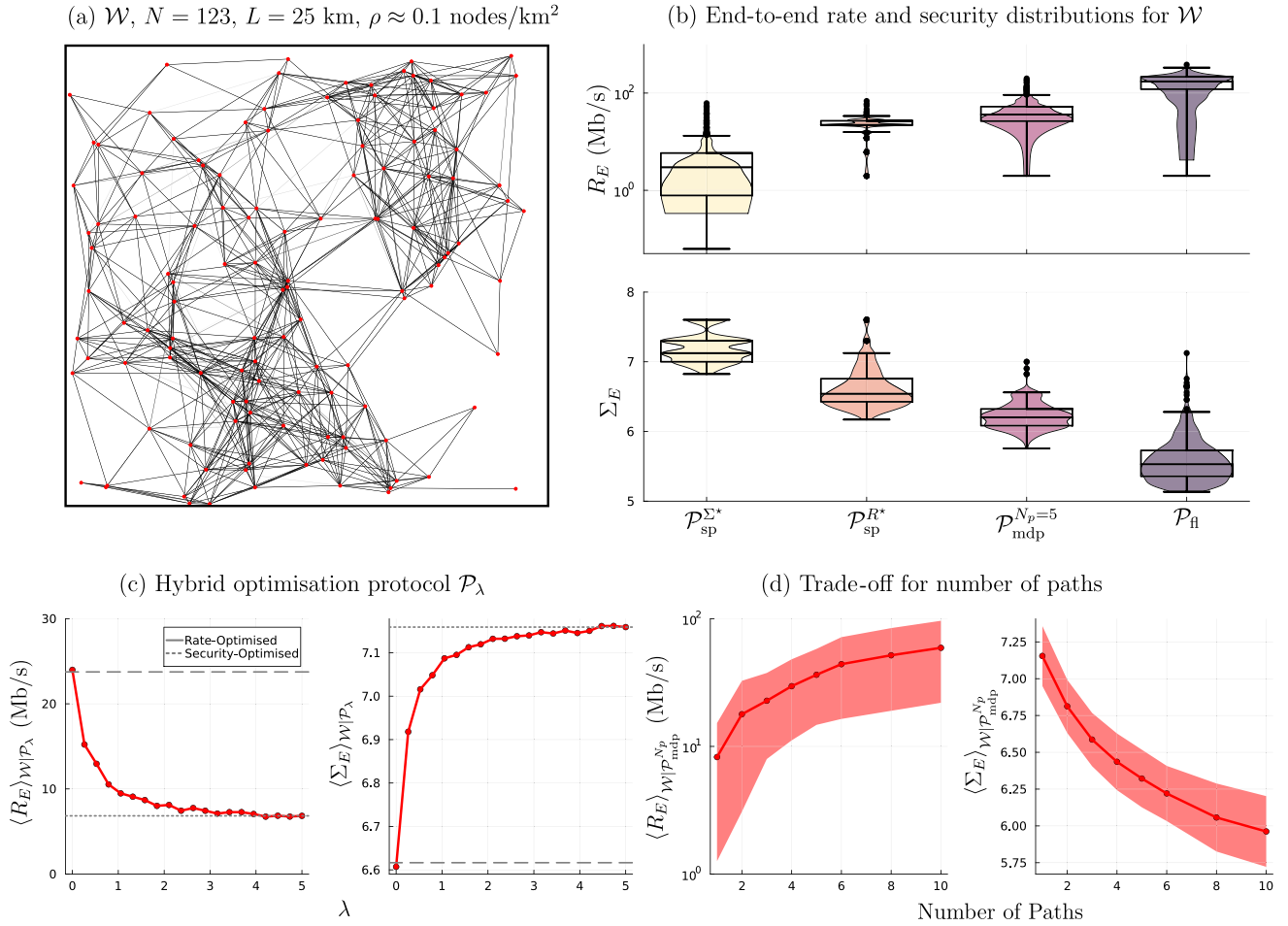


FIG. 8. Rate-security trade-off analyzed for a specific Waxman network $\mathcal{W}$ with a CV-QKD link layer utilizing a homodyne protocol with parameters in Table III of Appendix B (with block-by-block PE). The network topology is visualized in panel (a). Panel (b) illustrates its distributions of ETE rates and security levels, relative to different routing protocols, established via 1000 sampled ETE route computations on the network per protocol: security-optimized single-path $\mathcal{P}_{sp}^{\Sigma^*}$, rate-optimized single-path $\mathcal{P}_{sp}^{R^*}$, five-path edge-disjoint multipath routing $\mathcal{P}_{mdp}^{N_p=5}$, and flooding $\mathcal{P}_{fl}$ (note that these data are represented using box and violin plots). Panel (c) illustrates the average performance/security of the hybrid rate-security routing strategy, the lambda protocol $\mathcal{P}_\lambda$, for different hyperparameter tunings. The horizontal gray lines plot the average rate/security achieved by the standard rate/security optimized single-path protocols, which bound the hybrid method. Finally, in panel (d), we inspect the rate-security trade-off relative to multipath routing for network $\mathcal{W}$, looking at the average performance/security of the N_p -path edge-disjoint routing strategy $\mathcal{P}_{mdp}^{N_p}$, showing the security degradation/rate amplification associated with multipath routing (here we show both average and standard deviation).

diamond-cell connectivity. This is a structure that may resemble the backbone of a metropolitan QKD network.

IV. MORE COMPLEX NETWORKS

In this section, we investigate more complex topologies, revealing network characteristics and benchmarking security versus performance for different traffic scenarios.

A. Spatial networks and generators

Typically, network analyses necessitate the consideration of explicit topologies. Hence, the analysis of quantum/QKD networks is often challenging due to the fact that such networks do not yet exist at scale. Random networks pose a very useful tool for the analysis of future QKD networks,

as they can be used to embody expected features of these next-generation networks. One can define a class of random networks from which a network instance can be sampled.

The inexorable trade-off between rate and distance in quantum communications invokes a natural interest in the study of spatial quantum networks, i.e., networks with topologies that carry spatial information about the positions of nodes, channel lengths, etc. To capture this information, we can extend the notion of a network from $\mathcal{N} = (P, E) \rightarrow (P, E, \phi)$, where ϕ is a nodewise set of spatial positions defined in a d -dimensional network region,

$$\phi := \{\mathbf{x}_i \in \mathbb{R}^d \mid i \in P\}. \quad (36)$$

In our work, we consider networks defined in $d = 2$ -dimensional areas. Combining ϕ with Eq. (19), we arrive at the complete definition of a spatial QKD network that we will

utilize in our work,

$$\mathcal{N} = (P, E, \phi, \mathcal{R}, \epsilon). \quad (37)$$

To provide more rigor with respect to definition of network classes, we introduce the notion of a *spatial network generator*,

$$\mathcal{G} := (\Phi, \Upsilon). \quad (38)$$

This is used to define the generation/sampling of spatial networks according to a procedure defined by a specific network class. In this formalism,

$$\Phi : P \mapsto \phi \in \mathbb{R}^2 \quad (39)$$

defines a spatial embedding function that maps nodes from a node set to coordinates in a two-dimensional area, $\mathbf{i} \rightarrow \phi_i = \mathbf{x}_i = (x_i, y_i) \in \mathbb{R}^2$. Meanwhile, Υ defines a network connection rule set governing the allocation of point-to-point network edges,

$$\Upsilon : P \times \Phi(P) \mapsto (E, \mathcal{R}, \epsilon), \quad (40)$$

such that the interconnection of network nodes is dependent upon both the node set P and their spatial embedding $\Phi(P)$. It is clear that if Υ mediates the connection of network nodes, it is also responsible for the resultant single-link rate and ϵ -security distributions $\mathcal{R}$ and ϵ .

The notion of a spatial network generator is very useful when considering networks that are systematically designed, such as random network or lattice-based models.

B. ETE network analysis

Consider a network created by a generator, $\mathcal{N} \sim \mathcal{G}$, with total number of nodes N , and a specific routing protocol $\mathcal{P}$ implemented over such network. One can define *average* ETE metrics that are computed as statistical averages over possible end-user pairs in the network. Denote the complete set of possible end-user pairs as $\mathcal{I} := \{\mathbf{i}_j\}_j = \{\{\mathbf{a}_j, \mathbf{b}_k\}\}_{j \neq k}$. The network average (or unicast average) of an ETE metric O then takes the form

$$\langle O \rangle_{\mathcal{N}|\mathcal{P}} := \frac{1}{|\mathcal{I}|} \sum_{\mathbf{i} \in \mathcal{I}} O(\mathbf{i}, \mathcal{N}|\mathcal{P}), \quad (41)$$

where $O(\mathbf{i}, \mathcal{N}|\mathcal{P})$ is an ETE metric computed as a function of the routing solution between the end-user pair (unicast) $\mathbf{i} = \{\mathbf{a}, \mathbf{b}\}$ via the routing protocol $\mathcal{P}$. Since $|\mathcal{I}|$ may very quickly grow much too large to consider in totality, we can sample L end-user pairs to approximate the average ETE metric

$$\langle O \rangle_{\mathcal{N}|\mathcal{P}} \approx \frac{1}{L} \sum_{j=1}^L O(\mathbf{i}_j \sim \mathcal{I}, \mathcal{N}|\mathcal{P}), \quad (42)$$

which can be computed along with statistical errors.

To glean general information about a class of network, not just individual network instances, we need to extend our statistical analysis and we may collect ensemble averages. Given a network generator $\mathcal{G}$, and given that any network $\mathcal{N} \sim \mathcal{G}$ is drawn with equal probability, then the *ensemble average* ETE rate is approximated by sampling M networks and using the individual network averages $\{\langle O \rangle_{\mathcal{N}_i|\mathcal{P}}\}_{i=1}^M$ to approximate

a class average. More precisely,

$$\langle O \rangle_{\mathcal{G}|\mathcal{P}} \approx \frac{1}{M} \sum_{i=1}^M \langle O \rangle_{\mathcal{N}_i \sim \mathcal{G}|\mathcal{P}} \quad (43)$$

$$= \frac{1}{LM} \sum_{i=1}^M \sum_{j=1}^L O(\mathbf{i}_j \sim \mathcal{I}, \mathcal{N}_i \sim \mathcal{G}|\mathcal{P}). \quad (44)$$

This approximation can be made sufficiently accurate by taking large enough end-user samples L and network class samples M .

C. Perturbed lattice-based networks

An interesting class that is often considered for assessing network performance is lattices. These are networks with underlying graphical models that obey consistent connectivity patterns, and when embedded into a geometrical space form a regular tiling or tessellation. They are useful due to their ease of description, symmetries, and expedience of analytical modeling. For these reasons, lattice-based topologies are often used as a sandbox for network protocol benchmarking. Recently, the class of weakly regular networks was used to invoke analytical connections between network properties such as connectivity/nodal density and ETE capacities [32]. This model utilized infinite graphs with close to degree-regular connectivity patterns, but did not enforce strict spatial distributions for the positioning of nodes in the graphical structures.

Inspired by this approach, we also consider network topologies with latticelike connectivity structures. We will consider more realistic instances of such topologies via networks embedded in finite spatial areas, and once omitting any strict spatial positioning of network nodes according to the lattice model. We achieve this through a perturbative topology generation procedure.

Let us define the classes of spatial network generator used to model ideal lattice-based networks,

$$\mathcal{G}_l = (\Phi_l, \Upsilon_l), \quad (45)$$

where l labels the lattice type. In this work, we consider spatial networks generated in generally rectangular areas with sides of length L_x, L_y km and area of $A = L_x \times L_y$ km². We define $\mathcal{G}_l$ as *deterministic* network generators, as they are used to construct lattice networks that obey strict connectivity rules and strict spatial positioning.

Core features of ideal lattices are the symmetries and degeneracies embedded within their spatial distributions. In turn, there exist characteristic link lengths that emerge from such symmetries that are consequential for quantum networks, due to the intrinsic relationships between single link lengths and link rates. These characteristic lengths are functions of the lattice class and the number of network nodes (or network density, ρ). For the ideal lattice classes considered in this work, their symmetries invoke a complete degeneracy of link lengths across a network instance resulting in a unique characteristic length $a_l(\rho)$, such that

$$d(\mathbf{i}, \mathbf{j}) = \|\mathbf{x}_i - \mathbf{x}_j\| \geq a_l(\rho), \quad \forall \mathbf{i} \neq \mathbf{j} \in P. \quad (46)$$

Hence, $a_l(\rho)$ is a property of the chosen lattice structure, nodal density ρ , and spatial network region.

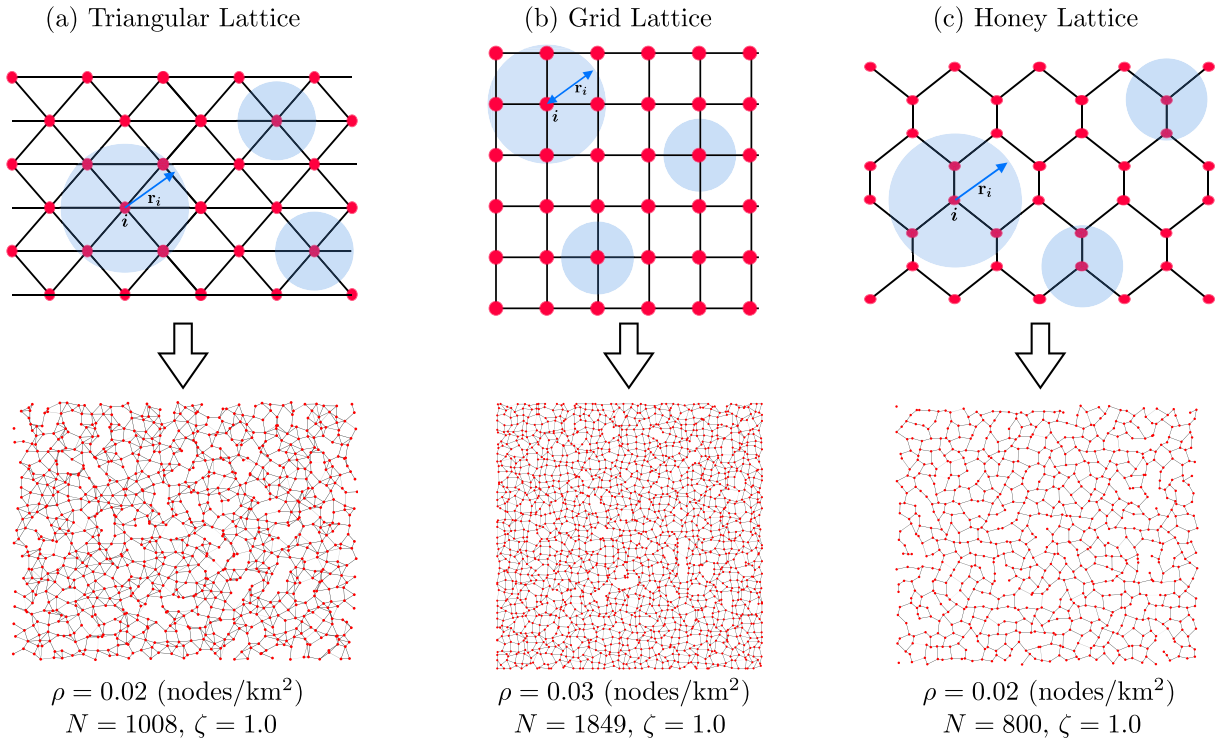


FIG. 9. Lattice network topologies of different classes. The upper panels illustrate the ideal lattice topologies, which can be generalized through the perturbative class described in Sec. IV C to generate more realistic networks via random frustrations $\mathbf{r}_i$ applied to the ideal nodal positions. The lower panels illustrate example, large-scale topologies with nonzero rigidity $\zeta = 1$ for (a) the triangular lattice, (b) the grid lattice, and (c) the honeycomb lattice. Each perturbed network shown is generated with a homodyne CV-QKD link layer according to the parameters in Table III of Appendix B (with block-by-block PE).

In this work, we consider three main lattice classes:

- (1) *Honeycomb*: Resembling a honeycomb tessellation, this is a graphical model with regular nodal degree of $k = 3$ within the internal network structure.
- (2) *Grid*: A standard lattice model; internal nodes are $k = 4$ regular.
- (3) *Triangular*: A more highly connected lattice within internal regularity of $k = 6$.

See the upper panels in Fig. 9. For more information on the construction and characteristics of the ideal lattices see Appendix E.

