Experimental anonymous conference key agreement using linear cluster states

Lukas Rückle,[1,2] Jakob Budde,[1,2] Jarn de Jong,[3] Frederik Hahn,[3,4] Anna Pappa,[3,5] and Stefanie Barz[1,2,*]

[1]*Institute for Functional Matter and Quantum Technologies, Universität Stuttgart, 70569 Stuttgart, Germany*
[2]*Center for Integrated Quantum Science and Technology (IQST), Universität Stuttgart, 70569 Stuttgart, Germany*
[3]*Electrical Engineering and Computer Science Department, Technische Universität Berlin, 10587 Berlin, Germany*
[4]*Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany*
[5]*Fraunhofer-Institute for Open Communication Systems FOKUS, 10589 Berlin, Germany*

(Received 4 August 2022; revised 21 April 2023; accepted 1 August 2023; published 28 September 2023)

Multipartite entanglement enables secure and anonymous key exchange between multiple parties in a network. Greenberger-Horne-Zeilinger states have been introduced as resource states for anonymous key exchange protocols, in which an anonymous subset of parties within a larger network establishes a secret key. However, the use of other types of multipartite entanglement for such protocols remains relatively unexplored. Here, we demonstrate that linear cluster states can serve as a versatile and potentially scalable resource in such applications. We implemented an anonymous key exchange protocol with four photons in a linear cluster state and established a shared key between three parties in our network. We show how to optimize the protocol parameters to account for noise and to maximize the finite key rate under realistic conditions. As cluster states have been established as a flexible resource in quantum computation, we expect that our demonstration provides a first step towards their hybrid use for networked computing and communication.

DOI: 10.1103/PhysRevResearch.5.033222

I. INTRODUCTION

Quantum communication has been expanded from the initially proposed bipartite key exchange [1,2] to networked settings [3–6]. One particularly interesting application of quantum networks is conference key agreement [7,8]. In such protocols, multipartite entangled states are used to realize a key exchange in a quantum network. It has been shown that such a networked key exchange is possible by sharing Greenberger-Horne-Zeilinger (GHZ) states in a network [9]. Their quantum correlations can be harnessed for establishing a joint key and for performing verification. In the latter step, an eavesdropper or any other deviation in the protocol can be detected, similar to the bipartite case, making the protocol *secret* [7]. Networked key exchange has been shown to be more efficient than several bipartite links [10,11].

Furthermore, the use of multipartite entanglement made it possible to efficiently realize another security feature beyond secrecy: *anonymity*. It hides the information who a party is communicating with. This is an important feature, for example, in situations where the message is more or less clear once the recipient is known: an employee communicating with a head hunter of a competing company, a whistle blower contacting investigative journalists, or a person consulting a medical specialist. By exploiting the particular properties of GHZ states, multiple parties can communicate in a quantum network with their identities protected [11,12]. In other words, a key is exchanged between a subset of parties of a network while it remains hidden which parties belong to this subset. Such anonymous quantum conference key agreement also allows verification and thus the detection of deviating parties or eavesdroppers. There are various implementations of conference key agreement and its anonymous equivalent in quantum networks [10,13,14].

So far, many protocols for conference key agreement build strongly on the particular correlations of GHZ states [11–14]. This invites the question whether multipartite entangled states other than GHZ might be suitable as a resource for such protocols. Of particular interest with regard to scalability are physical quantum networks that, due to their topology or physical hardware, favor building up links in the form of linear cluster states.

Here, we study the use of linear cluster states as a resource for anonymous conference key agreement between a subset of parties in a larger network. We generate four-photon cluster states and demonstrate that they provide a basis for key exchange between three parties of our network by implementing a recently introduced protocol [15]. Specifically, we exchange a key with a length of 40 kbit and demonstrate the encryption, sharing, and decryption of a picture over the network. We evaluate the success rates of the protocol for different network configurations and examine the influence of experimental imperfections, in particular, how the parameters of the protocol can be adapted to certain noise values.

While we are focusing on a key exchange in a network of four parties, the protocol can be scaled to anonymous three-partite communication in a larger network. As such, our work establishes the potential of cluster states beyond applications in quantum computing [16].

*Corresponding author: barz@fmq.uni-stuttgart.de

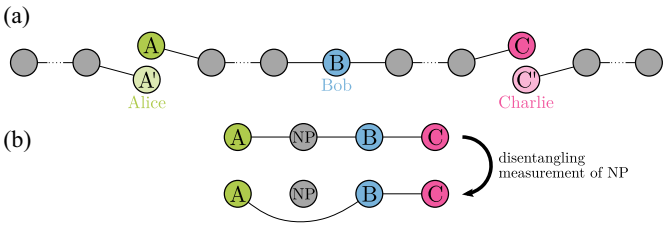

FIG. 1. Sharing of the linear cluster state in the network. (a) The general protocol uses three linear cluster states, where Alice and Charlie have the first and last qubit, respectively, of the central linear cluster state, which is used for key generation. The use of three cluster states ensures the anonymity of all parties (see Ref. [15] for details). (b) In this paper, we generate a four-qubit linear cluster state acting as the central cluster state. From this, we extract a state that is locally equivalent to a three-qubit linear cluster state. The exact state held by ABC depends on the disentangling measurement as well as on the outcome of that measurement.

II. PROTOCOL

We start by introducing the main steps of the protocol [15]. The first step is the creation of the resource state, a linear cluster state, followed by its distribution to all parties in the network. Such a linear cluster state can, in general, be created by each party holding a qubit in the state $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, where $|0\rangle$ and $|1\rangle$ are the computational basis states. The qubits are then entangled by applying *CPhase* gates between pairs of neighboring qubits [17]. Here, the action of the CPhase gate is $\mathrm{CPhase}|ij\rangle = (-1)^{ij}|ij\rangle$ [18]. In photonic settings, like the one studied here, one often starts with two-qubit entangled states that are then fused to a larger cluster state [19]. This cluster state is then shared in the network such that each party receives one qubit.

The general protocol for a network with n parties has been introduced in Ref. [15] and is summarized in Appendix A. In this paper, we focus on a network with four parties and aim at exchanging a key between three of the parties: Alice, Bob, and Charlie (ABC). The protocol now works as follows: We generate a four-qubit linear cluster state $|\mathrm{LC}_4\rangle$ that is shared within the network. The state is defined as described above and reads

$$|\mathrm{LC}_4\rangle = \tfrac{1}{2}(|{+}0{+}\rangle + |{+}1{-}\rangle + |{-}1{+}\rangle - |{-}1{-}\rangle), \quad (1)$$

where $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$. The party not participating in the key exchange (NP) then performs a measurement in the Pauli basis σ_X or σ_Y. This measurement effectively removes their qubit from the four-qubit cluster state and leaves a three-qubit cluster state with ABC (see Fig. 1).

ABC measure their qubits to generate a key, exploiting the correlations of the three-qubit linear cluster state which is locally equivalent to a three-qubit GHZ state. This measurement type is called *key generation*. As the exact three-qubit state between ABC depends on the outcome of the disentangling measurement of the NP party, ABC perform bitflips conditioned on that measurement outcome (see Appendix C). In addition, ABC perform *verification* measurements by measuring a stabilizer element of the linear cluster state. This is a Pauli operator to which the state is an eigenstate with eigenvalue $+1$. Thus, the correlations between the measurement outcomes of ABC are known. If ABC receive measurement outcomes that are correlated in an unexpected way, this reveals possible eavesdropping attacks and parties in the network that deviate from the protocol. Before they perform a measurement, ABC coordinate if it is a key generation or a verification measurement and encrypt this communication with a preshared key. The measurement bases for both types of measurement are given in Appendix C.

The protocol not only enables the creation of a secret key between ABC, but also guarantees their *anonymity*. This means that only within the group of ABC, the network positions of each other are known, but are kept secret from all NP parties [15]. The protocol maintains the anonymity of ABC during all its steps, as long as there is no collective attack. This entails that the NP parties are not able to conclude anything about the positions of ABC from their measurement instruction, measurement outcome, or any public communication they follow. For detailed explanations and proofs, we refer to Appendix A and Ref. [15].

We perform verification measurements in a percentage p of the L total rounds, where one round is a four-fold event. For both types of settings, we define $Q_{\mathrm{keygen}} = n_{\mathrm{keygen,incorr}}/n_{\mathrm{keygen,tot}}$ and $Q_{\mathrm{verif}} = n_{\mathrm{verif,incorr}}/n_{\mathrm{verif,tot}}$ as the ratio of incorrect rounds n_{incorr} to total rounds n_{tot} for the respective round types. The success rates of the key generation and the verification rounds are then given by $1 - Q_{\mathrm{keygen}}$ and $1 - Q_{\mathrm{verif}}$, respectively. The parameter Q_{keygen} is an upper bound for the pairwise bit error rates $Q^{A,B}_{\mathrm{keygen}}$, between Alice and Bob, and $Q^{A,C}_{\mathrm{keygen}}$, between Alice and Charlie. Q_{verif} allows us to infer the maximal knowledge that a potential eavesdropper could gain about the key. In the postprocessing steps, error correction and privacy amplification can be applied to the raw key to receive a correct and secret key. The number of key bits needed for those postprocessing routines depends on the maximum of the pairwise bit error rates $Q^{A,B}_{\mathrm{keygen}}$ and $Q^{A,C}_{\mathrm{keygen}}$ for error correction and on Q_{verif} for privacy amplification as well as on the desired level of security (see Sec. 3.2 of Ref. [15]).

III. EXPERIMENT AND RESULTS

In our implementation, we generate two pairs of entangled photons and fuse them to a four-photon linear cluster state by applying a photonic CPhase gate on one photon from each pair [20]. We use polarization encoding $|0/1\rangle = |H/V\rangle$. The two entangled photon pairs are created by spontaneous parametric down conversion (SPDC) in barium borate (BBO) crystals. A scheme of the protocol implementation is shown and described in Fig. 2. The experimental setup is shown in Fig. 6.

The generated state is characterized using quantum state tomography and a maximum likelihood estimation [21]. The fidelity at a pump power of 400 mW is estimated to $79.9 \pm 0.8\%$. The main source of noise is higher-order emission of the SPDC sources, which accounts for 4% of all events. In addition, partial distinguishability as well as spectral mixedness affect the two photon interference at the polarization-dependent beam splitter (PDBS) in the implementation of the CPhase gate.

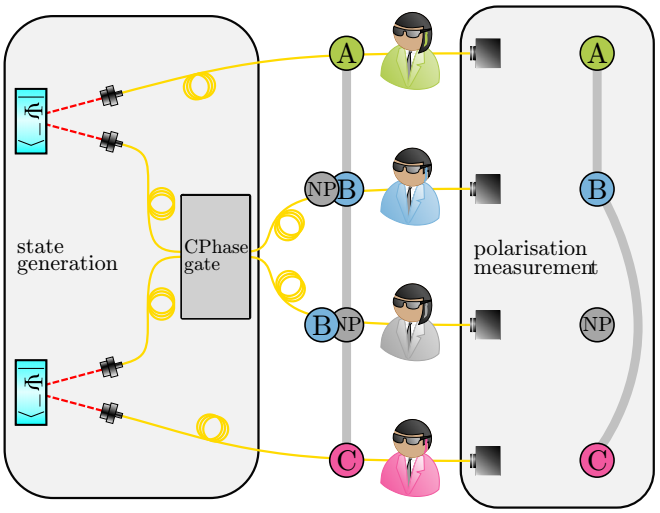

FIG. 2. Schematic view of the implementation of the protocol. We generate two pairs of entangled photons in the state $|\Psi^-\rangle = (|H\rangle|V\rangle - |V\rangle|H\rangle)/\sqrt{2}$. Applying a CPhase gate to one photon from each pair generates a four-photon cluster state upon postselection. The four qubits are allocated to the different parties Alice, Bob, and Charlie, (ABC) and a not participating party (NP). Polarization measurements of the qubits can be realized in any basis and thus allow disentangling, key generation, and verification operations. Note that we create a state that is locally equivalent to the state given in Eq. (1) and account for the local unitaries by adapted measurement settings; see Appendix C for details.