To generalize these lattice models and give us access to more practical topologies, we can loosen the constraints on ideal nodal positioning. This is achieved by modifying the ideal spatial embedding function Φ_l to include random perturbations to the spatial positions of nodes from their initial location. This extends our lattice generators to embody a random network model. More precisely, consider the nodewise application of the ideal lattice spatial embedding function $\Phi_l(\mathbf{i}) = \mathbf{x}_i \in \mathbb{R}^2$. We generalize the spatial embedding function to become

$$\Phi_{l,\zeta}(\mathbf{i}) = \Phi_l(\mathbf{i}) + a_l(\rho) \cdot \zeta \left[\mathbf{r}_i - \frac{1}{2} \right],$$

where $\mathbf{r}_i \sim \text{Uniform}([0, 1]^2)$, $\zeta \in \mathbb{R}_0^+$. (47)

That is, we invoke a random perturbation to each node across the lattice, shifting it some distance from the ideal position. The parameter ζ defines a *rigidity factor* that controls the magnitude of these random perturbations. Clearly, $\zeta = 0$ will

retrieve ideal lattices. See the perturbed lattices shown in the bottom panels of Fig. 9.

Note that, if the class parameters satisfy

$$\zeta \geq \frac{\sqrt{2} \max(L_x, L_y)}{a_l(\rho)}, \quad (48)$$

it is certain that the magnitude of the possible spatial perturbations exceeds the boundary of the network area. In our model, when nodes are perturbed outside the boundary, we resample them until they are found within the network area. More precisely, if Eq. (48) is satisfied, then it is guaranteed that the magnitude of the perturbation will exceed the network area, considering the worst case via a corner node and a maximum perturbation to its further opposing corner. Hence, when ζ is sufficiently large, the spatial embedding function can be considered to be simply a random sample from the network area,

$$\Phi_{l,\zeta \rightarrow \infty}(\mathbf{i}) \rightarrow \mathbf{x}_i \sim \text{Uniform}([-L, L]^2). \quad (49)$$

Similarly, each ideal lattice model obeys a network connection rule set defined by Υ_l . This rule set establishes the connectivity of the network, conditioned lattice class, and the spatial distribution of the nodes. In the perturbed lattice model, we retain precisely the same connectivity structure *when possible*; however, the introduction of perturbations will manifest as an increased variance on the single-link rate distribution, $\mathcal{R}$. As previously discussed, we also consider a pruning rate $\delta \in \mathbb{R}_0^+$ for our generalized model, which restricts the minimum acceptable single-link rate $R_{ij} \geq \delta$ between any two network

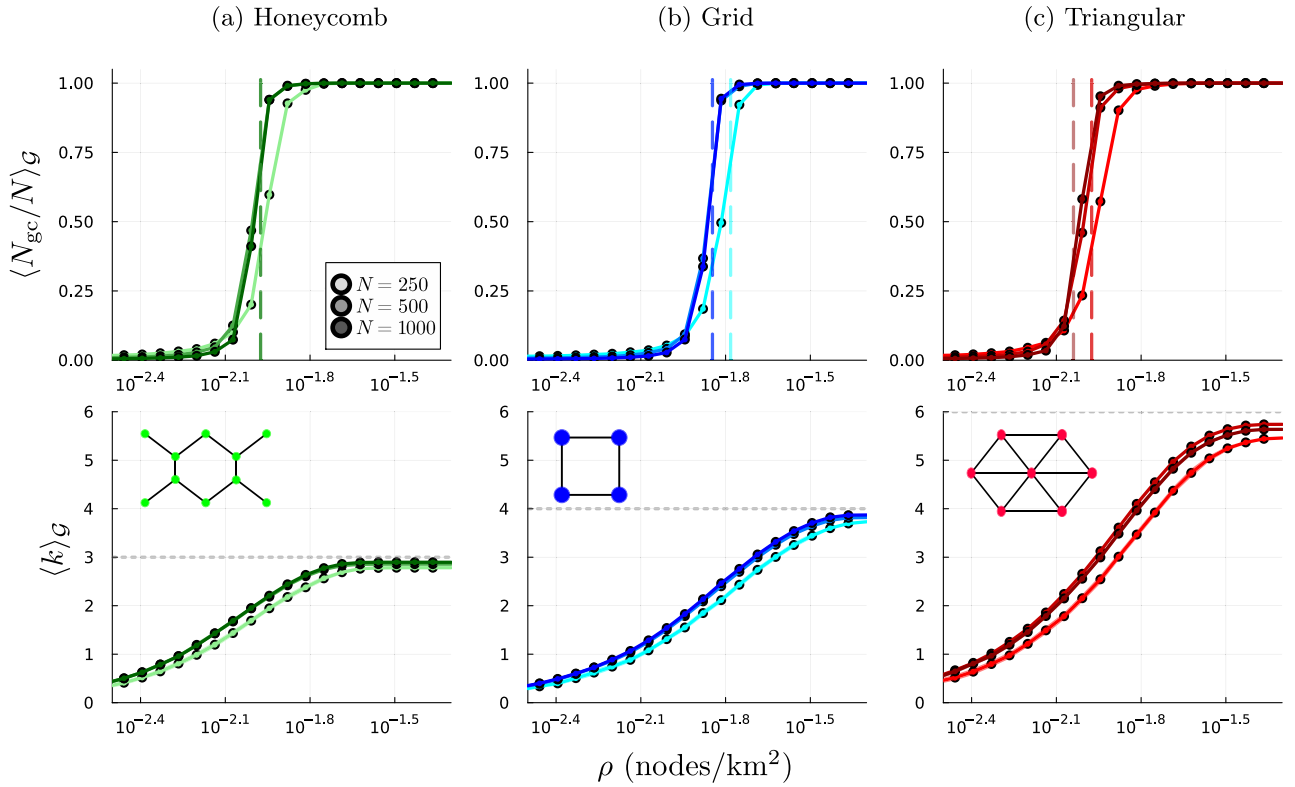


FIG. 10. Relationships between the ensemble-average giant-component fraction $\langle N_{gc}/N \rangle_G$, average degree $\langle k \rangle_G$, and nodal density ρ for CV-QKD networks using a homodyne protocol with parameters defined in Table III of Appendix B (with block-by-block PE). Ensemble-averages are computed over each class $\mathcal{G}_{l,\zeta,\delta}$, where $l \in \{\text{honey}, \text{grid}, \text{tri}\}$, $\zeta = 1$, and $\delta = 1$ bit/s, sampling $M = 50$ networks at each density ρ , for fixed node numbers $N \in \{250, 500, 1000\}$.

nodes $(i, j) \in E$. From now on we set $\delta = 1$ bit/s, meaning that a link is established only if it can support a QKD rate of at least 1 bit/s. This is a very low requirement. Note that, via the pruning rate, one could potentially extend the analysis to high-rate networks where links must have a minimum QKD rate of 10 Kbit/s or so.

We thus arrive at a complete perturbed-lattice network generators,

$$\mathcal{G}_{l,\zeta,\delta} = (\Phi_{l,\zeta}, \Upsilon_{l,\zeta,\delta}),$$

$$l \in \{\text{honey}, \text{grid}, \text{tri}\}, \quad \zeta, \delta \in \mathbb{R}_0^+, \quad (50)$$

from which we can sample network instances $\mathcal{N} \sim \mathcal{G}_{l,\zeta,\delta}$.

D. QKD network properties and percolation

One lens through which we can analyze the properties of quantum networks is through their topological characteristics. Network structure analysis can be very insightful, particularly for network classes that may obey common behaviors and features across different parameter regimes. An important concept in this regard is the giant component of a network [33,34]. Here, we study this quantity, while in the next section, we discuss its role in the security-rate trade-off.

The giant component of a network N_{gc} is the largest subset of nodes between which all nodes possess paths to one another. The giant-component fraction (GCF) is thus the ratio between the size of the giant component N_{gc} and the number of nodes N defined to belong to the network. The GCF pro-

vides insight into the successful connectivity of a topology and its ability to establish communication between any/all parties.

In the context of quantum networks, the GCF is a very useful metric that is tightly coupled to the quality of point-to-point communication links across the network. If a network is sparse, nodes that should be connected in the ideal lattice may become very far apart. The QKD link connecting these nodes may be of insufficient quality to be established (zero rate, or below some threshold acceptable rate). Consequently, connections will fail during network generation, and the resulting topology will not be fully connected. This phenomenon leads to *percolative* behavior with respect to network connectivity, i.e., phase transitions between poorly connected and fully connected topologies.

We inspect this behavior in the context of perturbed lattice-based networks $\mathcal{G}_{l,\zeta,\delta}$. Such networks will undergo GCF transitions due to their dependence on single-link rates to establish internode connections. In Fig. 10, we have evaluated the ensemble average GCF $\langle N_{gc}/N \rangle$ by repeatedly sampling networks $\mathcal{N} \sim \mathcal{G}_{l,\zeta,\delta}$ at different densities, ρ . Here, we have considered CV-QKD networks using a homodyne protocol (parameters defined in Table III of Appendix B) and a nonzero rigidity $\zeta = 1$. We can clearly see GCF transitions at different nodal densities depending on the lattice class and the number of nodes in the network.

It is also interesting to investigate the degree k of such networks, to understand how their ability to connect to

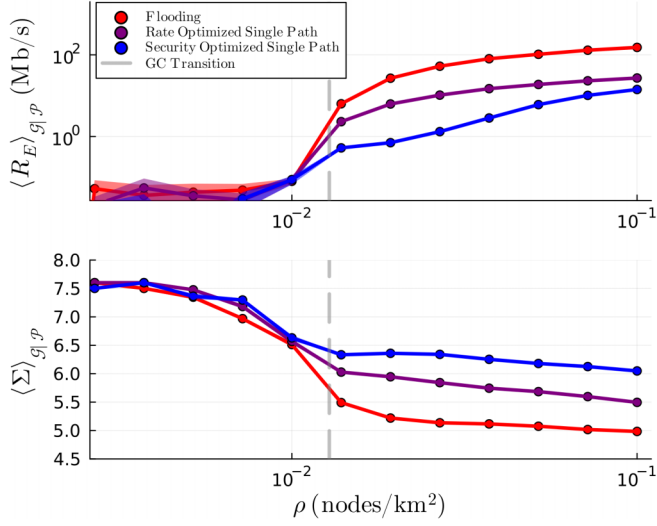


FIG. 11. Security-rate trade-off for a perturbed triangular network class ($l = \text{tri}$, $\zeta = 1$, $\delta = 1$ bit/s) defined within a rectangular area of side $L_x = 250$ km, $L_y = \frac{250\sqrt{3}}{2}$ km, using a CV-QKD homodyne protocol (parameters are given in Table III of Appendix B with block-by-block PE). We plot the ensemble average ETE rate on top, and the ETE security level below, considering the following protocols $\mathcal{P}_f$ (red), $\mathcal{P}_{sp}^{R^*}$ (purple), and $\mathcal{P}_{sp}^{\Sigma^*}$ (blue). The giant-component transition density ρ_{gc}^* is indicated via a gray dashed line on all plots. For each density ρ and routing protocol, we sample over random networks generated within the class and, for each network created in the class, we sample over random unicast connections (end-user pairs) generated in that network.

other nodes evolves across the giant-component transition. Figure 10 reports the ensemble average degree $\langle k \rangle$ for each lattice class and illustrates a slow progression, rather than a sharp transition across the giant-component transition. The strict connectivity properties of each lattice class place a fixed upper bound on the maximum possible degree of any node: honeycomb $k \leq 3$, grid $k \leq 4$, and triangular $k \leq 6$. Notice, however, that even at full connectivity, the network average does not converge perfectly to these upper bounds, since there will always exist boundary nodes with lower connective degrees.

E. Security-rate trade-off

While network structure analyses are useful, the most significant insight to understand the security-rate trade-off can be found through the study of ETE metrics via the methods of Sec. IV B. Indeed, it is interesting to investigate ensemble averages of the ETE and security level for the perturbed lattice classes $\mathcal{G}_{l,\zeta,\delta}$ and specific routing protocols $\mathcal{P}$, i.e.,

$$\langle R_E \rangle_{\mathcal{G}_{l,\zeta,\delta}|\mathcal{P}} \quad \text{and} \quad \langle \Sigma_E \rangle_{\mathcal{G}_{l,\zeta,\delta}|\mathcal{P}}.$$

Analysis of these metrics will deliver realistic insight into the security-rate trade-off, how it can manifest with respect to scaling network infrastructure, and its effect with respect to different routing strategies.

In Figs. 11 and 12, we study this trade-off in the context of two spatial network classes. Figure 11 looks at a perturbed triangular network class ($l = \text{tri}$, $\zeta = 1$, $\delta = 1$ bit/s) defined

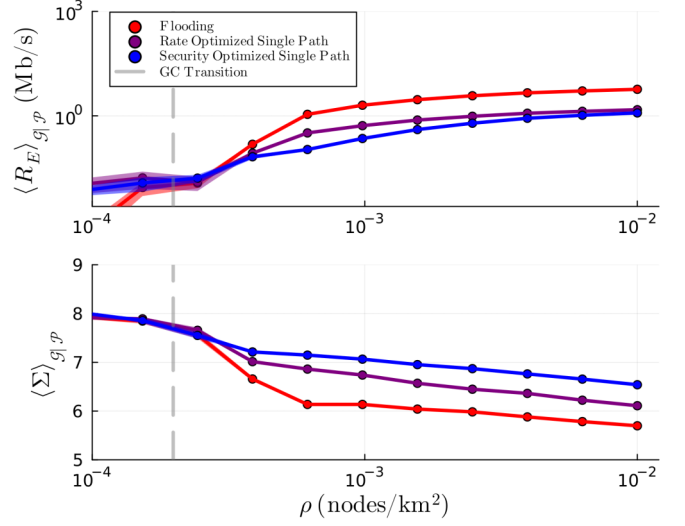


FIG. 12. Security-rate trade-off for perturbed grid network class ($l = \text{grid}$, $\zeta = 1$, $\delta = 1$ bit/s) defined within a square area of side $L = 500$ km using one-decoy BB84 protocol (parameters are given in Table II of Appendix A with block-by-block PE). We plot the ensemble average ETE rate on top, and the ETE security level below, considering the following protocols $\mathcal{P}_f$ (red), $\mathcal{P}_{sp}^{R^*}$ (purple), and $\mathcal{P}_{sp}^{\Sigma^*}$ (blue). The giant-component transition density ρ_{gc}^* is indicated via a gray dashed line on all plots. For each density ρ and routing protocol, we sample over random networks generated within the class and, for each network created in the class, we sample over random unicast connections (end-user pairs) generated in that network.

within a rectangular region of side lengths $L_x = 250$ km, $L_y = \frac{250\sqrt{3}}{2}$ km. All links employ a CV-QKD homodyne protocol, with parameters found in Table III of Appendix B. Meanwhile, Fig. 12 looks at a perturbed grid network class ($l = \text{grid}$, $\zeta = 1$, $\delta = 1$ bit/s) defined within a square area of side $L = 500$ km. All links employ a one-decoy BB84 protocol, with parameters found in Table II of Appendix A. In both cases, these results illustrate the ensemble-average relationships between ETE rate, security, and nodal density, with interesting insights related to the giant-component transition.

1. ETE performance

At low densities, below the giant-component transition densities $\rho < \rho_{gc}^*$, networks fail to fully connect. This transition takes place for both networks, but at very different densities. The ability for the BB84 protocol to sustain nonzero rates over very long distances $d(i, j) \lesssim 180$ km means that it transitions much earlier than the CV-QKD network, which comparatively has maximum link lengths of $d(i, j) \lesssim 15$ km. Shortly after the giant-component transition, as networks become reliably fully connected and nodes squeeze closer together, the average rates achieved between end-user pairs grow and scale positively.

The manner of this scaling can be very different, depending on a number of factors:

(1) *Routing protocol*: The choice of routing protocol plays a large role. The scaling of single-path $\mathcal{P}_{sp}$ performance depends on the quality of single links across the network. The faster point-to-point rates improve in quality, the sooner

single-path routing can reach high performance. However, multipath routing strategies like $\mathcal{P}_\Pi$ can scale more rapidly thanks to the exploitation of more paths between end users, amplifying ETE rates. This is evident in both Figs. 11 and 12. Furthermore, Fig. 11 particularly emphasizes the gap between rate-focused and security-focused route optimization.

(2) *QKD protocol features*: The relationship between channel length and QKD protocol performance hugely influences the trends. While the BB84 protocol is better at sustaining nonzero rates over long distances, its performance over short distances is much more limited. This is what causes the lower rate plateaus below 10 Mb/s seen in Fig. 12, due to the lack of rate increase when channels shorten. The similar rate performance between $\mathcal{P}_{sp}^{R^*}$ and $\mathcal{P}_{sp}^{\Sigma^*}$ can also be attributed to the weak sensitivity that the BB84 protocol has with respect to channel length (as the prioritization of high-rate paths makes little difference on the ETE rate). The contrary is seen in Fig. 11, where the CV-QKD protocol is very poor at longer channel lengths, but can achieve much higher rates as the density increases (on the order of 10^2 Mb/s in the figure).

(3) *Network connectivity features*: The connectivity properties of the lattice models are responsible for the upper limits of network performance. The performance plateau seen in Fig. 12 is brought about not only by the choice of QKD protocol, but also by the fact that the nodal degree is forcibly limited to $k_{\text{grid}} = 4$. Hence, multipath routing cannot continue to amplify the rate even when the network density increases. A similar phenomenon would be evident for its CV-QKD counterpart as the densities continue to grow.