In our implementation, Alice and Charlie hold the first and last qubit of the state, respectively, meaning that either the party holding the second or third qubit is NP. In total, there exist four experimental configurations, as either measurement in the Pauli basis σ_X or σ_Y can be used to disentangle the parties that do not participate while preserving entanglement between ABC. We label those configurations X_2, Y_2, X_3, Y_3, where the letter indicates the type of disentangling measurement and the number the party removing themselves from the network. The choice of the configuration determines the measurement settings of each party for the key generation and verification rounds. We measure all eight measurement settings—one key generation setting and one verification setting for each of the four configurations—and determine the success rate for each setting (see Fig. 3).

For the implementation of the protocol in a realistic setting, we choose randomly whether key generation or verification is performed using a biased random number generator. A single fourfold event is considered a round. For each setting, we integrate over a time of 60 s, which we call a run containing multiple rounds. At the start of each run, the biased random number generator indicates if the next run is a key generation run or a verification run.

We set $p = 10\%$; we exchange 41 033 bits for key generation and measure 3794 rounds for verification. From the measurements, we obtain a success probability of $(87.76 \pm 0.54)\%$ for the key generation rounds and $(87.01 \pm 0.55)\%$ for the verification rounds. Note that because the random number generator is called only a finite number of times, the ratio of verification rounds from total rounds slightly differs from the value of p.

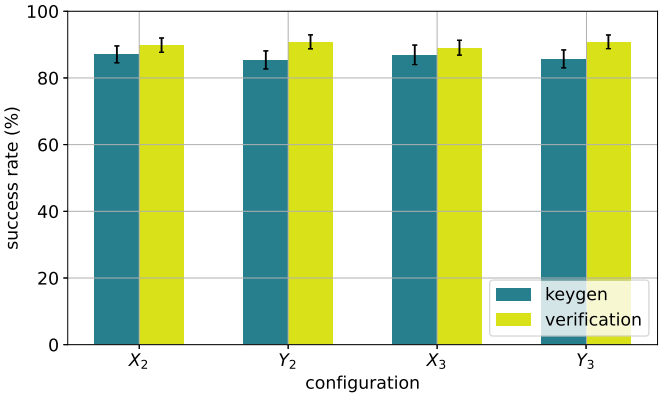

FIG. 3. Success rate of key generation and verification rounds for the four different configurations.

To illustrate the protocol and give a visual idea for the error rate, we use the key to encrypt a binary image by performing an XOR operation for every image pixel with a bit from Alice's key (see Fig. 4). The XOR operation is defined for binary numbers x, y as the sum modulo 2: $\mathrm{XOR}(x, y) = x \oplus y$. Bob and Charlie can decrypt the image using their keys: If the bits of the encryption and decryption key are the same, the original image pixel is retrieved. However, if due to errors, bits of the encryption and decryption key are different, this will result in an incorrectly communicated image pixel. In our implementation, this noise arises from imperfections in the state preparation and transmission. As a result, classical postprocessing is necessary because, with an erroneous key, the sent message will also contain errors.

Error correction

We use low-density parity check codes (LDPCs) to perform *error correction* (see Appendix D). Alice computes parity bits called *error syndrome* from her raw key and sends them to Bob and Charlie, who then correct their key. The ratio r between the number of raw key bits and the sum of raw key and parity bits is called the *code rate* and its chosen value is dependent on the error rate. If r is too high, meaning if not enough parity bits are used, not all errors can be corrected. For different values of r and depending on $Q^{A,B}_{\mathrm{keygen}}$ and $Q^{A,C}_{\mathrm{keygen}}$, respectively, the error can be corrected partially or completely (see Table I). For the keys in this paper, all errors could be corrected using a code rate of $r = 0.5$. A detailed explanation

TABLE I. Ratio of bits different from Alice's key in the keys of Bob and Charlie for the raw key and error corrected keys with different code rates r.

	Bob (%)	Charlie (%)
Raw key	10.37	9.67
$r = 2:3$	10.22	8.88
$r = 3:5$	6.82	0
$r = 1:2$	0	0

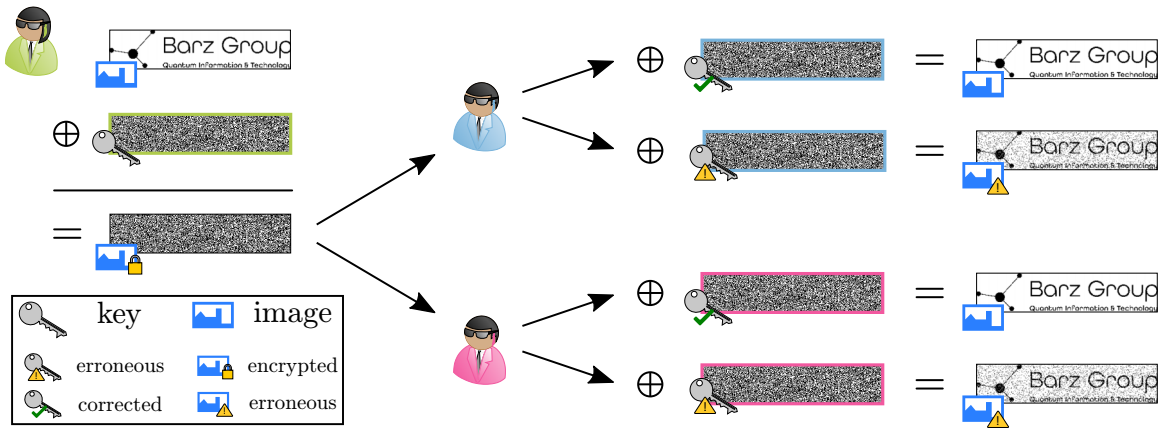

FIG. 4. Encryption and decryption with the generated key. Left: Alice encrypts a binary image using her key and sends the encrypted image to Bob and Charlie. In this exemplary case, an XOR operation is used for encryption. Right: Bob and Charlie use their keys to decrypt the sent encrypted image. In case of using a not corrected key, the obtained image contains errors. Note that this figure at the same time illustrates the necessity of privacy amplification, as it must not be possible for any adversary to guess parts of the transmitted message when having a partially correlated key.

and the chosen parameters of the error correction procedure can be found in Appendix D.

IV. ANALYSIS OF NOISE-ADAPTED PARAMETER CHOICE

In this section, we analyze how the choice of the protocol parameters affects the performance of the protocol. As the optimal postprocessing methods may depend on the particular noise level and raw key length, our analysis is independent of specific postprocessing methods. For this analysis, we perform another run of the protocol in the configuration X_2 and set $p = 2\%$. We retrieved 11 108 rounds in total, including 10 814 key generation rounds and 294 verification rounds. From the measurement outcomes, we estimate the values $Q_{\text{verif}} = (11.2 \pm 1.8)\%$ and $\max(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}}) = (9.59 \pm 0.28)\%$.

In a realistic setting, errors in the key could be introduced by an eavesdropper trying to gain knowledge about the key, thus compromising the security of the protocol. Therefore, privacy amplification is needed in addition to error correction. For both error correction and privacy amplification, a fraction of the exchanged key has to be used. Note that different to other conference key agreement schemes, for maintaining the anonymity of ABC, the error syndrome is encrypted. However, this still leads to the same amount of key used for the postprocessing in total, since less key is needed for privacy amplification compared to other schemes. The fraction of the key needed for those postprocessing steps should be small compared to the key length to obtain a positive key rate. It depends on the parameters Q_{verif} and $\max(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}})$ as they indicate the level of information leakage and errors in the key, respectively. If Q_{verif} and $\max(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}})$ are too high, no secure and correct key can be achieved using the postprocessing steps.

The *asymptotic key rate* (AKR) is an upper bound to the maximal achievable fraction of the raw key which can be used as a correct and secure key. For the protocol used here, it is given by

$$\text{AKR} = 1 - h(Q_{\text{verif}}) - h\big(\max\big(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}}\big)\big), \tag{2}$$

where h is the binary entropy:

$$h(x) := -x\log_2(x) - (1-x)\log_2(1-x). \tag{3}$$

The binary entropy takes values between 0 and 1, therefore the AKR has a value between -1 and 1. A negative value indicates that Q_{verif} and $\max(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}})$ are too large and thus the postprocessing steps cannot be carried out. The values measured in our experiment correspond to $h(Q_{\text{verif}}) = 0.507 \pm 0.055$ and $h(\max(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}})) = 0.456 \pm 0.001$, thus leading to a positive value of $\text{AKR} = 0.0375 \pm 0.0557$.

For finite keys of length L, the communication of the verification rounds and statistical uncertainties in the estimation of the errors lead to a smaller ratio of secure and correct key to raw key. This ratio is called the *finite key rate* (FKR). The AKR is the limit of the FKR for $L \to \infty$. The fact that the estimated parameters Q_{verif} and $\max(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}})$ are subject to statistical fluctuations leads to the possibility that even after postprocessing the key is not secret or incorrect. The tolerated level of the probability for a not secret or for an incorrect key is specified by the parameters ε_{S} and ε_{C}, respectively. Here, ε_{S} is the *secrecy* parameter which implies that the generated key is ε_{S} close to uniformly random for any adversary. ε_{C} is the *correctness* parameter that implies that the probability that the keys of the participants are different is smaller than ε_{C}. Together, these are referred to as the *security* parameters and are chosen close to zero. The postprocessing steps are adjusted to these parameters, meaning that the postprocessing is carried out like in a scenario where a higher ratio of the key is leaked and more errors occurred than the estimated values of Q_{verif} and $\max(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}})$ suggest. Hence, in addition to Q_{verif} and $\max(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}})$, the FKR depends on the parameters ε_{S}, ε_{C} as well as on L, p, and a free parameter ε. The exact formula is given in Appendix E. If all other parameters are given, one can estimate by numerical means

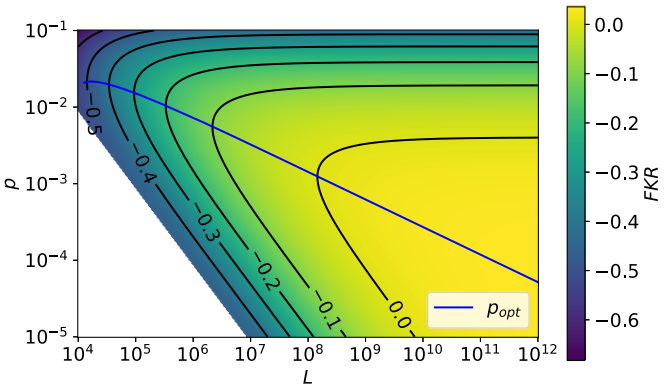

FIG. 5. Dependence of finite key rate FKR on the raw key length L and p. For the security parameters, the values $\varepsilon = 10^{-5}$, $\varepsilon_C = 10^{-5}$, and $\varepsilon_S = 2 \times 10^{-5}$ are chosen [see Appendix E, Eq. (E1)]. The minimal L at which one can get a positive FKR with the parameters estimated in this setup is given by $L = 1.46 \times 10^8$ when choosing a $p = 0.12\,\%$. The blue line indicates p_{opt}, which is the optimal choice of p for each L, meaning the value of p at which the FKR becomes maximal for each fixed L. In the limit of $L \to \infty$, p_{opt} goes to zero. Therefore, the AKR does not depend on p.

the optimal choice p_{opt} of the parameter p, which maximizes the FKR. In Fig. 5, the FKR is shown for the values of Q_{verif} and max $(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}})$ that we measured in our setup for different values of p and L, together with p_{opt} for each L.