2. ETE security

Several factors contribute to trends visualized for $\langle R_E \rangle_{\mathcal{G}_{l,\zeta,\delta}|\mathcal{P}}$ in Figs. 11 and 12. Now, we wish to understand the trade-off between these trends and those that we witness for the ETE security $\langle \Sigma_E \rangle_{\mathcal{G}_{l,\zeta,\delta}|\mathcal{P}}$. Indeed, we witness the direct trade-off between rate and security in both network contexts, but in subtly different ways.

Once more, at low densities $\rho \leq \rho_{\text{gc}}^*$ we do not expect many networks to be fully connected. As a consequence, the networks for which ETE routes are successfully sampled and analyzed will typically consist of very short paths, between spatially nearby end users (distant users will not be connected). This gives rise to the trend we see of higher initial security levels for both network classes. However, as the density increases, networks become more connected, average rates increase, and average security *decreases* as we would expect. Greater connectivity leads to more resources being consumed and thus a degradation of Σ_E .

In both network classes, but more evidently in the CV-QKD triangular grid, we observe an interesting feature when the ETE security is optimized, i.e., $\mathcal{P}_{sp}^{\Sigma^*}$ as the routing protocol. The ETE security experiences a clear “rebound” shortly after the giant-component transition, where the security $\langle \Sigma_E \rangle_{\mathcal{G}_{l,\zeta,\delta}|\mathcal{P}}$ actually improves with respect to increasing density. This can be attributed to a phase experienced by these networks where shorter paths suddenly become available, which did not exist before due to incomplete connective structures. Eventually, this rebound fades as the networks become denser and the average path length between nodes inevitably

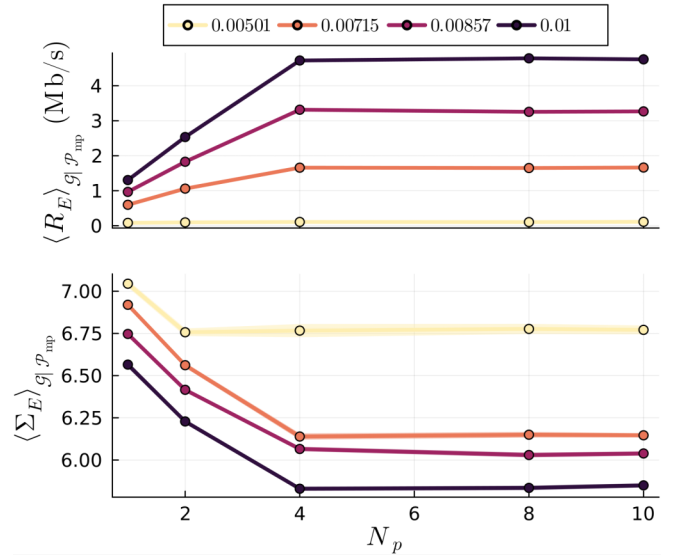


FIG. 13. We plot the ensemble-average ETE rate $\langle R_E \rangle_{\mathcal{G}|\mathcal{P}}$ (upper panel) and the ETE security level $\langle \Sigma_E \rangle_{\mathcal{G}|\mathcal{P}}$ (lower panel) vs the number of paths N_p used in the multipath protocol $\mathcal{P} = \mathcal{P}_{\text{mdp}}^{N_p}$. Here, we consider grid lattice networks from $\mathcal{G}_{\text{grid},\zeta,\delta}$ ($\zeta = 1$, $\delta = 1$ bit/s) using one-decoy BB84 with the parameters in Table II of Appendix A (with block-by-block PE). These are generated according to the densities ρ listed in the top legend. For each density and number of paths, we sample over random networks generated within the class and, for each network created in the class, we sample over random unicast connections (end-user pairs) generated in that network.

increases again. Note, clearly this rebound is not experienced for the strategies that only deploy rate optimization, as their objectives do not incentivize shortening routes at the expense of performance. This feature is indicative of the need for security-focused route optimization.

3. Security-rate trade-off in multipath routing

Finally, it is interesting to directly inspect the security-rate trade-off for the number of paths deployed in multipath protocols $\mathcal{P}_{\text{mdp}}^{N_p}$. In Fig. 13, we inspect this relationship for a perturbed grid network class ($l = \text{grid}$, $\zeta = 1$, $\delta = 1$ bit/s) defined within a square area of side $L = 500$ km. All links employ a one-decoy BB84 protocol, with parameters found in Table II of Appendix A.

We display the ensemble-average ETE rate $\langle R_E \rangle_{\mathcal{G}_{l,\zeta,\delta}|\mathcal{P}}$ and security level $\langle \Sigma_E \rangle_{\mathcal{G}_{l,\zeta,\delta}|\mathcal{P}}$ for a number of nodal densities. In this way, one can clearly observe the trade-off in action with respect to increasing path usage. The rate amplification/security degradation is more pronounced at higher densities, where the opportunity for using more paths is greater. Nonetheless, in each case a maximum number of paths N_p^{max} is reached after which the security/rate plateaus. For lattice networks, this maximum value will equal the maximum degree of the lattice $N_p^{\text{max}} = k_l$, since the degree k_l bounds the maximum number of disjoint paths possible on the network. Indeed, this is observed for the grid network in Fig. 13, where $N_p > 4$ induces plateaus in the ensemble averages.

V. EXTENSION TO AUTHENTICATION

Our previous results for single links and networks can be extended to include authentication. Suppose that, on a link, we run an $\varepsilon_{\text{block}}$ -secure QKD protocol and use an $\varepsilon_{\text{auth}}$ -secure authentication scheme that is iterated a times per round (block) of the QKD session (with n_{bks} blocks). Then, following Ref. [35], the overall epsilon security of the session is bounded by $n_{\text{bks}}(a\varepsilon_{\text{auth}} + \varepsilon_{\text{block}})$.

For the first QKD block, the parties need an initial preshared authentication key of length $K_{\text{init}} = b_{\text{hash}} + ab_{\text{tag}}$, where b_{hash} is the number of bits needed to select a universal hash function (hash function key bits) and $b_{\text{tag}} \simeq \lceil \log_2(1/\varepsilon_{\text{auth}}) \rceil$ is the bit length of the tag used in the Wegman-Carter authentication of each of the a authentication messages [36]. Using key recycling [35], the initial b_{hash} bits are recycled in later blocks (and sessions), while the ab_{tag} bits are consumed as fresh one-time-pad material for the authentication tags. After the first block, fresh tag keys can be drawn from the QKD output of previous blocks.

The impact on the key size can therefore be estimated as follows. Assume that a single block produces a secret key length of l bits. Then, thanks to key recycling [35], this would be reduced to $l \rightarrow l - ab_{\text{tag}}$. For typical parameters, such as $a \simeq 10$ and $b_{\text{tag}} \simeq 10^2$, we have $\delta_{\text{auth}} \simeq 10^3$ per block. If the QKD implementation is high rate, producing key generation blocks with $l = 10^5 - 10^6$ bits, then this contribution is negligible, and the only overhead is at the level of the total epsilon security. In other low-rate scenarios, the cost for link authentication needs to be accounted for as a further penalty for the key rate.

VI. CONCLUSIONS

We have investigated the fundamental trade-off between composable security and key rate in QKD at both the link and network layers. For individual links, our analysis has examined the interplay between the security level and the achievable key rate for DV- and CV-QKD protocols, such as the BB84 and the Gaussian-modulated protocols, assuming various strategies for the crucial process of parameter estimation. We have examined how enforcing a given security level impacts the key rate of these protocols, both for single-block processing and for multiblock session processing (more suitable for practical implementations).

We have then extended our analysis to the framework of multihop QKD networks, starting from basic topologies to more complex configurations (Waxman meshes, perturbed lattices). In this setting, we have analyzed the interplay between security and rate for ETE connections, which are randomly selected within the network. We have quantified how ETE security degrades with the use of multipath routing strategies, despite the associated rate gains. For single-path routing, the “lambda protocol,” a security-rate-aware modification of Dijkstra’s algorithm, enables route selection that balances these competing demands. This protocol can be tuned to be security optimized or rate optimized in extreme cases.

Our investigation addresses two levels of ETE metrics: average ETE metrics, obtained by averaging over all ETE connections within a single network, and ensemble-average

ETE metrics, obtained by further averaging across networks belonging to the same class. In the latter case, we demonstrate that giant-component transitions, occurring at specific critical nodal densities, can trigger a sharp decoupling between security and rate in the trade-off. The magnitude of this behavior depends on several factors, including the QKD protocol implemented at the link layer and the type of lattice assumed for the network.

In conclusion, our results provide a unified framework for understanding and optimizing the fundamental security-rate trade-off in QKD networks, bridging the gap between link-level protocol design and large-scale network architecture.

ACKNOWLEDGMENTS

UKRI supported this work through the project “QAsure” (Innovate UK, Grant No. 10102791) and the Integrated Quantum Networks (IQN) Research Hub (EPSRC, Grant No. EP/Z533208/1). S.P. would like to thank Matthew Barber, Shounak Joshi, Alex Mountogiannakis, and Ignazio Pedone for discussions.

DATA AVAILABILITY

The data and code supporting the findings of this article are available from the corresponding author upon reasonable request, except for the underlying data, code, and execution environment used to generate the numerical results shown in Figs. 8–17. These figures were generated using a proprietary digital-twin platform developed by nodeQ. The platform and associated materials are not publicly available because of commercial sensitivity, confidentiality obligations, and licensing restrictions. Enquiries regarding access under appropriate confidentiality and licensing arrangements may be directed to nodeQ at Ref. [37].

APPENDIX A: COMPOSABLY SECURE DV-QKD

Here, we start with a detailed description of the one-photon BB84 protocol and its key rate (Appendix A 1), together with a simple proof of its composable security (Appendix A 2). Afterward, in Appendix A 3, we consider one-decoy BB84 in the composable security framework, for which we present the detailed steps of the protocol and a revised rate that combines elements from Refs. [38,39].

1. One-photon BB84: Description and key rate

We consider symmetric BB84, meaning that the two bases Z and X are treated equivalently for key generation and PE, even though they may be chosen with different probabilities. In contrast, in the asymmetric BB84, one basis is completely used for PE while the other basis is used for key generation with a potential partial sacrifice for PE (e.g., this is the case studied in Refs. [40,41]). The schematic steps of the symmetric BB84 protocol are the following.

(1) *Encoding/decoding*: Alice and Bob exchange N states (this is the initial block size). The bases Z and X are chosen with probabilities q_z and $q_x = 1 - q_z$. So we have the sifting

probability

$$p_{\text{sift}} = q_z^2 + q_x^2. \quad (\text{A1})$$

(2) *Channel*: The states are exchanged with total transmissivity $\eta = \eta_{\text{ch}}\eta_{\text{eff}}$ due to both channel loss and Bob's setup efficiency. In particular, in a fiber of length d (in km) and loss rate α_L (typically, 0.2 dB/km), the channel transmissivity is given by

$$\eta_{\text{ch}} = 10^{-\alpha_L d/10}. \quad (\text{A2})$$

(3) *Sifting*: Bob registers a detection because of either a received signal or a dark count, which occurs with probability p_{dc} . Bob's detection probability is therefore

$$Q_1 = \eta + (1 - \eta)p_{\text{dc}}, \quad (\text{A3})$$

reducing to $Q_1 = \eta$ with no dark counts. The probability of a sifted signal is $p = p_{\text{sift}}Q_1$. Given N initial signals (size of the block), the number of sifted signals is

$$M = pN = p_{\text{sift}}Q_1N. \quad (\text{A4})$$

(4) *Parameter estimation*: For some agreed PE ratio r_{PE} , Alice and Bob randomly sacrifice

$$m = r_{\text{PE}}M \quad (\text{A5})$$

signals (from both bases) to estimate the QBER E up to an error ε_{PE} . Let us call $\hat{E}$ this estimator. They also compute the worst-case value

$$\tilde{E} = \hat{E} + \sqrt{(2/m) \ln[(m+1)/\varepsilon_{\text{PE}}]}. \quad (\text{A6})$$

Note that the expected value of the QBER E^* is determined by various known parameters, including the dark counts p_{dc} and the misalignment error e_{det} , as given by

$$E^* = \frac{p_{\text{dc}}/2 + e_{\text{det}}\eta}{\eta + p_{\text{dc}}}. \quad (\text{A7})$$

In an experiment, $\hat{E}$ is computed from the data as an output. In a theoretical study, we set $\hat{E} = E^*$.

(5) *Error correction*: The instances surviving PE form the “raw key,” with length

$$n = M - m = \kappa N, \quad (\text{A8})$$

where κ is the raw key rate, expressed by

$$\kappa = (1 - r_{\text{PE}})p_{\text{sift}}Q_1. \quad (\text{A9})$$

The raw key undergoes EC with success probability p_{EC} and “correctness” error ε_{cor} , which is associated with the collision probability during hashing verification.

An important term in EC is $\text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}}$. This is the total number of bits leaked in the procedure of EC, due to the disclosure of the syndromes for correction and the communication of the hashes. The leakage term can be written as

$$\begin{aligned} \text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}} &= n f_{\text{EC}} h(\hat{E}) + \lceil \log_2(1/\varepsilon_{\text{cor}}) \rceil \\ &\simeq n f_{\text{EC}} h(\hat{E}) + \log_2(2/\varepsilon_{\text{cor}}), \end{aligned} \quad (\text{A10})$$

where $h(\cdot)$ is the binary Shannon entropy and $f_{\text{EC}} \geq 1$ is the reconciliation efficiency (e.g., equal to 1.16 or 1.22).

Assuming that low-density parity-check (LDPC) codes are used for EC, we have that the reconciliation efficiency depends on the rate of the code R_{code} according to

$$f_{\text{EC}} = \frac{1 - R_{\text{code}}}{h(\hat{E})}, \quad (\text{A11})$$

where the actual value of R_{code} is chosen according to the estimated QBER.

Remark A1. Note that there is an implicit trade-off between ε_{cor} and p_{EC} . The lower the value of ε_{cor} , the more difficult is the verification step (since we need to compare longer hashes); in turn, this decreases the probability p_{EC} of successful EC.

(6) *Privacy amplification*: During PA, Alice and Bob compress their corrected and verified raw keys (of length n) into the final secret key of length l . The secret key length l can be expressed as a fraction of the raw key length n as $l = rn$, where r is the secret fraction. The process of PA has a total error called “epsilon secrecy” $\varepsilon_{\text{sec}} = \varepsilon_s + \varepsilon_h$, where ε_s is the smoothing parameter and ε_h is the hashing parameter.

The secret fraction can be computed from protocol parameters and estimators. We have

$$r = 1 - h(\tilde{E}) - (\text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}} + \Delta)/n, \quad (\text{A12})$$

where the finite-size term Δ includes terms associated with PA (converse leftover hash bound) and the asymptotic equipartition property (AEP). It can be broken down as follows:

$$\Delta = 2 \log_2(1/2\varepsilon_h) + \sqrt{n} \Delta_{\text{AEP}}(\varepsilon_s), \quad (\text{A13})$$

where the AEP term is given by [Eq. (8) of Ref. [40]]

$$\Delta_{\text{AEP}}(\varepsilon_s) = 7\sqrt{\log_2(2/\varepsilon_s)}. \quad (\text{A14})$$

By substituting Eq. (A10) in Eq. (A12), we get

$$r = 1 - h(\tilde{E}) - f_{\text{EC}} h(\hat{E}) - \frac{\log_2(2/\varepsilon_{\text{cor}}) + \Delta}{n}. \quad (\text{A15})$$

Remark A2. Note that, if we take the limit for an infinite number of signals (i.e., large n), we have $\tilde{E} \rightarrow \hat{E} \rightarrow E$ and $\text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}} \rightarrow nh(E)$ (where, in the limit, we can also assume perfect error correction $f_{\text{EC}} = 1$). Overall, the expression in Eq. (A15) collapses to $r = 1 - 2h(E)$, which is the well-known asymptotic key rate.

Remark A3. For the AEP term, we can also find the expression (Theorem 6.4 in Ref. [42]) (see also Eq. (G11) of Ref. [6])

$$\tilde{\Delta}_{\text{AEP}}(\varepsilon_s) = 4 \log_2(2\sqrt{2} + 1) \sqrt{\log_2 2/\varepsilon_s^2}. \quad (\text{A16})$$

However, Eq. (A14) is less pessimistic than Eq. (A16). In fact, checking $\Delta_{\text{AEP}} \leq \tilde{\Delta}_{\text{AEP}}$ is equivalent to check

$$\sqrt{\frac{1-2x}{1-x}} \geq \frac{7}{4 \log_2(2\sqrt{2} + 1)}, \quad (\text{A17})$$

for $x = \log_2 \varepsilon_s < 0$. Indeed, this is always true for negative x . In other words, Eq. (A14) is a better choice.

(7) *Secret key rate*: Introducing Eq. (A13) into Eq. (A15), we get the following secret fraction:

$$\begin{aligned} r &= 1 - h(\tilde{E}) - f_{\text{EC}} h(\hat{E}) \\ &\quad + \frac{1 + \log_2(\varepsilon_{\text{cor}}\varepsilon_h^2)}{n} - \frac{\Delta_{\text{AEP}}(\varepsilon_s)}{\sqrt{n}}. \end{aligned} \quad (\text{A18})$$

See Appendix A2 for a complete proof of Eq. (A18). We obtain the secret key rate R (bits/use) by multiplying the secret fraction r by the raw key rate κ , i.e.,

$$R = \kappa r = l/N, \quad (\text{A19})$$

with total epsilon security $\varepsilon = \varepsilon_{\text{cor}} + p_{\text{EC}}\varepsilon_{\text{PE}} + \varepsilon_{\text{sec}}$. More simply, we use the upper bound

$$\varepsilon = \varepsilon_{\text{cor}} + \varepsilon_{\text{PE}} + \varepsilon_{\text{sec}}, \quad \varepsilon_{\text{sec}} = \varepsilon_s + \varepsilon_h. \quad (\text{A20})$$

Note that we may simplify the security level of the protocol if we set $\varepsilon_s = \varepsilon_h = \varepsilon_{\text{sec}}/2$. In this way, the secret fraction becomes

$$r = 1 - h(\tilde{E}) - f_{\text{EC}}h(\hat{E}) + \frac{\log_2(\varepsilon_{\text{cor}}\varepsilon_{\text{sec}}^2) - 1}{n} - \frac{\Delta_{\text{AEP}}(\varepsilon_{\text{sec}}/2)}{\sqrt{n}}. \quad (\text{A21})$$

Alternatively, we may also rewrite it as follows:

$$r = 1 - h(\tilde{E}) + \frac{2\log_2(\varepsilon_{\text{sec}}) - \text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}}}{n} - \frac{\Delta_{\text{AEP}}(\varepsilon_{\text{sec}}/2)}{\sqrt{n}}, \quad (\text{A22})$$

where $\text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}}$ is given in Eq. (A10).

Remark A4. The formula for the secret fraction in Eq. (A18) is more precisely an upper bound $r \leq r_{\text{UB}}$. The meaning is that, for a fixed value of the epsilon security ε , the length of the final key l cannot exceed a maximum value l_{UB} , which means that PA must compress below this upper threshold. We can provide a more pessimistic estimate of this threshold if we consider the direct leftover hash bound instead of the converse bound (see Appendix A2 for details). In such a case, we get the secret fraction below:

$$r_{\text{LB}} = 1 - h(\tilde{E}) - f_{\text{EC}}h(\hat{E}) + \frac{\log_2(\varepsilon_{\text{cor}}\varepsilon_h^2)}{n} - \frac{\Delta_{\text{AEP}}(\varepsilon_s)}{\sqrt{n}}. \quad (\text{A23})$$

Note that r_{LB} is $1/n$ bit less than the upper bound in Eq. (A18). For large n , the difference becomes negligible.

(8) *Secret key rate (bits/s):* To get the rate in terms of bits/s, we need to do the following:

(a) Multiply by the repetition rate r_s .

(b) Account for the detector's dead time τ_{dt} , so there is a dead time rate [Eq. (10) of Ref. [16]]

$$c_{\text{dt}} = \frac{1}{1 + (r_s\tau_{\text{dt}} - 1)Q_1} \simeq \frac{1}{1 + r_s\tau_{\text{dt}}Q_1}. \quad (\text{A24})$$

Accounting for these features, we write the key rate of the block in bits/s, i.e.,

$$R_{\text{block}} = r_s c_{\text{dt}} R. \quad (\text{A25})$$

This is the conditional key rate of the protocol (conditioned on successful EC). The unconditional key rate of the protocol will be given by the average value $p_{\text{EC}}R_{\text{block}}$.

(9) *QKD session:* In a realistic scenario, the parties perform a QKD session of n_{bks} blocks, each block with size N (for a total of N_{ses} states transmitted). Each block survives EC with probability p_{EC} , so there will be an average of $p_{\text{EC}}n_{\text{bks}}$

successful blocks. The average key rate of the session is equal to the unconditional key rate

$$R_{\text{ses}} = p_{\text{EC}}R_{\text{block}}, \quad (\text{A26})$$

and the total epsilon security will be

$$\varepsilon_{\text{ses}} = n_{\text{bks}}\varepsilon. \quad (\text{A27})$$

In a QKD session, there are two options for PE:

(a) PE is done block by block. This scenario is robust to potential changes during the session. The session rate will be averaged over the block-based rates.

(b) PE is done on the entire $n_{\text{bks}}m$ points of the session. This generates better estimators and therefore improves the rate. However, this approach relies on the stability of the channel.

In Sec. II, we consider both block-by-block and session-wide PE.

2. Proof of Eq. (A18)

Note that we denote systems in bold for clarity, so $\mathbf{X}$ is Alice's classical system and $\mathbf{E}$ is Eve's quantum system. The latter must not be confused with the QBER E later used in the proof.

The *secrecy condition* stated in Remark 6.1.3 in Ref. [43] corresponds to the following bound on the trace distance d between Alice-Eve state after PA $\bar{\rho} := \bar{\rho}_{\mathbf{XE}}$ and the ideal state $\rho_{\text{id}} := \frac{\text{id}_{\mathbf{X}}}{\dim(\mathcal{H}_{\mathbf{X}})} \otimes \rho_{\mathbf{E}}$:

$$d(\bar{\rho}, \rho_{\text{id}}) \leq \varepsilon_{\text{sec}}, \quad (\text{A28})$$

where

$$d(\bar{\rho}, \rho_{\text{id}}) := \frac{1}{2} \left\| \bar{\rho}_{\mathbf{XE}} - \frac{\text{id}_{\mathbf{X}}}{\dim(\mathcal{H}_{\mathbf{X}})} \otimes \rho_{\mathbf{E}} \right\|_1. \quad (\text{A29})$$

From the PA described in Corollary 5.6.1 in Ref. [43], one has

$$d(\bar{\rho}, \rho_{\text{id}}) \leq \varepsilon_s + \frac{1}{2} 2^{-\frac{1}{2}[H_{\min}^{\varepsilon_s}(\bar{\rho}) - l]}, \quad (\text{A30})$$

where l is the length of the secret key and $H_{\min}^{\varepsilon_s}(\bar{\rho})$ is the conditional smooth min-entropy of Alice-Eve joint state before PA $\bar{\rho} := \bar{\rho}_{\mathbf{XE}}$.

To guarantee that Eq. (A28) is true, one imposes the following sufficient condition:

$$\varepsilon_s + \frac{1}{2} 2^{-\frac{1}{2}[H_{\min}^{\varepsilon_s}(\bar{\rho}) - l]} \leq \varepsilon_{\text{sec}}. \quad (\text{A31})$$

By reformulating the above relation, we obtain

$$\varepsilon_s + \frac{1}{2} 2^{-\frac{1}{2}[H_{\min}^{\varepsilon_s}(\bar{\rho}) - l]} \leq \varepsilon_{\text{sec}}, \quad (\text{A32})$$

$$2^{-\frac{1}{2}[H_{\min}^{\varepsilon_s}(\bar{\rho}) - l]} \leq 2(\varepsilon_{\text{sec}} - \varepsilon_s), \quad (\text{A33})$$

$$-\frac{1}{2}[H_{\min}^{\varepsilon_s}(\bar{\rho}) - l] \leq \log_2[2(\varepsilon_{\text{sec}} - \varepsilon_s)], \quad (\text{A34})$$

$$l - H_{\min}^{\varepsilon_s}(\bar{\rho}) \leq 2\log_2[2(\varepsilon_{\text{sec}} - \varepsilon_s)], \quad (\text{A35})$$

$$l \leq H_{\min}^{\varepsilon_s}(\bar{\rho}) - 2\log_2\left[\frac{1}{2(\varepsilon_{\text{sec}} - \varepsilon_s)}\right] =: l^{\text{UB}}. \quad (\text{A36})$$

This upper bound provides the longest key that we are allowed to extract for a fixed epsilon secrecy ε_{sec} . For $l \leq l^{\text{UB}}$, we certainly have $d(\bar{\rho}, \rho_{\text{id}}) \leq \varepsilon_{\text{sec}}$. Note that:

(1) By setting $\varepsilon_s := \frac{2}{3}\varepsilon_{\text{sec}}$, one obtains the upper bound in the form

$$l \leq H_{\min}^{\varepsilon_s}(\tilde{\rho}) - 2 \log_2(1/\varepsilon_s), \quad (\text{A37})$$

with the same correction term as in Lemma 6.4.1 in Ref. [43].

(2) In general, we set $\varepsilon_h = \varepsilon_{\text{sec}} - \varepsilon_s$ and write

$$l^{\text{UB}} = H_{\min}^{\varepsilon_s}(\tilde{\rho}) - 2 \log_2 \frac{1}{2\varepsilon_h}. \quad (\text{A38})$$

We can include the EC in the description following Lemma 6.4.1 in Ref. [43] but with our definition of trace distance. A protocol is epsilon secret ε_{sec} and epsilon correct ε_{cor} when the key length satisfies the converse leftover hash bound [44]

$$l \leq l^{\text{UB}} = H_{\min}^{\varepsilon_s}(\tilde{\rho}) - 2 \log_2 \frac{1}{2\varepsilon_h} - \text{leak}_{\text{EC}}. \quad (\text{A39})$$

Next, we use the AEP and the explicit expression of the leakage term. For n uses ($\tilde{\rho} = \tilde{\rho}_1^{\otimes n}$), the AEP takes the form

$$H_{\min}^{\varepsilon_s}(\tilde{\rho}) \geq nH(\mathbf{X}|\mathbf{E})_{\tilde{\rho}_1} - \sqrt{n}\Delta_{\text{AEP}}(\varepsilon_s), \quad (\text{A40})$$

$$\Delta_{\text{AEP}}(\varepsilon_s) = -7\sqrt{\log_2(2/\varepsilon_s)}. \quad (\text{A41})$$

Remark A5. It is important to note that the tensor-product structure $\tilde{\rho}_1^{\otimes n}$ (and other aspects in this proof) relies on the assumption of collective attack. As also discussed in Ref. [41], the assumption of collective attacks can be made without loss of generality for the BB84 protocol, so that proving security against collective attacks automatically shows security against general coherent attacks [45–47].

Remark A6. Note that in Theorem 6.4 in Ref. [42] [see also Eq. (G11) of Ref. [6]], we have the different direct bound

$$H_{\min}^{\varepsilon_s}(\tilde{\rho}) \geq nH(\mathbf{X}|\mathbf{E})_{\tilde{\rho}_1} - \sqrt{n}\tilde{\Delta}_{\text{AEP}}(\varepsilon_s), \quad (\text{A42})$$

$$\tilde{\Delta}_{\text{AEP}}(\varepsilon_s) = 4 \log_2(2\sqrt{2} + 1) \sqrt{\log_2(2/\varepsilon_s^2)}. \quad (\text{A43})$$

As in Eq. (A10), we may then write

$$\text{leak}_{\text{EC}} \leq n f_{\text{ECH}}(E) + \log_2(2/\varepsilon_{\text{cor}}), \quad (\text{A44})$$

where E is the QBER. By replacing in Eq. (A39), we then get a more pessimistic UB for l , i.e.,

$$\begin{aligned} l &\leq l^{\text{PUB}} = nH(\mathbf{X}|\mathbf{E}) - \sqrt{n}\Delta_{\text{AEP}}(\varepsilon_s) \\ &\quad - 2 \log_2 \frac{1}{2\varepsilon_h} - n f_{\text{ECH}}(E) - \log_2(2/\varepsilon_{\text{cor}}) \\ &= n[H(\mathbf{X}|\mathbf{E}) - f_{\text{ECH}}(E)] \\ &\quad - \sqrt{n}\Delta_{\text{AEP}}(\varepsilon_s) + 1 + \log_2(\varepsilon_{\text{cor}}\varepsilon_h^2). \end{aligned} \quad (\text{A45})$$

Note that we may impose a stricter upper bound by using $H(\mathbf{X}|\mathbf{E}) \geq 1 - h(\tilde{E})$ similar to the formula given for asymmetric BB84 in Eq. (6) of Ref. [41].

We then account for PE with an epsilon error ε_{PE} . This corresponds to considering an estimator $\tilde{E}$ and a worst-case value $\hat{E}$ for the QBER as described in Appendix A1. We therefore get

$$\begin{aligned} l &\leq l^{\text{PUB-PE}} = n[1 - h(\tilde{E}) - f_{\text{ECH}}(\hat{E})] \\ &\quad - \sqrt{n}\Delta_{\text{AEP}}(\varepsilon_s) + 1 + \log_2(\varepsilon_{\text{cor}}\varepsilon_h^2), \end{aligned} \quad (\text{A46})$$

which becomes Eq. (A18) when we consider the secret fraction $r = l/n$.

Remark A7. Note that we can also consider the direct leftover hash bound [Eq. (8.7) of Ref. [42]]

$$l \geq l^{\text{LB}} = H_{\min}^{\varepsilon_s}(\tilde{\rho}) - 2 \log_2 \frac{1}{\sqrt{2\varepsilon_h}} - \text{leak}_{\text{EC}}. \quad (\text{A47})$$

When we consider the direct leftover hash bound instead of the converse and we repeat the steps above, the only difference is in the term $2 \log_2(\sqrt{2\varepsilon_h})$ instead of $2 \log_2(2\varepsilon_h)$, so we have to subtract 1 from the final key size in Eq. (A46). This leads to the lower bound

$$\begin{aligned} l^{\text{PLB-PE}} &= n[1 - h(\tilde{E}) - f_{\text{ECH}}(\hat{E})] \\ &\quad - \sqrt{n}\Delta_{\text{AEP}}(\varepsilon_s) + \log_2(\varepsilon_{\text{cor}}\varepsilon_h^2), \end{aligned} \quad (\text{A48})$$

which leads to the secret fraction in Eq. (A23).

3. One-decoy BB84 with composable security

This is the protocol from Ref. [38], but where we improve the gamma function following Ref. [39]. From an operational point of view, we can see the one-decoy protocol as a specific case of the two-decoy protocol [48], where the second decoy (third intensity) is selected with zero probability $p_3 = 0$. However, this choice also requires extracting error rates from the Z basis. This can be done in two ways: Either by sacrificing some of the Z instances during PE, or by performing the error analysis after EC. In this second approach, Bob compares his initially decoded string with the error-corrected one. Assuming that the latter is equal to Alice's string, the bitwise difference will quantify the bit flips introduced by the channel.

We adopt the second strategy. Let us break down the steps and theoretical elements for the one-decoy protocol.

(1) *Encoding:*

(a) Alice randomly chooses a bit $y \in \{0, 1\}$.

(b) Alice chooses a basis $a \in \{Z, X\}$ with probability q_z and $q_x = 1 - q_z$.

(c) Alice chooses an intensity $k \in \mathcal{K} = \{\mu_1, \mu_2\}$, with $\mu_1 > \mu_2$, according to a probability distribution $\{p_k\} = \{p_{\mu_1}, p_{\mu_2}\}$. We identify the first intensity as signal ($\mu_1 = \mu_s$, $p_{\mu_1} = p_s$), and the second intensity as decoy ($\mu_2 = \mu_d$, $p_{\mu_2} = p_d$).

(d) For the chosen intensity k , Alice generates an n -photon state with probability

$$p_A^k(n) = \frac{k^n}{n!} e^{-k}. \quad (\text{A49})$$

Overall, Alice sends an n -photon state with probability

$$\tau_n = \sum_k p_k p_A^k(n). \quad (\text{A50})$$

(2) *Transmission (loss and QBER):*

(a) Total transmissivity $\eta = \eta_{\text{ch}}\eta_{\text{eff}}$ with contributions from the channel (η_{ch}) and the setup (η_{eff}). See Eq. (A2) for the relation between η_{ch} , the fiber loss rate α_L , and the fiber distance d .

(b) QBER. The channel has a QBER E evaluated from the errors in the X basis. This will be evaluated in the first stage of parameter estimation. See point (6) below.

(3) *Decoding*:

(a) Bob chooses a basis $b \in \{Z, X\}$ with probability q_z and $q_x = 1 - q_z$ (as Alice).

(b) Bob uses two single-photon detectors that provide a measurement outcome $y' \in \{0, 1, \emptyset, \perp\}$, where

- (i) 0 if only first detector clicks;
- (ii) 1 if only second detector clicks;
- (iii) $\emptyset$ corresponds to no detection;
- (iv) $\perp$ corresponds to a double detection.