V. CONCLUSION AND OUTLOOK

In this paper, we demonstrated anonymous quantum conference key agreement using a linear cluster state. We showed the encryption of a picture and analyzed the security parameters in the experimental setup and from this the finite and (positive) AKR. The change of the network configuration—meaning who belongs to the communicating subgroup and who does not—only requires the change of the measurement settings, which makes the introduced protocol a feasible and flexible technique of networked communication. This paper widens the range of possible resource states used for such networked communication tasks. As cluster states are an important resource in quantum computation, this could open an avenue to the hybrid use of these resource states for networked computing and communication.

The development of novel schemes for state generation and state transmission have the potential to further improve the protocol performance and enable its application for real world communication tasks. One possibility would be to use quantum repeaters to achieve a subexponential rate-distance scaling. However, quantum repeaters may introduce errors on the quantum state itself and, thus, examining the trade-off between using repeaters and the fidelity of the shared multipartite entangled state will be an important research topic for the future. Alternatively, there also exist schemes of photon-loss-tolerant encoding which could be used to overcome the distance limitation introduced by loss [22]. In addition to loss, one challenge when scaling the system up is the scaling of the state generation. The use of quantum dot emitters generating high numbers of entangled photons is a promising way to tackle the current challenge of probabilistic linear-optics entangling gates [23]. Future work could address the question of finding optimal network architectures that reduce the overall transmission distance and incorporate such deterministic sources of entangled states.

When extending the network to larger systems, questions for an efficient usage of the states arise: Is it possible to use one resource state for several communicating parties in parallel? Can parts of a state still be used if the state was due to losses not entirely transmitted? Furthermore, a detailed study of the noise occurring in the implementations will be the key to developing adapted protocols.

ACKNOWLEDGMENTS

We thank N. Walk and J. Eisert for useful discussions, and C. Thalacker for setting up the early stages of the experiment. L.R., J.B., and S.B. acknowledge support from the Carl Zeiss Foundation, the Centre for Integrated Quantum Science and Technology (IQ^{ST}), the German Research Foundation (DFG), the Federal Ministry of Education and Research (BMBF, project SiSiQ and PhotonQ), and a the Federal Ministry for Economics and Climate Action (BMWK, project PlanQK). J.d.J. and A.P. acknowledge support from the Emmy Noether DFG Grant No. 418294583. F.H. acknowledges support from the German Academic Scholarship Foundation. A.P. also acknowledges support from the Einstein Research Unit on Quantum Devices.

L.R. and J.B. set up the experiment, measured the data for the implementation of the protocols, and analyzed the data. J.d.J., F.H., and A.P. performed the theoretical analysis, S.B. led the project. All authors discussed the results and contributed to writing and commenting on the paper.

APPENDIX A: SUMMARY OF THE PROTOCOL

The general protocol was introduced in Ref. [15]. There, every party shares in the beginning a Bell pair with each of their two neighbors, meaning every party holds two qubits. In the first step, all parties except Alice and Charlie fuse their two qubits together, which leads to three linear cluster states. In the middle linear cluster state, Alice and Charlie are the first and last party, respectively, which is necessary for the following steps of the protocol to work. The middle state is used for key generation. The outer two states hide the identity of Alice and Charlie. In the next step, the NP parties disentangle themselves from the state by performing either a measurement in the Pauli basis σ_X or σ_Y. The first party decides randomly which basis to choose, and tells the next party the basis. The following parties always take the measurement basis different from the preceding party, leading to an alternating way of disentangling measurements. Alice, Bob, and Charlie measure in different measurement bases, either to perform key generation or verification. Afterward, they perform error correction and privacy amplification.

During all protocol steps, the anonymity of ABC is hidden from all the NP parties: For all possible positions of ABC in the network, there exist two ways for the NP parties to perform the disentangling measurement, and which one they perform is decided uniformly at random. Thus, if a NP party

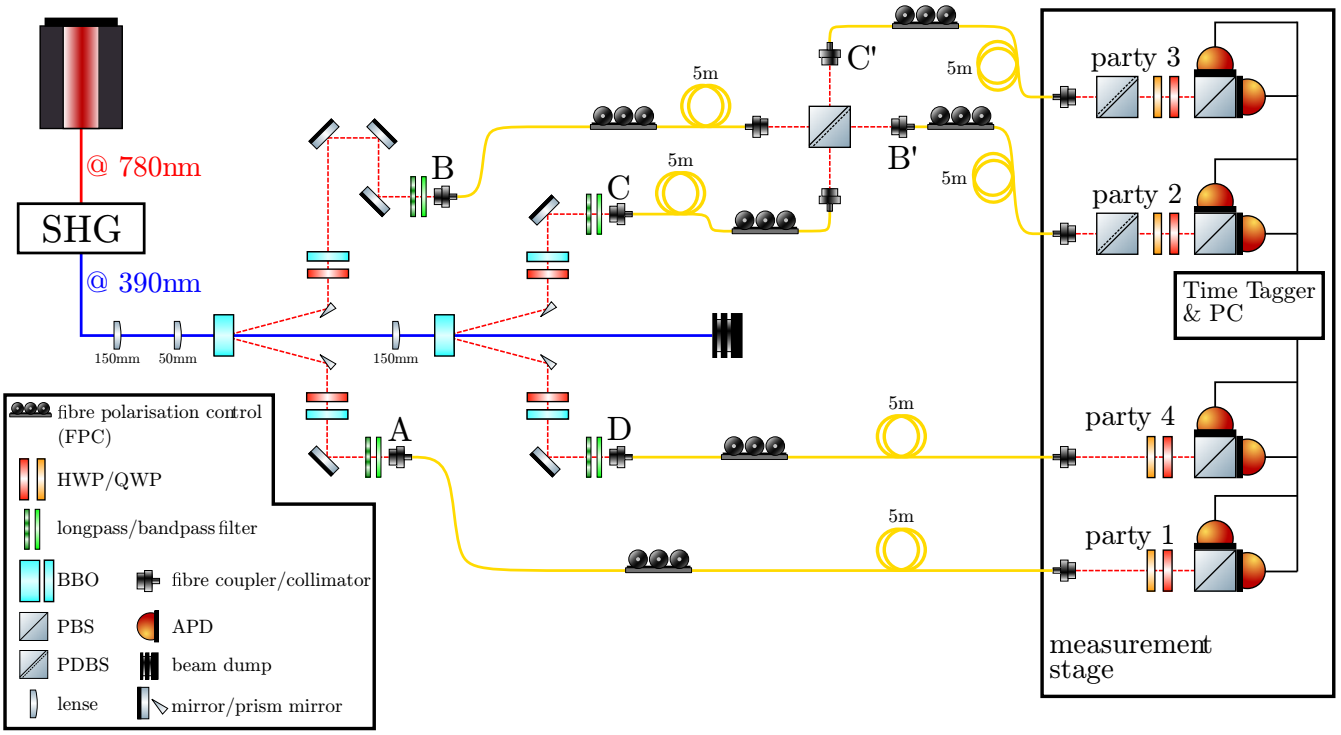

FIG. 6. Experimental setup.

is instructed to perform a measurement in a certain basis, this could belong to any possible positioning of ABC in the network with equal probability, meaning that the party does not gain any knowledge about the positions of ABC. It must also not be possible for the NP parties to find out the measurement bases of the other parties, as ABC measure in bases different from the ones of the NP parties. In principle, they could use their measurement outcomes for this. However, this is not possible due to the no-signaling theorem [24]. Hence, the measurement outcome cannot be used by the NP parties to undermine the anonymity of ABC. All communication needed during the postprocessing steps is encrypted with the preshared key and then broadcasted; the NP parties send random bit strings of the same length. Therefore, it remains hidden which parties actually communicate during these steps and which only mimic this. In this way, the communicating parties ABC remain hidden also during the postprocessing steps.

APPENDIX B: SETUP

In our setup (see Fig. 6), we generate two entangled photon pairs by pumping two BBO crystals cut for type-II-SPDC with a pulsed titanium-sapphire laser ($\tau = 140$ fs, $\lambda = 780$ nm upconverted to $\lambda = 390$ nm, $P = 400$ mW). The generated Bell pairs are fiber coupled and one photon of each Bell pair is overlapped at a PDBS. This PDBS transmits horizontally polarized photons and reflects 2/3 of the vertically polarized photons. Two further PBDSs, one in each output mode of the first, reflect 2/3 of the $|H\rangle$ photons and transmit all $|V\rangle$ photons. All PDBSs together form a photonic CPhase gate upon coincidence detection, postselecting all events which result in one photon per output mode (1/9 of the cases) [20]. For each channel, a combination of a half wave plate (HWP), a quarter wave plate (QWP), and a polarizing beam splitter enables the measurement of the respective qubit in the necessary bases, followed by an avalanche photo diode for photon detection. The setup generates a state which is locally equivalent to a four-photon linear cluster state up to local unitaries which we absorb in the measurement bases.

APPENDIX C: MEASUREMENT SETTINGS OF THE PROTOCOL

The measurement of the NP parties projects the state held by ABC on a state that is locally equivalent to the three-qubit linear cluster state. The exact state depends on the measurement setting of the NP party as well as on the outcome of that measurement. The exact states between ABC are given in Table II.

In our setup, we generated the state

$$|\mathrm{LC}_4'\rangle = \tfrac{1}{2}(|HVHV\rangle + |HVVH\rangle - |VHHV\rangle + |VHVH\rangle), \tag{C1}$$

which is locally equivalent to the state

$$\begin{aligned}|\mathrm{LC}_4\rangle &:= \mathrm{CZ}_{1,2}\mathrm{CZ}_{2,3}\mathrm{CZ}_{3,4}|{+}{+}{+}{+}\rangle \\ &= \tfrac{1}{2}(|{+}00{+}\rangle + |{+}01{-}\rangle + |{-}10{+}\rangle - |{-}11{-}\rangle),\end{aligned} \tag{C2}$$

where $\mathrm{CZ}_{i,j}$ is a CPhase gate acting on qubits i and j. The two states $|\mathrm{LC}_4\rangle$ and $|\mathrm{LC}_4'\rangle$ are related by $|\mathrm{LC}_4'\rangle = (H \otimes \sigma_X \otimes \sigma_X \otimes H)|LC_4\rangle$, where H denotes a Hadamard gate. The measurement settings $\hat{M}_i'$ for the state $|\mathrm{LC}_4'\rangle$ are related to the ones of $|\mathrm{LC}_4\rangle$ by $\hat{M}_i' = (H \otimes \sigma_X \otimes \sigma_X \otimes H)\,\hat{M}_i\,(H \otimes \sigma_X \otimes \sigma_X \otimes H)^\dagger$. The measurement bases for the different configurations of both states are listed in Table III.

TABLE II. Shared state between ABC after the disentangling measurement of NP and measurement settings for the different configurations. $|\pm_i\rangle = (|0\rangle \pm i|1\rangle)/\sqrt{2}$ are the eigenstates to the Pauli matrix σ_Y. For all configurations, the state between ABC is locally equivalent to the three-qubit linear cluster state $|\mathrm{LC}_3\rangle = (|{+}0{+}\rangle + |{-}1{-}\rangle)/\sqrt{2}$. The exact state between ABC depends on the measurement outcome $M = \pm 1 = (-1)^m$ of the disentangling measurement of NP, where $m \in \{0, 1\}$. For the key generation measurement, ABC use a measurement basis where they will get all the same measurement outcome. If m is 1, one party has to perform a bitflip which can be done after the measurement. For the verification measurement, an element of the stabilizer of the state held by ABC is measured. For readability, the notation is changed: Z (X, Y) denotes a measurement in the Pauli basis σ_Z (σ_X, σ_Y) and the index denotes the qubit on which the measurement is applied.