(c) Bob assigns an output bit as follows:

- (i) $0 \rightarrow 0$;
- (ii) $1 \rightarrow 1$;
- (iii) $\emptyset \rightarrow$ no assignment (discard instance);
- (iv) $\perp \rightarrow 0$ or 1 randomly.

(4) *Basis reconciliation (sifting)*:

(a) For each instance i of quantum communication, Alice and Bob use a public channel to do the following:

(i) Select the instances chosen on the same basis, i.e., i such that $a_i = b_i$.

(ii) Remove the instances with no detection on Bob's side.

(iii) Classify the instances in terms of the chosen basis and Alice's chosen intensity.

(b) Overall, for each basis and $k \in \mathcal{K}$, they create the following sets of sifted instances:

- (i) $\mathcal{Z}_k = \{i : a_i = b_i = Z \wedge k_i = k \wedge y'_i \neq \emptyset\}$.
- (ii) $\mathcal{X}_k = \{i : a_i = b_i = X \wedge k_i = k \wedge y'_i \neq \emptyset\}$.

From these sets, we can define the associated strings of values possessed by Alice and Bob:

- (iii) $\mathcal{Z}_k^A = \{y_i, i \in \mathcal{Z}_k\}$ and $\mathcal{X}_k^A = \{y_i, i \in \mathcal{X}_k\}$ for Alice.
- (iv) $\mathcal{Z}_k^B = \{y'_i, i \in \mathcal{Z}_k\}$ and $\mathcal{X}_k^B = \{y'_i, i \in \mathcal{X}_k\}$ for Bob.

(c) Alice and Bob perform the quantum communication until they reach a block size N , such that

$$|\mathcal{Z}_k| \geq n_{Z,k}, \quad |\mathcal{X}_k| \geq n_{X,k}, \quad (\text{A51})$$

where the values of $n_{Z,k}$ and $n_{X,k}$ must be statistically significant (the minimum values of these event numbers can be agreed in advance by the parties). Note that, starting from the satisfaction of Eq. (A51), Alice and Bob can always choose random subsets of their distributed data to match those lengths exactly.

(d) The total sets of Z - and X -sifted instances are $\mathcal{Z} = \cup_k \mathcal{Z}_k$ and $\mathcal{X} = \cup_k \mathcal{X}_k$. Their sizes are equal to

$$n_Z = \sum_k n_{Z,k}, \quad n_X = \sum_k n_{X,k}. \quad (\text{A52})$$

Note that $\mathcal{Z}^A = \cup_k \mathcal{Z}_k^A$ and $\mathcal{X}^A = \cup_k \mathcal{X}_k^A$ represent Alice's strings distributed in Z and X bases. Similarly, for Bob we have $\mathcal{Z}^B = \cup_k \mathcal{Z}_k^B$ and $\mathcal{X}^B = \cup_k \mathcal{X}_k^B$.

(e) For the event numbers, we may theoretically write

$$n_{Z,k} \simeq n_{Z,k}^*, \quad n_{X,k} \simeq n_{X,k}^*, \quad (\text{A53})$$

where n are the observed values, while n^* are the expected (i.e., asymptotic) values. The expected values are

$$n_{Z,k}^* = c_{\text{dt}} q_z^2 p_k Q_k N, \quad n_{X,k}^* = c_{\text{dt}} (1 - q_z)^2 p_k Q_k N, \quad (\text{A54})$$

where Q_k is the total gain related to intensity k and c_{dt} is a factor depending on the dead time τ_{dt} of the detector. In

particular, we have

$$Q_k = 1 - e^{-\mu_k \eta} + p_{\text{dc}},$$

$$c_{\text{dt}} = \frac{1}{1 + (r_s \tau_{\text{dt}} - 1) \sum_k p_k Q_k} \simeq \frac{1}{1 + r_s \tau_{\text{dt}} \sum_k p_k Q_k}, \quad (\text{A55})$$

where p_{dc} is the dark count probability and r_s is the repetition rate of the source (in Hz or uses/s). Note that we model c_{dt} following Ref. [16], so we have the scaling $c_{\text{dt}} \simeq \tau_{\text{dt}}^{-1}$. A different modeling in Ref. [38] leads to the more optimistic scaling $c_{\text{dt}} \simeq \tau_{\text{dt}}^{-1/2}$. For reference, this other model provides the following expression for the Z basis:

$$c_{\text{dt}} = \frac{\sqrt{4w_{\text{dt}} + 1} - 1}{2w_{\text{dt}}},$$

$$w_{\text{dt}} := r_s q_z^2 \tau_{\text{dt}} \sum_k p_k Q_k. \quad (\text{A56})$$

(f) In the finite-size regime, we can use Hoeffding's inequality to bound the difference between the observed data and the expected asymptotic data [49]. This means that, up to a probability $1 - 2\varepsilon_Z$, we have

$$|n_{Z,k} - n_{Z,k}^*| \leq \delta(n_Z, \varepsilon_Z).$$

This result is used in the final key rate formula to bound the worst-case scenario for the expected values from the knowledge of the observed values.

(g) In an experiment, Alice and Bob know exactly the event numbers $n_{Z,k}$ and $n_{X,k}$ (via analysis of their data). In a simulation, we directly set $n_{Z,k} = n_{Z,k}^*$ and $n_{X,k} = n_{X,k}^*$. Then, we compute n_Z^* and n_X^* following Eq. (A52).

(5) *Raw key*:

(a) The set of Z -sifted instances $\mathcal{Z}$ is entirely used for key generation (all the instances are used, no matter if they correspond to a signal or a decoy). The associated bit strings $\mathcal{Z}^A$ and $\mathcal{Z}^B$ represent Alice's and Bob's raw keys, with length n_Z , that enter the postprocessing described below in point (7). The value n_Z represents the effective EC block size. Its value must be on the order of 10^5 at least. Its expected value is

$$n_Z^* = c_{\text{dt}} q_z^2 \left(\sum_k p_k Q_k \right) N. \quad (\text{A57})$$

(b) The set of X -sifted instances $\mathcal{X}$ is only used for parameter estimation via the associated subsets $\mathcal{X}_k^A$ and $\mathcal{X}_k^B$.

(c) Differently from a two-decoy protocol, Alice and Bob also need to extract an error rate from the Z -sifted instances, but this will be done after EC.

(6) *Parameter estimation (X errors and QBER)*:

(a) In the first stage of PE, Alice and Bob compute the QBER from X instances.

(b) By declaring the sets $\mathcal{X}_k^A$ and $\mathcal{X}_k^B$, Alice and Bob can compute the number of bit errors in the X -sifted instances for the various intensities, denoted as $m_{X,k}$. The total number of bit errors in X is

$$m_X = \sum_k m_{X,k}. \quad (\text{A58})$$

(c) In an experimental implementation, Alice and Bob analyze their data to extract exact values for the observed

bit-error numbers $m_{X,k}$ and m_X . These will provide them with the observed QBER $E = m_X/n_X$.

(d) The error numbers in the X basis have expected values $m_{X,k}^*$ that are determined by

$$\begin{aligned} m_{X,k}^* &= c_{\text{dt}}(1 - q_z)^2 p_k P_k^{\text{err}}, \\ P_k^{\text{err}} &= (1 - e^{-\mu_k \eta}) e_{\text{det}} + p_{\text{dc}}/2, \end{aligned} \quad (\text{A59})$$

in terms of the error probability P_k^{err} , which depends on the dark counts p_{dc} and the misalignment error e_{det} . Up to a probability $1 - 2\varepsilon_X$, we have

$$|m_{X,k} - m_{X,k}^*| \leq \delta(m_X, \varepsilon_X'). \quad (\text{A60})$$

(e) In a simulation, we set $m_{X,k} = m_{X,k}^*$ and $m_X = m_X^*$. The total QBER will be given by

$$E = m_X^*/n_X^* = \frac{\sum_k p_k P_k^{\text{err}}}{\sum_k p_k Q_k}. \quad (\text{A61})$$

(f) In an alternative scenario where we know the QBER (e.g., this may occur in a simulation), we assume the QBER as an input parameter $E = E_{\text{in}}$. First, we check that

$$E_{\text{in}} \geq \frac{\sum_k p_k P_k^{\text{err}}}{\sum_k p_k Q_k}. \quad (\text{A62})$$

If the test is successful, we compute all the error numbers via $m_{X,k} = E_{\text{in}} n_{X,k}^*$ [and the total m_X via Eq. (A58)]. Note that this scenario can be made more precise by considering the various QBERs associated with the different intensities, but this is not relevant for the scope of the present work.

(7) *Error correction*:

(a) Alice and Bob proceed with the data processing of the raw keys Z^A and Z^B , each of length n_Z .

(b) They first perform error correction so that Bob extracts a corrected key string $Z^B \rightarrow Z_{\text{cor}}^B$.

(c) They then perform the verification step by comparing hashes. If successful, Alice's and Bob's verified key strings are equal $Z^A = Z_{\text{cor}}^B$ up to an error ε_{cor} .

(d) Overall, the leakage $\text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}}$ is quantified by

$$\text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}} = n_Z f_{\text{EC}}(E) + \log_2(2/\varepsilon_{\text{cor}}), \quad (\text{A63})$$

in terms of ε_{cor} and the QBER E , for some reconciliation efficiency f_{EC} .

(e) EC has an overall success probability p_{EC} whose value depends on the EC method adopted.

(f) In a theoretical analysis or a simulation, the values of f_{EC} and p_{EC} can be given as input.

(8) *Parameter estimation (Z errors)*:

(a) Alice and Bob need to compute the error numbers $m_{B,k}$ in the two bases $B = Z, X$ for the various intensities k . For the X basis, the computation of $m_{X,k}$ derives from comparing their strings $\mathcal{X}_k^A$ and $\mathcal{X}_k^B$. This step was already done above for deriving the QBER [see point (6)].

(b) For the Z basis, Bob can derive the error numbers $m_{Z,k}$ by comparing the initial strings Z_k^B with the error-corrected ones Z_{cor}^B . Then, $m_Z = \sum_k m_{Z,k}$, since $Z^B = \cup_k Z_k^B$ and $Z_{\text{cor}}^B = \cup_k Z_{k,\text{cor}}^B$.

(c) The error numbers in the Z basis have expected values $m_{Z,k}^*$. Up to a probability $1 - 2\varepsilon_Z'$, we have

$$|m_{Z,k} - m_{Z,k}^*| \leq \delta(m_Z, \varepsilon_Z'). \quad (\text{A64})$$

(d) Similar to the X basis, for the Z basis we may write

$$\begin{aligned} m_{Z,k}^* &= c_{\text{dt}} q_z^2 p_k P_k^{\text{err}} N, \\ P_k^{\text{err}} &= (1 - e^{-\mu_k \eta}) e_{\text{det}} + p_{\text{dc}}/2, \end{aligned} \quad (\text{A65})$$

with e_{det} being the misalignment error (wrong detector click) and p_{dc} being the dark count rate.

(e) In our theoretical simulation, we set $m_{Z,k} = m_{Z,k}^*$. Alternatively, if the QBER is passed as an input parameter $E = E_{\text{in}}$, then we compute $m_{Z,k} = E_{\text{in}} n_{Z,k}^*$ (and the corresponding total m_Z).

(9) *Parameter estimation (decoy bounds)*:

(a) Let us use a single epsilon parameter $\tilde{\varepsilon}$ for the decoy bounds (in general, they may have different epsilons).

(b) For $B = Z, X$ and generic intensity k , we consider $n_{B,k}^{\pm}$ and $m_{B,k}^{\pm}$ as follows:

$$\begin{aligned} n_{B,k}^{\pm} &:= \frac{e^k}{p_k} [n_{B,k} \pm \delta(n_B, \tilde{\varepsilon})], \\ m_{B,k}^{\pm} &:= \frac{e^k}{p_k} [m_{B,k} \pm \delta(m_B, \tilde{\varepsilon})]. \end{aligned} \quad (\text{A66})$$

(c) For one-decoy BB84, we need to upper bound the number of vacuum events in Z and X , denoted as $s_{Z,0}$ and $s_{X,0}$. For $B = Z, X$, we may write

$$s_{B,0} \leq s_{B,0}^U := 2[\tau_0 \min(m_{B,k}^+) + \delta(n_B, \tilde{\varepsilon})], \quad (\text{A67})$$

where we choose k so that the bound is the tightest possible [see also discussion around Eq. (98) of Ref. [39]].

(d) As for a two-decoy protocol, we then need to perform the following:

(i) Lower bound the number of vacuum events in Z , denoted as $s_{Z,0}$. We write

$$s_{Z,0} \geq s_{Z,0}^L := \frac{\tau_0}{\mu_1 - \mu_2} (\mu_1 n_{Z,\mu_2}^- - \mu_2 n_{Z,\mu_1}^+). \quad (\text{A68})$$

(ii) Lower bound the number of single-photon events in Z and X , denoted as $s_{Z,1}$ and $s_{X,1}$. For $B = Z, X$, we write

$$\begin{aligned} s_{B,1} \geq s_{B,1}^L &:= \frac{\tau_1 \mu_1}{\mu_2 (\mu_1 - \mu_2)} \\ &\times \left(n_{B,\mu_2}^- - \frac{\mu_2^2}{\mu_1^2} n_{B,\mu_1}^+ - \frac{\mu_1^2 - \mu_2^2}{\mu_1^2} \frac{s_{B,0}^U}{\tau_0} \right). \end{aligned} \quad (\text{A69})$$

(iii) Upper bound the number of bit errors associated with the single-photon events in X , denoted as $v_{X,1}$. We write

$$v_{X,1} \leq v_{X,1}^U := \frac{\tau_1}{\mu_1 - \mu_2} (m_{X,\mu_1}^+ - m_{X,\mu_2}^-). \quad (\text{A70})$$

(e) Using the upper bound above together with the lower bounds on the single-photon events, we may upper bound the phase error rate $\phi_Z \leq \phi_Z^U$ as in

$$\begin{aligned} \phi_Z^U &= \frac{v_{X,1}^U}{s_{X,1}^L} + \gamma(\tilde{\varepsilon}, s_{Z,1}^L, s_{X,1}^L), \\ \gamma(a, b, c) &:= \sqrt{\frac{(b+c)(c+1)}{bc^2} \ln(1/a)}. \end{aligned} \quad (\text{A71})$$

Remark A8. Note that the phase error rate above involves a bound on the single-photon QBER in the X basis $v_{X,1}/s_{X,1}$

plus a gamma function. This function can take different forms. The one above is a rigorous formula (see Methods of Ref. [50]), representing an effective bound, based on the Serfling inequality [51]. An alternative, but nonrigorous, formula is based on an interpolation, instead of a bound [52].

(10) *Privacy amplification, epsilon security, and key rate:*

(a) Alice and Bob perform privacy amplification by compressing their corrected strings in two identical keys (S_A, S_B) with $|S_A| = |S_B| = l$ secret bits. This operation is performed up to an epsilon secrecy ε_{sec} .

(b) The protocol has total epsilon security

$$\varepsilon = \varepsilon_{\text{cor}} + \varepsilon_{\text{sec}} \quad (\text{A72})$$

and secret key rate (bits/s)

$$R = r_s l / N, \quad (\text{A73})$$

where the secret key length l is upper bounded as follows:

$$l \leq s_{Z,0}^L + s_{Z,1}^L [1 - h(\phi_Z^U)] - \text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}} - 6 \log_2(1/\tilde{\varepsilon}), \quad (\text{A74})$$

with

$$\tilde{\varepsilon} := \varepsilon_{\text{sec}}/19. \quad (\text{A75})$$

Remark A9. Note that Eqs. (A73) and (A74) represent a revision of the rate in Ref. [38]. Another revision is present in Ref. [39], which considers $\tilde{\varepsilon} := \varepsilon_{\text{sec}}/15$ and

$$l \leq s_{Z,0}^L + s_{Z,1}^L [1 - h(\phi_Z^U)] - \text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}} - 4 \log_2 \left(\frac{1}{\sqrt[4]{2\tilde{\varepsilon}}} \right). \quad (\text{A76})$$

(11) *Extension to QKD session:* In a realistic implementation, Alice and Bob exchange a number n_{bks} of blocks (each of size N). Each block is generated with an EC success probability p_{EC} and epsilon security ε . Overall, the key rate associated with the session will be

$$R \longrightarrow R_{\text{ses}} = p_{\text{EC}} R \quad (\text{A77})$$

and the total epsilon security of the session will be

$$\varepsilon \longrightarrow \varepsilon_{\text{ses}} = n_{\text{bks}} \varepsilon. \quad (\text{A78})$$

As for the one-photon BB84 protocol, we may consider block-by-block or session-wide PE. However, for our QKD network studies of Sec. IV, we only consider block-by-block PE. Finally, in Table II, we present the typical set of parameters associated with the one-decoy protocols, along with possible values.