Configuration	State shared between ABC	Key generation measurement	Verification measurement
X_2	$\frac{1}{\sqrt{2}} X_1^m(\|00{+}\rangle_{134} + \|11{-}\rangle_{134})$	$Z_1Z_3X_4$	$X_1X_3Z_4$
Y_2	$\frac{1}{\sqrt{2}} Z_1^m(\|{+_i}0{+}\rangle_{134} + i\,\|{-_i}1{-}\rangle_{134})$	$Y_1Z_3X_4$	$X_1X_3Z_4$
X_3	$\frac{1}{\sqrt{2}} X_4^m(\|{+}00\rangle_{124} + \|{-}11\rangle_{124})$	$X_1Z_2Z_4$	$Z_1X_2X_4$
Y_3	$\frac{1}{\sqrt{2}} Z_4^m(\|{+}0{+_i}\rangle_{124} + i\,\|{-}1{-_i}\rangle_{124})$	$X_1Z_2Y_4$	$Z_1X_2X_4$

APPENDIX D: ERROR CORRECTION PROCEDURE

To correct the errors $\mathrm{Err}(\mathrm{Key}_A, \mathrm{Key}_B)$ and $\mathrm{Err}(\mathrm{Key}_A, \mathrm{Key}_C)$ between the key of Alice and the keys of the other participants, LDPC matrices are used. Using such a matrix, Alice calculates parity check bits from the raw key in a first step. Then Alice sends her parity bits to Bob and Charlie via a classical channel. From the bits sent by Alice and their raw key Bob and Charlie can infer which bits were (most likely) subject to noise and therefore flipped. The identified bits are then corrected by flipping them back.

Specifically, the DVB-S2 standard [25] is applied, providing matrices of the form

$$H_{\mathrm{EC}} = [H'|S], \tag{D1}$$

where H' is a sparse matrix of dimension $(N-k)\times k$ and S is a staircase matrix of dimension $(N-k)\times(N-k)$. Here k refers to the number of information bits, while N is the combined number of parity check bits and information bits. Possible values of N within the standard are 64 800 and 16 200. Due to the length of the created key, the latter one is chosen. Depending on the error rate, a code rate of $r = k/N$ is chosen. A higher r hereby corresponds to less parity check bits and hence is used for low error rates. To correct all errors in Bob's and Charlie's keys, r is set to $1/2$.

Alice divides her key in blocks of k bits and calculates the parity check bits for each block using H_{EC}. Bob and Charlie receive these parity check bits over a classical verified channel. At Bob's and Charlie's site, they now have their respective key which contains errors and the check bits of Alice, which are assumed to be transmitted without errors. With the knowledge of H_{EC}, Bob and Charlie can correct their errors. For this, the `ldpcEncode` and `ldpcDecode` functions provided by MATLAB are used. For the latter, the belief propagation algorithm is chosen [26].

Note that the shortening of the key in the finite case is typically higher than the value of $h(\max(Q^{A,B}_{\mathrm{keygen}}, Q^{A,C}_{\mathrm{keygen}}))$. Similar to the choice of p_{opt}, there exists an optimal choice of the particular error correction scheme for every raw key length L and given noise parameters $\max(Q^{A,B}_{\mathrm{keygen}}, Q^{A,C}_{\mathrm{keygen}})$. The longer the raw key length L gets, the smaller the resulting inefficiency of the optimal error correction scheme becomes. As it can become arbitrarily small, it is not considered in the AKR, which is an upper bound to the FKR.

APPENDIX E: FINITE KEY ANALYSIS

The finite key analysis is done using the following equation:

$$\begin{aligned}\mathrm{FKR} = (1-p)\Bigg(1 - h\Bigg(Q_{\mathrm{verif}} + \mu\left(\frac{\varepsilon_{\mathrm{S}} - \varepsilon}{2}\right)\Bigg) \\ - h\big(\max\left(Q^{A,B}_{\mathrm{keygen}}, Q^{A,C}_{\mathrm{keygen}}\right)\big)\Bigg) - h(p) \\ + \frac{1}{L}(\log_2(\varepsilon^2\varepsilon_{\mathrm{C}}) - 2),\end{aligned} \tag{E1}$$

where

$$\mu(x) = \sqrt{\frac{l+\kappa}{l\kappa}\frac{l+1}{l}\ln\frac{1}{x}}, \tag{E2}$$

TABLE III. Measurement settings for the different network configurations. The configuration is indicated by a letter and an index denoting the measurement basis σ_X/σ_Y and the not participating party, respectively. For each configuration, the measurement settings for parties 1–4 are shown for key generation and verification measurements. For readability, the notation is changed: Z (X, Y) denotes a measurement in the Pauli basis σ_Z (σ_X, σ_Y) and the index denotes the qubit on which the measurement is applied. The measurements are realized by setting the quarter-wave plate and half-wave plate to the corresponding angles. With the notation (angle QWP, angle HWP), these are (45°, 22.5°) for X, (45°, 0°) for Y and (0°, 0°) for Z.

Configuration	Key generation $\|\mathrm{LC}_4\rangle$	Key generation $\|\mathrm{LC}'_4\rangle$	Verification $\|\mathrm{LC}_4\rangle$	Verification $\|\mathrm{LC}'_4\rangle$
X_2	$Z_1X_2Z_3X_4$	$X_1X_2Z_3Z_4$	$X_1X_2X_3Z_4$	$Z_1X_2X_3X_4$
Y_2	$Y_1Y_2Z_3X_4$	$Y_1Y_2Z_3Z_4$	$X_1Y_2X_3Z_4$	$Z_1Y_2X_3X_4$
X_3	$X_1Z_2X_3Z_4$	$Z_1Z_2X_3X_4$	$Z_1X_2X_3X_4$	$X_1X_2X_3Z_4$
Y_3	$X_1Z_2Y_3Y_4$	$Z_1Z_2Y_3Y_4$	$Z_1X_2Y_3X_4$	$X_1X_2Y_3Z_4$

with l being the number of verification and κ the number of key generation rounds, which asymptotically gives $l = pL$ and $\kappa = (1-p)L$. The parameter $\varepsilon > 0$ is a free parameter; one can optimize over ε and p for a given L, Q_{verif}, $\max(Q^{A,B}_{\text{keygen}}, Q^{A,C}_{\text{keygen}})$ and security parameter ε_S. As a usual approach, ε was fixed in our analysis. The factor $(1-p)$ takes into account that only a fraction $(1-p)$ of all rounds key generation is performed. The contribution of the privacy amplification is modified with respect to the asymptotic case to account for statistical effects in the finite case. Also, there is an additional term $h(p)$ for the communication of the round types. For details, see Ref. [15].

[1] C. H. Bennett and G. Brassard, Quantum cryptography: Public key distribution and coin tossing, in *IEEE International Conference on Computers, Systems, and Signal Processing* (IEEE, New York, 1984), pp. 175–179.

[2] A. K. Ekert, Quantum Cryptography Based on Bell's Theorem, Phys. Rev. Lett. **67**, 661 (1991).

[3] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. Dynes *et al.*, The SECOQC quantum key distribution network in Vienna, New J. Phys. **11**, 075001 (2009).

[4] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka *et al.*, Field test of quantum key distribution in the Tokyo QKD network, Opt. Express **19**, 10387 (2011).

[5] J. Dynes, A. Wonfor, W.-S. Tam, A. Sharpe, R. Takahashi, M. Lucamarini, A. Plews, Z. Yuan, A. Dixon, J. Cho *et al.*, Cambridge quantum network, npj Quantum Inf. **5**, 101 (2019).

[6] S. Wengerowsky, S. K. Joshi, F. Steinlechner, H. Hübel, and R. Ursin, An entanglement-based wavelength-multiplexed quantum communication network, Nature **564**, 225 (2018).

[7] F. Grasselli, H. Kampermann, and D. Bruß, Finite-key effects in multipartite quantum key distribution protocols, New J. Phys. **20**, 113014 (2018).

[8] G. Murta, F. Grasselli, H. Kampermann, and D. Bruß, Quantum conference key agreement: A review, Adv. Quantum Technol. **3**, 2000025 (2020).

[9] K. Chen and H.-K. Lo, Multi-partite quantum cryptographic protocols with noisy GHZ states, Quantum Inf. Comput. **7**, 689 (2007).

[10] A. Pickston, J. Ho, A. Ulibarrena, F. Grasselli, M. Proietti, C. L. Morrison, P. Barrow, F. Graffitti, and A. Fedrizzi, Experimental network advantage for quantum conference key agreement, npj Quantum Inf **9**, 82 (2023).

[11] F. Grasselli, G. Murta, J. de Jong, F. Hahn, D. Bruß, H. Kampermann, and A. Pappa, Secure anonymous conferencing in quantum networks, PRX Quantum **3**, 040306 (2022).

[12] F. Hahn, J. de Jong, and A. Pappa, Anonymous quantum conference key agreement, PRX Quantum **1**, 020325 (2020).

[13] M. Proietti, J. Ho, F. Grasselli, P. Barrow, M. Malik, and A. Fedrizzi, Experimental quantum conference key agreement, Sci. Adv. **7**, eabe0395 (2021).

[14] C. Thalacker, F. Hahn, J. de Jong, A. Pappa, and S. Barz, Anonymous and secret communication in quantum networks, New J. Phys. **23**, 083026 (2021).

[15] J. de Jong, F. Hahn, J. Eisert, N. Walk, and A. Pappa, Anonymous conference key agreement in linear quantum networks, Quantum **7**, 1117 (2023).

[16] R. Raussendorf and H. J. Briegel, A One-Way Quantum Computer, Phys. Rev. Lett. **86**, 5188 (2001).

[17] M. Hein, W. Dür, J. Eisert, R. Raussendorf, M. van den Nest, and H. J. Briegel, in *Proceedings of the International School of Physics "Enrico Fermi"*, edited by G. Casati (IOS Press, Amsterdam), Vol. 162, Chap. 5, pp. 115–218.

[18] M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2002).

[19] D. E. Browne and T. Rudolph, Resource-Efficient Linear Optical Quantum Computation, Phys. Rev. Lett. **95**, 010501 (2005).

[20] N. Kiesel, C. Schmid, U. Weber, R. Ursin, and H. Weinfurter, Linear Optics Controlled-Phase Gate Made Simple, Phys. Rev. Lett. **95**, 210505 (2005).

[21] D. F. V. James, P. G. Kwiat, W. J. Munro, and A. G. White, Measurement of qubits, Phys. Rev. A **64**, 052312 (2001).

[22] F. Ewert and P. van Loock, Ultrafast fault-tolerant long-distance quantum communication with static linear optics, Phys. Rev. A **95**, 012327 (2017).

[23] D. Cogan, Z.-E. Su, O. Kenneth, and D. Gershoni, Deterministic generation of indistinguishable photons in a cluster state, Nat. Photonics **17**, 324 (2023).

[24] G.-C. Ghirardi, A. Rimini, and T. Weber, A general argument against superluminal transmission through the quantum mechanical measurement process, Lett. Nuovo Cimento **27**, 293 (1980).

[25] A. Morello and V. Mignone, DVB-S2: The second generation standard for satellite broad-band services, Proc. IEEE **94**, 210 (2006).

[26] R. G. Gallager, *Low-Density Parity-Check Codes* (MIT Press, Cambridge, MA, 1963).