APPENDIX B: COMPOSABLY SECURE CV-QKD

We consider Gaussian-modulated coherent-state protocols where the receiver implements homodyne detection [19] or heterodyne detection [20]. We treat the two cases compactly as a single CV-QKD protocol with two possible types of detection. We use the improved secret key rate developed in Ref. [53], introducing a suitable simplification of the epsilon parameters. In Appendix B 1, we describe the protocol and its composable key rate. In Appendix B 2, we discuss how to compute the asymptotic key rate, which is part of the composable formula.

TABLE II. Parameters for the one-decoy BB84 protocols. We also show some possible values. More specifically, these are the values considered in our network numerical investigations described in Sec. IV. Note that $\varepsilon = 2 \times 10^{-10}$.

Parameter	Value	Notes
n_{bks}	50	Number of blocks
N	10^{10}	Block size
r_s	100 MHz	Repetition rate
q_x	1/16	X-basis probability
μ_1	0.5	First intensity (signal)
μ_2	0.1	Second intensity (decoy)
p_{μ_1}	0.9	Signal probability
p_{μ_2}	0.1	Decoy probability
α_L	0.2 dB/km	Fiber loss rate
d	Variable (km)	Fiber distance
E	Variable	QBER
η_{eff}	0.7	Detector efficiency
e_{det}	2×10^{-2}	Misalignment error
p_{dc}	10^{-8}	Dark count probability
τ_{dt}	100 ns	Detector dead time
p_{EC}	0.95	EC success probability
f_{EC}	1.16	Reconciliation efficiency
ε_{cor}	10^{-10}	Correctness
ε_{sec}	10^{-10}	Secrecy

1. CV-QKD protocols: Description and key rate

The schematic steps of the protocol are the following.

(1) *Encoding:* Alice has a bosonic mode A with quadratures $\hat{q}$ and $\hat{p}$. This is prepared in a coherent state $|\alpha\rangle$ with amplitude $\alpha = (q + ip)/2$, where q and p are the mean values of the two quadratures. We adopt the following notation: (a) Alice's generic variable x represents either q or p ; and (b) Alice's vector variable $\mathbf{x}$ is x in the homodyne protocol and (q, p) in the heterodyne protocol. The generic variable x is chosen according to a zero-mean Gaussian distribution with variance $\sigma_x^2 := \mu - 1 \geq 0$. Here, μ is the total quantum variance, combining the classical variance σ_x^2 with the unit vacuum noise.

(2) *Channel:* The coherent state is transmitted through an optical fiber of length d , which can be modeled as a thermal-loss channel with transmissivity $\eta_{\text{ch}} := T$ and thermal noise $\omega = 2\bar{n} + 1$, where $\bar{n}$ represents the mean number of photons injected by the environment [see Eq. (A2) for the relation between channel transmissivity, fiber loss rate α_L , and fiber distance d]. This Gaussian channel can be dilated into a beam splitter with transmissivity T mixing mode A with an environmental mode E . In the worst-case scenario, the latter mode is part of a two-mode squeezed vacuum state used by Eve to perform a collective entangling-cloner attack. It is important to recall that channel noise ω is often represented via the “excess noise,” defined as

$$\xi := \frac{1 - T}{T}(\omega - 1). \quad (\text{B1})$$

(3) *Decoding:* At the receiver side, Bob can have homodyne (switching between the measurement of the two quadratures $\hat{q}$ and $\hat{p}$) or heterodyne (simultaneously measuring both $\hat{q}$ and $\hat{p}$ with extra vacuum noise). Let us call q' and p' Bob's output values. We then adopt the same compact

notation adopted for Alice: (a) Bob's generic variable y can be either q' or p' ; and (b) Bob's vector variable $\mathbf{y}$ is y for the homodyne detection and (q', p') for heterodyne. We also compactly describe the two detection strategies by defining the noise variance $V_0 = 1$ for homodyne, while $V_0 = 2$ for heterodyne.

Assume that Bob's detector has efficiency η_{eff} and electronic noise v_{el} . These are assumed to be trusted quantities, since they are calibrated and well controlled in typical CV-QKD experiments. Given Alice's generic input variable x , Bob's generic output variable will be

$$y = \sqrt{\eta}x + z, \quad \eta = T\eta_{\text{eff}}, \quad (\text{B2})$$

where z is a Gaussian noise variable with zero mean and variance

$$\sigma_z^2 = V_0 + v_{\text{el}} + \eta\xi := \sigma_z^2(V_0). \quad (\text{B3})$$

(4) *Asymptotic rate*: The asymptotic key rate of the protocol is based on an infinite number of exchanged signal states and the perfect knowledge of the channel parameters T and ξ . In terms of secret bits per channel use, it takes the form

$$R_\infty = \beta I(\mathbf{x} : \mathbf{y}) - \chi(E : \mathbf{y}), \quad (\text{B4})$$

where $\beta \leq 1$ is the reconciliation efficiency, the first term $I(\mathbf{x} : \mathbf{y}) = I(T, \xi)$ is Alice and Bob's mutual information, and the second term $\chi(E : \mathbf{y}) = \chi(T, \xi)$ is Eve's Holevo information on Bob's variable. See Appendix B 2 on how to compute $R_\infty = R_\infty(T, \xi)$ as a function of T and ξ .

(5) *Parameter estimation (channel parameters)*: In the finite-size regime, Alice sends a block of N coherent states, whose m are selected for PE, so that $n = N - m$ are used for KG [54]. We may define the parameter estimation ratio as $r_{\text{PE}} := m/N$. During PE, Alice and Bob build unbiased estimators of the channel parameters, $\hat{T} \simeq T$ and $\hat{\xi} \simeq \xi$. They also construct worst-case values $\tilde{T}$ and $\tilde{\xi}$. Up to an epsilon error ε_{PE} , the true transmissivity satisfies $T \geq \tilde{T}$, where the worst-case value is theoretically computed as

$$\tilde{T} := \hat{T} - w_{\text{PE}}\sigma_{\hat{T}} \simeq T - w_{\text{PE}}\sigma_T, \quad (\text{B5})$$

where [53]

$$\sigma_{\hat{T}} \simeq \sigma_T := \frac{2T}{\sqrt{V_0 m}} \sqrt{\frac{\xi + \frac{V_0 + v_{\text{el}}}{\eta_{\text{eff}} T}}{\sigma_x^2}} \quad (\text{B6})$$

and w_{PE} is a penalty factor depending on ε_{PE} . One possible choice is to use $w_{\text{PE}} = \sqrt{2}\text{erf}^{-1}(1 - 2\varepsilon_{\text{PE}})$, with erf^{-1} being the inverse of the error function. Alternatively, we can use an upper bound, as given by $w_{\text{PE}} = \sqrt{2\ln(\varepsilon_{\text{PE}}^{-1})}$. The latter has the advantage of being well defined for small values of ε_{PE} (below 10^{-14}). We use this second option for the numerical calculations in our paper. Similarly, up to ε_{PE} , the true excess noise satisfies $\xi \leq \tilde{\xi}$ where the worst-case value is theoretically computed as [53]

$$\tilde{\xi} \simeq \frac{T}{\tilde{T}}\xi + w_{\text{PE}}\sqrt{\frac{2}{V_0 m} \frac{\eta_{\text{eff}} T \xi + V_0 + v_{\text{el}}}{\eta_{\text{eff}} \tilde{T}}}. \quad (\text{B7})$$

(6) *Analog-to-digital conversion*: After the first phase of PE, Alice and Bob perform ADC of their continuous variables

$\mathbf{x}$ and $\mathbf{y}$ into discrete-alphabet variables, k and l . In particular, for the heterodyne protocol, we adopt a concatenation process:

$$\begin{aligned} \mathbf{x} &= (q, p) \xrightarrow{\text{ADC}} (k_q, k_p) \rightarrow k, \\ \mathbf{y} &= (q', p') \xrightarrow{\text{ADC}} (l_q, l_p) \rightarrow l. \end{aligned} \quad (\text{B8})$$

This means that, after conversion, each letter is equivalent to a bit string of length $d = V_0 D$, where D is the digitalization per quadrature and V_0 accounts for the single or double contribution of the quadrature components, depending on the use of homodyne or heterodyne.

(7) *Parameter estimation (key entropy)*: After ADC, the parties perform a second phase of PE (which does not require any public disclosure of data). It is crucial to estimate Bob's Shannon entropy $H(l)$ from the n KG points. Bob calculates the frequency f_l of each value l and builds the estimator

$$\hat{H}(l) = - \sum_{l=0}^{2^d-1} f_l \log_2 f_l. \quad (\text{B9})$$

Up to an error probability ε_{ent} , this estimator bounds the actual value as

$$H(l) \geq \hat{H}(l) - \delta_{\text{ent}}, \quad \delta_{\text{ent}} = (\log_2 n) \sqrt{\frac{2 \ln(2/\varepsilon_{\text{ent}})}{n}}. \quad (\text{B10})$$

(8) *Error correction*: In the EC step, Alice and Bob correct and verify their strings. Specifically, they work in reverse reconciliation, meaning that it is Alice correcting her string to match Bob's. In the standard approach with LDPC codes, Bob reveals $\text{leak}_{\text{EC}}^{\text{syn}}$ bits of information to help Alice compute her guess (see Refs. [55,56] for more details of the data processing). In the subsequent verification step, Bob sends $[-\log_2 \varepsilon_{\text{cor}}]$ bits to Alice, so we have the following total number of bits disclosed in the process:

$$\text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}} = \text{leak}_{\text{EC}}^{\text{syn}} + \lceil \log_2 (1/\varepsilon_{\text{cor}}) \rceil. \quad (\text{B11})$$

The verification step is successful with probability p_{EC} , and the correctness parameter ε_{cor} bounds the probability that EC is successful and Alice's and Bob's strings are different. The EC process defines the so-called "reconciliation parameter" $\beta \in [0, 1]$ as follows:

$$\hat{H}(l) - n^{-1} \text{leak}_{\text{EC}}^{\varepsilon_{\text{cor}}} = \beta I(\hat{T}, \hat{\xi}), \quad (\text{B12})$$

where $I(\hat{T}, \hat{\xi})$ is Alice's and Bob's mutual information computed over the estimated channel parameters.

We can write the finite-size key rate, which accounts for the imperfections due to PE. The asymptotic key rate term R_∞ will be modified as follows:

$$R_\infty(T, \xi) \rightarrow \frac{n}{N} R_m^{\text{PE}}, \quad R_m^{\text{PE}} := \beta I(\hat{T}, \hat{\xi}) - \chi(\tilde{T}, \tilde{\xi}). \quad (\text{B13})$$

(9) *Privacy amplification*: After EC, Alice and Bob perform PA where they randomly choose a 2-universal hash function to compress their nd -long binary error-corrected and verified binary strings [57]. In this way, they reduce their strings into a key pair $\{\hat{s}, s\}$ of $s_n < nd$ secret random bits. The process of PA is affected by an overall error probability ε_{sec} that quantifies how far the final keys are from being completely decoupled from Eve (this is measured in terms of trace distance). The epsilon secrecy can be written as the

sum of smoothing and hashing parameters, i.e., $\varepsilon_{\text{sec}} = \varepsilon_s + \varepsilon_h$. These parameters are introduced by the application of the “leftover hash bound,” which bounds the key length s_n in terms of the smooth-min entropy $H_{\min}^{\varepsilon_s}$. In turn, $H_{\min}^{\varepsilon_s}$ can be replaced by the simpler von Neumann entropy by applying the AEP, which reduces the formulas to a single-copy bound for s_n (see Ref. [53] for more details).

Overall, these steps lead to the bound [53]

$$s_n \leq nR_m^{\text{PE}} - n\delta_{\text{ent}} - \sqrt{n}\Delta_{\text{AEP}} + \log_2(2\varepsilon_h^2\varepsilon_{\text{cor}}),$$

$$\Delta_{\text{AEP}} \simeq 4\log_2(2^{d/2} + 2)\sqrt{\log_2(2/\varepsilon_s^2)}. \quad (\text{B14})$$

(10) *Secret key rate*: Assuming successful EC, the secret key rate is given by the ratio s_n/N (bits per channel use). Including the repetition rate r_s , we get the rate in terms of bits per second, which is

$$R_{\text{block}} = \frac{r_s s_n}{N}, \quad (\text{B15})$$

with associated epsilon security $\varepsilon_{\text{block}} = \varepsilon_{\text{cor}} + \varepsilon_{\text{sec}} + p_{\text{EC}}(\varepsilon_{\text{ent}} + 2\varepsilon_{\text{PE}})$. More simply, we consider the upper bound

$$\varepsilon_{\text{block}} = \varepsilon_{\text{cor}} + \varepsilon_{\text{sec}} + \varepsilon_{\text{ent}} + 2\varepsilon_{\text{PE}}. \quad (\text{B16})$$

As already discussed, the block rate R_{block} represents the conditional key rate of the protocol. The unconditional key rate will be given by the average

$$\bar{R}_{\text{block}} = p_{\text{EC}} R_{\text{block}}. \quad (\text{B17})$$

We can write the rate in a simpler form by setting

$$\varepsilon_{\text{ent}} = \varepsilon_{\text{PE}}, \quad \varepsilon_s = \varepsilon_h = \varepsilon_{\text{sec}}/2, \quad (\text{B18})$$

so the total epsilon security is simply

$$\varepsilon_{\text{block}} = \varepsilon_{\text{cor}} + \varepsilon_{\text{sec}} + 3\varepsilon_{\text{PE}} \quad (\text{B19})$$

and the block key rate takes the form

$$R_{\text{block}} \leq R_{\text{UB}} = \frac{r_s}{N}(nR_m^{\text{PE}} - F), \quad (\text{B20})$$

where, in the formula above, F accounts for finite-size terms as follows:

$$F = (\sqrt{n}\log_2 n)\sqrt{2\ln(2/\varepsilon_{\text{PE}})} + 4\sqrt{n}\log_2(\sqrt{2^{V_0}} + 2)\sqrt{\log_2(8/\varepsilon_{\text{sec}}^2)} - \log_2(\varepsilon_{\text{sec}}^2\varepsilon_{\text{cor}}/2). \quad (\text{B21})$$

(11) *QKD session*: The channel parameters may remain constant over a relatively long time in a fiber-based scenario. The theory can therefore be extended to a session of n_{bks} blocks, each of size N . The fraction of blocks that survive EC is given by the success probability p_{EC} . The process of PE, EC, and PA is performed block by block, with the final keys being concatenated. The average key rate of the session (in bits/s) is equal to the unconditional key rate of the protocol, i.e.,

$$R_{\text{ses}} = p_{\text{EC}} R_{\text{block}}, \quad (\text{B22})$$

and the overall security of the session is

$$\varepsilon_{\text{ses}} = n_{\text{bks}}\varepsilon_{\text{block}}. \quad (\text{B23})$$

TABLE III. Parameters for CV-QKD. We also show typical values. These are the values adopted in our network numerical investigations with the homodyne-based protocol described in Sec. IV. Note that $\varepsilon = 3 \times 10^{-10}$.

Parameter	Value	Notes
n_{bks}	50	Number of blocks
N	10^6	Block size
r_s	100 MHz	Repetition rate
σ_x^2	40	Signal modulation
α_L	0.2 dB/km	Fiber loss rate
d	Variable (km)	Fiber distance
ξ	0.01	Excess noise
η_{eff}	0.8	Detector efficiency
v_{el}	0.01	Electronic noise
r_{PE}	0.3	PE ratio
D	7 bits	Quadrature digitalization
p_{EC}	0.95	EC success probability
β	0.95	Reconciliation efficiency
ε_{PE}	10^{-10}	PE error
ε_{cor}	10^{-10}	Correctness
ε_{sec}	10^{-10}	Secrecy

In the alternative PE approach, we perform the estimation of the channel parameters in a single shot on the entire collection of $m' := n_{\text{bks}}m$ points (note that p_{EC} does not appear here because we also use the PE points from the discarded blocks). This approach leads to improved worst-case values for the channel parameters, $\tilde{T}$ and $\tilde{\xi}$. Alice and Bob will implement EC and PA block by block using these improved statistics. The session key rate is again given by Eq. (B22), but where the block rate R_{block} is computed using $R_{m'}^{\text{PE}}$, i.e., the m' -point channel estimators and worst-case values. Note that we may further improve the PE process if we also improve the estimator of the key entropy $\bar{H}$ by computing it over $n_{\text{bks}}n$ points. This will reduce the penalty δ_{ent} appearing in Eq. (B10) and later in the rate. In our main text, we do not consider this second aspect for the sake of simplicity, but the treatment can be easily extended.

In Sec. II, we consider both block-by-block and session-wide PE. However, in Sec. III, where we perform our QKD network studies, we only consider block-by-block PE. Finally, in Table III, we show the typical set of parameters associated with CV-QKD protocols, also showing possible values.

2. Asymptotic key rate for CV-QKD protocols

Here, we present the detailed calculation for the asymptotic key rate of a Gaussian-modulated coherent-state protocol with homodyne or heterodyne detection.

Assuming that a very large number of states are exchanged in the protocol, its security is well approximated by the asymptotic key rate, which is given by the formula

$$R_{\infty} = \beta I(\mathbf{x} : \mathbf{y}) - \chi(E : \mathbf{y}), \quad (\text{B24})$$

where β is the reconciliation efficiency (heuristically set to values ≤ 1), $I(\mathbf{x} : \mathbf{y})$ is Alice and Bob's mutual information, and $\chi(E : \mathbf{y})$ is Eve's Holevo information on Bob's variable

y. In particular, we have

$$I(\mathbf{x} : \mathbf{y}) = \frac{V_0}{2} \log_2 \left[1 + \frac{\eta \sigma_x^2}{\sigma_z^2(V_0)} \right] \quad (\text{B25})$$

and

$$\chi(E : \mathbf{y}) = h(v_+) + h(v_-) - h(\tilde{v}_+) - h(\tilde{v}_-). \quad (\text{B26})$$

In the expression above, we use the entropic function

$$h(v) := \frac{v+1}{2} \log_2 \frac{v+1}{2} - \frac{v-1}{2} \log_2 \frac{v-1}{2}, \quad (\text{B27})$$

which is computed over the symplectic eigenvalues, v_+ and v_- , of Eve's covariance matrix (CM) $\mathbf{V}_E$, and the symplectic eigenvalues, $\tilde{v}_+$ and $\tilde{v}_-$, of Eve's conditional CM $\mathbf{V}_{E|\mathbf{y}}(V_0)$, i.e., conditioned on Bob's outcome.

Let us define the following auxiliary matrices:

$$\mathbf{I} = \text{diag}(1, 1), \quad \mathbf{Z} = \text{diag}(1, -1), \quad \mathbf{\Pi} = \text{diag}(1, 0). \quad (\text{B28})$$

Then, Eve's CM can be written as

$$\mathbf{V}_E = \begin{pmatrix} \omega \mathbf{I} & \psi \mathbf{Z} \\ \psi \mathbf{Z} & \phi \mathbf{I} \end{pmatrix}, \quad (\text{B29})$$

where

$$\psi = \sqrt{T(\omega^2 - 1)}, \quad \phi = T\omega + (1 - T)\mu. \quad (\text{B30})$$

For the homodyne protocol, Eve's conditional CM is given by

$$\mathbf{V}_{E|\mathbf{y}}(1) = \mathbf{V}_E - b^{-1} \begin{pmatrix} \gamma^2 \mathbf{\Pi} & \gamma \theta \mathbf{\Pi} \\ \gamma \theta \mathbf{\Pi} & \theta^2 \mathbf{\Pi} \end{pmatrix}, \quad (\text{B31})$$

where

$$\begin{aligned} b &= \eta(\mu - 1 + \xi) + 1 + v_{\text{el}}, \\ \gamma &= \sqrt{\eta_{\text{eff}}(1 - T)(\omega^2 - 1)}, \\ \theta &= \sqrt{\eta(1 - T)(\omega - \mu)}. \end{aligned} \quad (\text{B32})$$

For the heterodyne protocol, Eve's conditional CM is instead given by

$$\mathbf{V}_{E|\mathbf{y}}(2) = \mathbf{V}_E - (b + 1)^{-1} \begin{pmatrix} \gamma^2 \mathbf{I} & \gamma \theta \mathbf{Z} \\ \gamma \theta \mathbf{Z} & \theta^2 \mathbf{I} \end{pmatrix}. \quad (\text{B33})$$

The analysis of the asymptotic security is done by studying the key rate R_∞ as a function of the transmissivity T and the excess noise ξ , i.e., considering the function

$$R_\infty(T, \xi) = \beta I(T, \xi) - \chi(T, \xi). \quad (\text{B34})$$

In particular, by setting $R_\infty = 0$, one can also derive the security threshold of the protocol in terms of maximum tolerable excess noise versus transmissivity $\xi = \xi(T)$. In the asymptotic scenario, channel's parameters are assumed to be perfectly known, because Alice and Bob can sacrifice a large number of their exchanged states to perfectly estimate these quantities.

APPENDIX C: ETE RATE FROM LINK RATES

An ETE key is generated by successively swapping keys along a path using one-time-pad sessions. When multipath routing is available, the ETE keys obtained from parallel paths

are concatenated, resulting in a larger composite key. Key swapping and composition will occur at every time T agreed by all the trusted parties. Under stationary conditions, the key-swapping interval T can be assumed to be much longer than the typical time required to complete a QKD session on an individual link ($T \gg n_{\text{bks}} N / r_s$). Consequently, many sessions are performed and many key blocks are generated within each interval. Under this assumption, truncation effects arising from unequal session or block lengths across different links can be neglected, and the session end points can be treated as synchronized across all links. For this reason, the (maximum) length of the key string generated on each link can be estimated as the product of the corresponding link key rate (in bits per second) and the total duration T .

Having established this general context, we now discuss how the ETE key rate between two end points is computed from the key rates of the individual links along the path(s) connecting them. The resulting ETE key generation rate is not always obvious to determine: It may be exact or expressed as an average, depending on whether the protocols operating on the individual links are deterministic or probabilistic.

1. Deterministic link protocols

For simplicity, consider two adjacent links $e_1 = (\mathbf{a}, \mathbf{c})$ and $e_2 = (\mathbf{c}, \mathbf{b})$ with associated key rates $R_1 = R_{e_1}$ and $R_2 = R_{e_2}$ (the reasoning can be implicitly extended to multiple links). Then, assume that their QKD protocols, $\mathbb{P}_1$ and $\mathbb{P}_2$, generating those rates, have unit success probabilities, i.e., $p_1 = p_2 = 1$. As a result, the composite link $e = (\mathbf{a}, \mathbf{b})$ will have an associated key rate (ETE rate), which is given by the minimum value $R = \min\{R_1, R_2\}$. In fact, under stationary conditions where the two links continuously generate keys at those rates, the key swapping session will occur at every (large) time T . The swapping will consume two key strings of lengths $K_1 = TR_1$ and $K_2 = TR_2$ to generate a single ETE key string $K = \min\{K_1, K_2\}$. Equivalently, the final key is generated with a rate $R = K/T$.

2. Probabilistic link protocols

The situation is different if the protocols performed on the links may have nonunit success probabilities. For a generic link labeled by i , let us assume that its protocol $\mathbb{P}_i$ generates a block of size N_i with block rate R_i and probability $p_i \leq 1$ (in the main text, these parameters correspond to R_{block} and p_{EC}). After time T , the protocol will have generated a key string of length $K_i \leq K_i^{\text{max}} = TR_i$. Let us call n_i the stochastic variable for the number of successfully corrected blocks. We have

$$0 \leq n_i \leq n_i^{\text{max}} \simeq \frac{K_i^{\text{max}}}{N_i}, \quad (\text{C1})$$

with expected value $\mathbb{E}(n_i) = p_i n_i^{\text{max}}$. We want to find the probability distribution $P(n_i)$ of having n_i successful blocks at time T . This is independent for each link, and is clearly a binomial $n_i \sim \text{Bin}(n_i^{\text{max}}, p_i)$, i.e.,

$$P(n_i) = \binom{n_i^{\text{max}}}{n_i} p_i^{n_i} (1 - p_i)^{n_i^{\text{max}} - n_i}. \quad (\text{C2})$$

Given n_i successful blocks, we have a key string of length $K_i = n_i N_i$. Thus, for two links, with key strings of lengths K_1 and K_2 , the composition of the keys will provide a string of length $K = \min\{K_1, K_2\}$, which is itself a stochastic variable $0 \leq K \leq \min\{K_1^{\max}, K_2^{\max}\}$.

The central point is now to study the mean value $\mathbb{E}(K)$, for which we can provide bounds. An upper bound can be computed by applying Jensen's inequality, which is $\mathbb{E}[f(x, y)] \leq f[\mathbb{E}(x), \mathbb{E}(y)]$ for a concave f . In fact, this can be applied to the (concave) minimum function

$$\min\{K_1, K_2\} = \frac{K_1 + K_2 - |K_1 - K_2|}{2}, \quad (\text{C3})$$

which leads to the upper bound

$$\mathbb{E}(K) \leq \min\{\mathbb{E}(K_1), \mathbb{E}(K_2)\}, \quad (\text{C4})$$

with

$$\mathbb{E}(K_i) = p_i n_i^{\max} N_i = p_i K_i^{\max}. \quad (\text{C5})$$

Dividing Eq. (C4) by the time T , we obtain

$$\mathbb{E}(R) \leq \min\{\mathbb{E}(R_1), \mathbb{E}(R_2)\}, \quad (\text{C6})$$

where $\mathbb{E}(R_i) = p_i R_i$.

It is important to note that the upper bound in Eq. (C6) represents a mean-field approximation

$$\mathbb{E}(R) \simeq \min\{\mathbb{E}(R_1), \mathbb{E}(R_2)\}. \quad (\text{C7})$$

Because we are in stationary conditions, we can always choose T large enough to have $\mathbb{E}(n_i) \gg 1$, so the approximation in Eq. (C7) is tight. For this reason, in our derivations, we can compute the ETE average rate as given by the bottleneck value computed from the average rates of the links along the path.

Remark C1. Note that we may also start from Eq. (C3) to write a lower bound. In fact, let us set $W := K_1 - K_2$, so we can apply the Cauchy-Schwarz inequality $\mathbb{E}(|W|) \leq \sqrt{\mathbb{E}(W^2)}$, where

$$\mathbb{E}(W^2) = \text{Var}(K_1) + \text{Var}(K_2) + [\mathbb{E}(K_1) - \mathbb{E}(K_2)]^2, \quad (\text{C8})$$

with $\mathbb{E}(K_i)$ given in Eq. (C5) and

$$\text{Var}(K_i) = p_i(1 - p_i)n_i^{\max}N_i^2. \quad (\text{C9})$$

Then, we can write the lower bound

$$\mathbb{E}(K) \geq \frac{\mathbb{E}(K_1) + \mathbb{E}(K_2) - \sqrt{\mathbb{E}(W^2)}}{2}. \quad (\text{C10})$$

APPENDIX D: QKD GRID ANALYSIS

Here, we provide an alternative analysis of the security-rate trade-off for a network that is not randomly generated. We consider an 8×8 square-lattice topology with diagonal diamond-cell connectivity for a total of 64 nodes and 210 links. This topology can be interpreted as a candidate backbone QKD network for a Manhattan-like metropolitan area. We assume that generic QKD protocols are implemented on the links, each characterized by a key rate R and an overall security parameter ε . The grid is constructed so that vertical and horizontal links have a lower rate, 1 Mbps, but higher security, at level 9, whereas diagonal links have a higher rate,

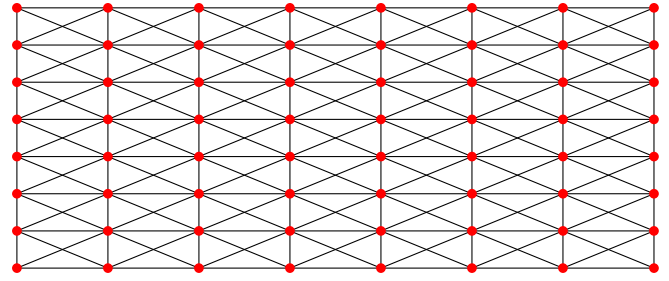


FIG. 14. Regular 8×8 diamond-cell grid topology with 64 nodes and 210 links. Horizontal and vertical links have a key rate of 1 Mbps with security level 9, while diagonal links have a key rate of 10 Mbps at security level 8.

10 Mbps, but lower security, at level 8. Although a realistic implementation would generally exhibit stochastic variability around these nominal values, using fixed values allows us to isolate the main behavior of a regular infrastructure with heterogeneous link figures of merit (see Fig. 14).

Our analysis of the “diamond-cell” network is based on a sample of 1000 pairs of end nodes, randomly selected to create a unicast connection. As we can see in Fig. 15, the various protocols analyzed in the main text feature the same behavior seen for the random network. In the present case, the security-optimized single-path $\mathcal{P}_{sp}^{\Sigma^*}$ selectively chooses the vertical and horizontal links with a lower rate (1 Mbps) and higher security level 9. Because of the composition of the epsilon security on the links of the network, the final ETE value would range between 9 and $\Sigma_{\min} = -\log_{10}(n_{\text{hops}} \times 10^{-9})$, where $n_{\text{hops}} = 14$ is the number of vertical/horizontal links connecting the top left node with the bottom right one. This leads to $\Sigma_{\min} \simeq 7.8$, which is compatible with the lower bound in the violin of $\mathcal{P}_{sp}^{\Sigma^*}$ in the bottom panel of Fig. 15.

When we use the rate-optimized single path $\mathcal{P}_{sp}^{R^*}$, we see that the rate is distributed between the two values of 1 and 10 Mbps. This protocol tends to choose the diagonal links in the construction of the path, but this is not always possible.

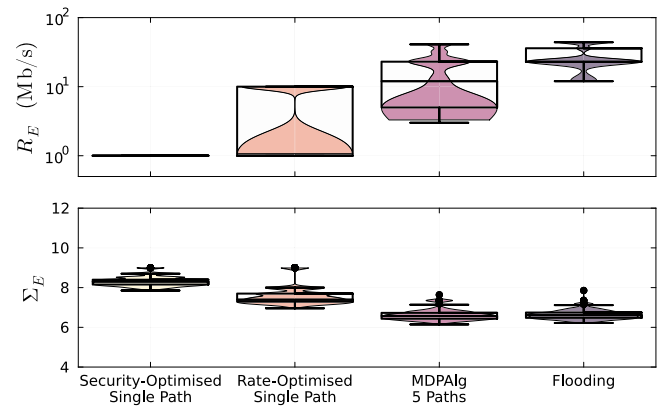


FIG. 15. Distributions of ETE rates and security levels, for different routing protocols, established via 1000 sampled ETE route computations on the network. We show security-optimized single-path $\mathcal{P}_{sp}^{\Sigma^*}$, rate-optimized single-path $\mathcal{P}_{sp}^{R^*}$, five-path edge-disjoint multipath $\mathcal{P}_{mdp}^{N_p=5}$, and flooding $\mathcal{P}_{fl}$.

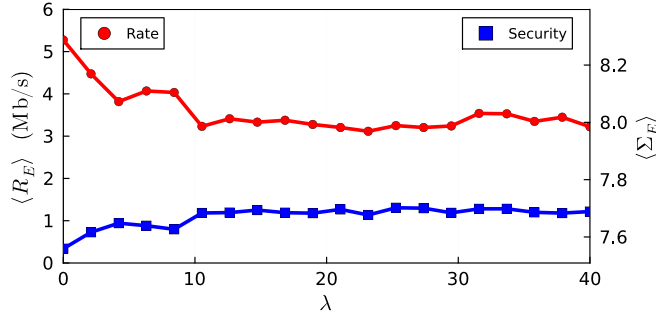


FIG. 16. Average rate and security level of the lambda protocol $\mathcal{P}_\lambda$, for different values of λ .

In fact, roughly half of the time, it needs to move along a horizontal/vertical link. To understand this behavior, consider that a generic node in the grid $\mathbf{u}$ is labeled by (i, j) , with $i, j = 1, \dots, 8$. Let us define the parity of the node as $p(\mathbf{u}) = i + j$ and introduce to subset of nodes:

$$\begin{aligned} P_{\text{odd}} &:= \{\mathbf{u} \in P : p(\mathbf{u}) \text{ is odd}\}, \\ P_{\text{even}} &:= \{\mathbf{u} \in P : p(\mathbf{u}) \text{ is even}\}. \end{aligned} \quad (\text{D1})$$

Then, it is clear that a path can only connect two nodes in the same parity subset. This explains why there is an ensemble of rates in the violin for $\mathcal{P}_{\text{sp}}^{R^*}$ in the upper panel of Fig. 15. Then, because diagonal links are used, security decreases, as shown in the bottom panel.

Finally, the behavior of the multipath protocols, $\mathcal{P}_{\text{mdp}}^{N_p=5}$ and $\mathcal{P}_{\text{h}}$, is as expected. The connectivity of the grid enables these protocols to effectively choose multiple paths between any pair of end nodes, therefore boosting their ETE key rate while decreasing the corresponding ETE security level (well below 8).

We complete the analysis in Figs. 16 and 17. In Fig. 16, we analyze the security-rate trade-off associated with the lambda protocol $\mathcal{P}_\lambda$. As expected, at low values of λ , it behaves like a rate-optimized protocol, while it tends to a security-optimized protocol for increasing λ . The plot exhibits a granular behav-

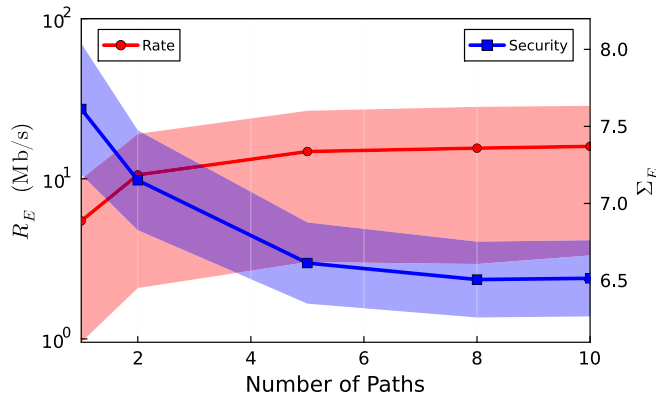


FIG. 17. Rate-security trade-off relative to multipath routing, considering the performance/security of the N_p -path edge-disjoint routing strategy $\mathcal{P}_{\text{mdp}}^{N_p}$, showing the security degradation/rate amplification associated with increasing number of paths (here we show both average and standard deviation).

ior, which can be attributed to finite-size geometric effects of the grid. In conclusion, in Fig. 17, we study the trade-off for the edge-disjoint routing protocol $\mathcal{P}_{\text{mdp}}^{N_p}$ as a function of the number of paths N_p , confirming the expected behavior.

APPENDIX E: IDEAL LATTICES

All of our lattice networks are generated within rectangular regions of side lengths L_x, L_y km. This gives rise to some characteristic properties of the ideal lattices, which are then perturbed in our more general model.

1. Grid lattice

For the grid lattice, we consider a square area with side length $L_x = L_y = L$ km. Given an ideal grid lattice of nodal density ρ nodes/km², there will be on average $N = \lceil \rho L^2 \rceil$ nodes. This gives rise to an $n \times n$ grid of $N = n^2$ nodes, where $n(\rho) := \lfloor \sqrt{\lceil \rho L^2 \rceil} \rfloor$. Consequently, this grid lattice gives rise to a single degenerate characteristic length that can be found by dividing the side length of the network region L by the number of links that will traverse it at density ρ . That is,

$$a_{\text{grid}}(\rho) = \frac{L}{n(\rho) - 1}. \quad (\text{E1})$$

Connections in both the x and y directions will be of this length, and will thus exhibit characteristic single-link rates/security values when paired with a QKD protocol.

2. Triangular lattice

The triangular lattice is based on alternating rows of nodes that are shifted in the x axis to create triangular cells of connections. It is easier to refer to horizontal *rows* (number of horizontal lattice grid lines) and node *columns* (number of nodes on each horizontal lattice grid line) in this lattice, such that an $n_{\text{row}} \times n_{\text{col}}$ triangular lattice contains $N = n_{\text{col}} n_{\text{row}}$ nodes.

Due to the diagonal connections in the triangular lattice, we will not necessarily be considering a square area, but a rectangular area of x length $L_x = L$ km and y length $L_y = (\sqrt{3}/2)L$ km. Given a nodal density ρ nodes/km², there will be on average $N = \frac{\sqrt{3}}{2} \rho L^2$ nodes. Asserting that

$$n_{\text{col}} = n_{\text{row}} = n(\rho), \quad (\text{E2})$$

then the separation between connected nodes along the horizontal rows will be

$$a_{\text{tri}}(\rho) = \frac{L}{n(\rho) - 1}. \quad (\text{E3})$$

By asserting a vertical spacing between the rows of $(\sqrt{3}/2)a_{\text{tri}}(\rho)$, it follows that the diagonal connections will also be of length $a_{\text{tri}}(\rho)$, i.e.,

$$a_{\text{tri}}^{\text{diag}}(\rho) = \sqrt{\left(\frac{a_{\text{tri}}(\rho)}{2}\right)^2 + \left(\frac{\sqrt{3}a_{\text{tri}}(\rho)}{2}\right)^2} = a_{\text{tri}}(\rho). \quad (\text{E4})$$

3. Honeycomb lattice

For the honeycomb lattice, we will use an identical construction to the triangular lattice; however, we will alter the

vertical spacing between rows in order to achieve a uniform equidistant link length similar to the other lattice types. Again, it is easier to refer to the honeycomb lattice in terms of row and column grid lines; however, we define these differently in the honeycomb case. We define a row and column as blocks of n_{row} and n_{col} nodes as shown in Fig. 18, such that an $n_{\text{row}} \times n_{\text{col}}$ lattice contains $N = \frac{n_{\text{col}} n_{\text{row}}}{2}$ nodes. We consider a rectangular area of x length $L_x = L$ km and y length $L_y = \frac{\sqrt{3}}{2} L$ km. For a density ρ nodes/km², we will consider $\frac{\sqrt{3}}{2} \rho L^2$ nodes. Once more, we will assert that $n_{\text{col}} = n_{\text{row}} = n$.

We can construct the lattice by thinking of an iterative block-row pattern. This can be seen in Fig. 18, where we stitch row blocks on top of inverted row blocks via vertical connections of characteristic length. In this model, the characteristic length will be

$$a_{\text{honey}} = \frac{2\sqrt{3}L}{3[n(\rho) - 1]}. \quad (\text{E5})$$

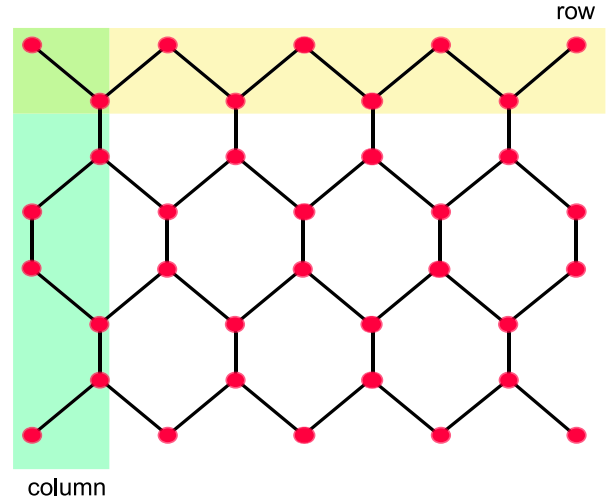


FIG. 18. Rows and columns in the honeycomb lattice.

- [1] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, *et al.*, Advances in quantum cryptography, *Adv. Opt. Photonics* **12**, 1012 (2020).
- [2] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, Quantum cryptography, *Rev. Mod. Phys.* **74**, 145 (2002).
- [3] C. H. Bennett and G. Brassard, Quantum cryptography: Public key distribution and coin tossing, in *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing* (IEEE, Bangalore, India, 1984), pp. 175–179.
- [4] A. K. Ekert, Quantum cryptography based on Bell's theorem, *Phys. Rev. Lett.* **67**, 661 (1991).
- [5] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi, Fundamental limits of repeaterless quantum communications, *Nat. Commun.* **8**, 15043 (2017).
- [6] S. Pirandola, Limits and security of free-space quantum communications, *Phys. Rev. Res.* **3**, 013279 (2021).
- [7] S. Pirandola, Satellite quantum communications: Fundamental bounds and practical security, *Phys. Rev. Res.* **3**, 023130 (2021).
- [8] M. Lucamarini, Z. L. Yuan, J. F. Dynes, and A. J. Shields, Overcoming the rate-distance limit of quantum key distribution without quantum repeaters, *Nature (London)* **557**, 400 (2018).
- [9] Z.-W. Yu, X.-L. Hu, C. Jiang, H. Xu, and X.-B. Wang, Sending-or-not-sending twin-field quantum key distribution in practice, *Sci. Rep.* **9**, 3080 (2019).
- [10] C. Jiang, Z.-W. Yu, X.-L. Hu, and X.-B. Wang, Unconditional security of sending or not sending twin-field quantum key distribution with finite pulses, *Phys. Rev. Appl.* **12**, 024061 (2019).
- [11] H. Xu, Z.-W. Yu, C. Jiang, X.-L. Hu, and X.-B. Wang, Improved results for sending-or-not-sending twin-field quantum key distribution: Breaking the absolute limit of repeaterless key rate, *Phys. Rev. A* **101**, 042330 (2020).
- [12] C. Cui, Z.-Q. Yin, R. Wang, W. Chen, S. Wang, G.-C. Guo, and Z.-F. Han, Twin-field quantum key distribution without phase postselection, *Phys. Rev. Appl.* **11**, 034053 (2019).
- [13] S. Pirandola, End-to-end capacities of a quantum communication network, *Commun. Phys.* **2**, 51 (2019).
- [14] O. Simeone, *Classical and Quantum Information Theory: Uncertainty, Information, and Correlation* (Cambridge University Press, Cambridge, 2026).
- [15] The block rate R_{block} represents the conditional rate of the protocol, i.e., the rate achievable when a block is successfully corrected (which happens with probability p_{EC}). Then, we can define the average key rate $p_{\text{EC}} R_{\text{block}}$, which represents the unconditional rate of the protocol, i.e., the average rate that is achievable considering that some of the blocks are successful and others are not. This also corresponds to the average rate of a session R_{ses} , where an average of $p_{\text{EC}} n_{\text{bks}}$ blocks are successfully corrected. Note that we are implicitly assuming a stationary QKD condition, i.e., stable channel and setup parameters. Under this condition, we assume that PE is successful (no abort due to denial-of-service attacks) and that the value of p_{EC} remains constant during the session.
- [16] V. Burenkov, B. Qi, B. Fortescue, and H.-K. Lo, Security of high speed quantum key distribution with finite detector dead time, *Quantum Inf. Comput.* **14**, 217 (2014).
- [17] As we will see in Appendixes A and B, the PE error would appear as $p_{\text{EC}} \epsilon_{\text{PE}}$ because PE is assumed to be done before EC. However, it is preferable to use the upper bound $p_{\text{EC}} \epsilon_{\text{PE}} \leq \epsilon_{\text{PE}}$ in the formulas, since the epsilon errors are usually set as input target parameters, while the exact value of p_{EC} is typically known only after the actual implementation of EC.
- [18] Note that the expression for the session security parameter should be understood as the standard composable-security upper bound $\epsilon_{\text{ses}} \leq n_{\text{bks}} \epsilon_{\text{block}}$. This additive form follows from the union bound and is tight to first order when the total failure probability is small. For typical values of ϵ_{block} between 10^{-15} and 10^{-5} , and n_{bks} on the order of a few hundred, one has $n_{\text{bks}} \epsilon_{\text{block}} \ll 1$, so the additive union-bound estimate is well justified.
- [19] F. Grosshans and P. Grangier, Continuous variable quantum cryptography using coherent states, *Phys. Rev. Lett.* **88**, 057902 (2002).

- [20] C. Weedbrook, A. M. Lance, W. P. Bowen, T. Symul, T. C. Ralph, and P. K. Lam, Quantum cryptography without switching, *Phys. Rev. Lett.* **93**, 170504 (2004).
- [21] Given some time interval over which we can assume stability, the effective number of session points $n_{\text{bks}}N$ depends on the repetition rate. This means that the stability is relative to the system's clock.
- [22] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. (Pearson, Upper Saddle River, NJ, 2010).
- [23] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks* (Princeton University Press, Princeton, NJ, 1962).
- [24] L. R. Ford, Jr. and D. R. Fulkerson, Maximal flow through a network, *Can. J. Math.* **8**, 399 (1956).
- [25] One way to mitigate this trade-off and meet a target value for the ETE epsilon security is to decrease the link epsilons, but this in turn means to decrease the link rates. At the end of the day, the best strategy may just consist in reducing the number of parallel paths.
- [26] The additive multipath rate considered here should be interpreted as an idealized upper-level model. In practice, classical-control overheads may lead to mildly subadditive aggregation. However, because these effects are typically separated from the physical process of quantum key generation, which remains the dominant bottleneck, they are neglected here without affecting the security-rate trade-off.
- [27] In the presence of horizontal depth, we can potentially meet a target value of the ETE epsilon security by reducing the link values. This is, of course, achieved at the cost of lowering the link rate and therefore the ETE rate.
- [28] E. W. Dijkstra, A note on two problems in connexion with graphs, *Numer. Math.* **1**, 269 (1959).
- [29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. (MIT Press, Cambridge, MA, 2009).
- [30] D. Lopez-Pajares, E. Rojas, J. A. Carral, I. Martinez-Yelmo, and J. Alvarez-Horcajo, The disjoint multipath challenge: Multiple disjoint paths guaranteeing scalability, *IEEE Access* **9**, 74422 (2021).
- [31] C. Harney and S. Pirandola, Practical routing and criticality in large-scale quantum communication networks, *Phys. Rev. Res.* **7**, 043168 (2025).
- [32] C. Harney and S. Pirandola, Analytical methods for high-rate global quantum networks, *PRX Quantum* **3**, 010349 (2022).
- [33] P. Erdős and A. Rényi, On the evolution of random graphs, *Publ. Math. Inst. Hungarian Acad. Sci.* **5**, 17 (1960).
- [34] M. E. J. Newman, *Networks: An Introduction* (Oxford University Press, Oxford, 2010).
- [35] C. Portmann, Key recycling in authentication, *IEEE Trans. Inf. Theory* **60**, 4383 (2014).
- [36] M. N. Wegman and L. Carter, New hash functions and their use in authentication and set equality, *J. Comput. Syst. Sci.* **22**, 265 (1981).
- [37] info@nodeq.io.
- [38] D. Rusca, A. Boaron, F. Grünenfelder, A. Martin, and H. Zbinden, Finite-key analysis for the 1-decoy state QKD protocol, *Appl. Phys. Lett.* **112**, 171104 (2018).
- [39] J. Wiesemann, J. Krause, D. Tupkary, N. Lütkenhaus, D. Rusca, and N. Walenta, A consolidated and accessible security proof for finite-size decoy-state quantum key distribution, *Quantum* **10**, 2037 (2026).
- [40] V. Scarani and R. Renner, Security bounds for quantum cryptography with finite resources, in *Proceedings of the Theory of Quantum Computation*, Lecture Notes in Computer Science (Springer, Berlin, 2008), Vol. 5106, pp. 83–95.
- [41] V. Scarani and R. Renner, Quantum cryptography with finite resources: Unconditional security bound for discrete-variable protocols with one-way post-processing, *Phys. Rev. Lett.* **100**, 200501 (2008).
- [42] M. Tomamichel, A framework for non-asymptotic quantum information theory, Ph.D. thesis, ETH Zurich, Zurich, Switzerland, 2012.
- [43] R. Renner, Security of quantum key distribution, Ph.D. thesis, ETH Zurich, Zurich, Switzerland, 2005.
- [44] Here, we set $\tilde{\rho} = \tilde{\rho}_{\text{XE}}$ where we exclude Eve's classical register C from $\mathbf{E} = \mathbf{E}'\mathbf{C}$. The contribution of system C has been considered in the leak_{EC} term.
- [45] D. Gottesman and H.-K. Lo, Proof of security of quantum key distribution with two-way classical communications, *IEEE Trans. Inf. Theory* **49**, 457 (2003).
- [46] B. Kraus, N. Gisin, and R. Renner, Lower and upper bounds on the secret-key rate for quantum key distribution protocols using one-way classical communication, *Phys. Rev. Lett.* **95**, 080501 (2005).
- [47] R. Renner, N. Gisin, and B. Kraus, Information-theoretic security proof for quantum-key-distribution protocols, *Phys. Rev. A* **72**, 012332 (2005).
- [48] C. C. W. Lim, M. Curty, N. Walenta, F. Xu, and H. Zbinden, Concise security bounds for practical decoy-state quantum key distribution, *Phys. Rev. A* **89**, 022307 (2014).
- [49] W. Hoeffding, Probability inequalities for sums of bounded random variables, *J. Am. Stat. Assoc.* **58**, 13 (1963).
- [50] H. L. Yin, M. G. Zhou, J. Gu, Y. M. Xie, Y. S. Lu, and Z. B. Chen, Tight security bounds for decoy-state quantum key distribution, *Sci. Rep.* **10**, 14312 (2020).
- [51] R. J. Serfling, Probability inequalities for the sum in sampling without replacement, *Ann. Stat.* **2**, 39 (1974).
- [52] C.-H. F. Fung, X. Ma, and H. F. Chau, Practical issues in quantum-key-distribution postprocessing, *Phys. Rev. A* **81**, 012318 (2010).
- [53] S. Pirandola and P. Papanastasiou, Improved composable key rates for CV-QKD, *Phys. Rev. Res.* **6**, 023321 (2024).
- [54] Note that there is no reduction of the block size N induced by sifting or channel loss as in DV-QKD.
- [55] A. G. Mountogiannakis, P. Papanastasiou, B. Braverman, and S. Pirandola, Composably secure data processing for Gaussian-modulated continuous-variable quantum key distribution, *Phys. Rev. Res.* **4**, 013099 (2022).
- [56] A. G. Mountogiannakis, P. Papanastasiou, and S. Pirandola, Data postprocessing for the one-way heterodyne protocol under composable finite-size security, *Phys. Rev. A* **106**, 042606 (2022).
- [57] Note that this random choice must be done for each block in case of a session, so each block is compressed using a potentially different function.