Article:
Solomons, Naomi R., Fletcher, Alasdair I., Aktas, Djeylan et al. (2022) Scalable authentication and optimal flooding in a quantum network. PRX Quantum. 020311. ISSN: 2691-3399

https://doi.org/10.1103/PRXQuantum.3.020311

Scalable Authentication and Optimal Flooding in a Quantum Network

Naomi R. Solomons,[1,†] Alasdair I. Fletcher,[2,†] Djeylan Aktas,[3,‡] Natarajan Venkatachalam,[3] Sören Wengerowsky,[4,5] Martin Lončarić,[6] Sebastian P. Neumann,[4] Bo Liu,[7] Željko Samec,[6] Mario Stipčević,[6] Rupert Ursin,[4] Stefano Pirandola,[2] John G. Rarity,[3] and Siddarth Koduru Joshi[3,*]

[1] *Quantum Engineering Technology Labs and Quantum Engineering Centre for Doctoral Training, Centre for Nanoscience and Quantum Information, University of Bristol, Bristol BS8 1FD, United Kingdom*

[2] *Department of Computer Science, University of York, York YO10 5GH, United Kingdom*

[3] *Quantum Engineering Technology Labs, H. H. Wills Physics Laboratory and Department of Electrical and Electronic Engineering, University of Bristol, Bristol BS8 1UB, United Kingdom*

[4] *Institute for Quantum Optics and Quantum Information—Vienna (IQOQI) and Vienna Center for Quantum Science and Technology (VCQ), 1090 Vienna, Austria*

[5] *Institut de Ciencies Fotoniques (ICFO), Barcelona Institute of Science and Technology, Castelldefels, Barcelona 08860, Spain*

[6] *Photonics and Quantum Optics Research Unit, Center of Excellence for Advanced Materials and Sensing Devices, Ruđer Bošković Institute, 10000 Zagreb, Croatia*

[7] *College of Advanced Interdisciplinary Studies, NUDT, Changsha 410073, China*

(Received 2 April 2021; accepted 16 November 2021; published 18 April 2022)

The global interest in quantum networks stems from the security guaranteed by the laws of physics. The deployment of quantum networks means facing the challenges of scaling up the physical hardware and, more importantly, of scaling up all other network layers and optimally utilizing network resources. Here, we consider two related protocols and their experimental demonstrations on an eight-user quantum network test bed, and discuss their usefulness with the aid of example use cases. First, we consider an authentication-transfer protocol to manage a fundamental limitation of quantum communication—the need for a preshared key between every pair of users linked together on the quantum network. By temporarily trusting some intermediary nodes for a short period of time (< 35 min in our network), we can generate and distribute these initial authentication keys with a very high level of security. Second, when end users quantify their trust in intermediary nodes, our flooding protocol can be used to improve both end-to-end communication speeds and increase security against malicious nodes.

DOI: 10.1103/PRXQuantum.3.020311

I. INTRODUCTION

Quantum key distribution (QKD) is a point-to-point protocol for communication with security based on fundamental laws of physics [1]. Recent advances in quantum networks have enabled two-party QKD protocols to interconnect an increasing number of users [2–6]. Minimization of the resources needed for such networks and optimization of their utilization are essential steps toward their real-world deployment. Several quantum networks trade security for practicality by using trusted nodes to relay the message and/or keys between end users [7–11], while access networks and entanglement-based networks do not rely on trusted nodes for their functionality [2,3,12–15].

There is an often overlooked cost to deploying a quantum network—authentication. Quantum communication assumes that users share a public but authenticated classical communication channel, which requires a preshared secret key. In a fully connected network, every user must maintain a secure database of preshared authentication keys with every other user. As quantum networks grow, this rapidly becomes impractical, to the extent that government cybersecurity agencies' [16] highlights the algorithms used for the inaugural authentication as a major security weakness. The ideal solution for quantum

*SK.Joshi@Bristol.ac.uk

†These authors contributed equally to this work

‡Current address: RCQI, Institute of Physics, Slovak Academy of Sciences, Dúbravská Cesta 9, 84511 Bratislava, Slovakia

networks is to find a practical way for users to establish initial authentication keys on the fly and as needed. The ideal quantum network solution should also have security that is better than any possible classical or postquantum algorithm. Furthermore, large networks need to optimally utilize the communication bandwidth of all available links to maximize network throughput.

Naively, both the above tasks could be solved by using trusted nodes. Authentication keys between two users could be established via referral from mutually trusted nodes. The more mutually trusted nodes used for this referral process, the greater is the security against any one trusted node potentially being malicious. Conversely, instead of using multiple independent paths to improve the final security, we could concatenate the resulting keys to boost the total end-to-end key-generation rate, as is the case in a flooding protocol. Naturally, this is only possible when the end users treat all nodes along each path as trusted nodes. Typically, when end users use trusted nodes in quantum networks, they must then place complete trust in those nodes forever. For long-term data security, this is neither viable nor practical. Is it possible for end users to place partial and/or temporary trust in intermediary nodes in a quantum network?

In this paper, we present two closely linked protocols to address the above question. First, we present a technique to authenticate users in a quantum network with the least number of preshared keys (Sec. III). Second, we present another technique to improve end-to-end key-generation speeds based on flooding, in accordance with previous information-theoretic results (Sec. V B) [17]. We use the definitions of trust and adversaries given in Sec. II. In Sec. V, we examine the utility of these protocols through illustrative use cases, which are demonstrated on our fully connected quantum network. Since any other network topology can be considered as a subgraph of such a fully connected network, our demonstrations are generally applicable to quantum networks. Furthermore, Sec. V A emphasizes the close relationship between these two protocols by demonstrating their use in adding a new user to the quantum network. All the demonstrations are implemented on an eight-user entanglement-based quantum network described in Ref. [3] and in Appendix A 1. We demonstrate a considerable improvement in terms of the scalability, versatility, security, and throughput of the quantum network.

II. PRACTICAL LIMITS ON AND TYPES OF TRUST IN NODES

Practical usage scenarios ultimately dictate the network design and protocols that operate on the network. Consider a typical scenario where various users on the network belong to multiple communities or organizations. Based on their interactions, each user in the network has different categories of trust for all other known nodes. Furthermore, there may be several users unknown to any given user. In a quantum network with information-theoretically perfect security, these considerations are paramount and in a large network it is often more important to assign a level of trust to nodes rather than a binary trusted or untrusted state. We therefore define four categories of trust. Note that a particular node can be assigned different levels of trust from different users:

(a) A **trusted** node (or passive eavesdropper) acts as is required of them within the protocol. They may read communication passed through them but they will not change it or broadcast it publicly (they may still access information that is broadcast publicly).
(b) A **dishonest** node may do everything in their power to interrupt and/or intercept communication between Alice and Bob, including (but not limited to) reading any communication between them and broadcasting the message publicly (potentially without the knowledge of Alice and Bob).
(c) **Partial or temporary trust**. An end user can assign an intermediary node a certain probability of being trusted. This subjective estimate is called partial trust. Users provide the network with this measure based on their experience with other nodes. It is also possible to specify a time duration within which any given intermediary node can be trusted or partially trusted. This is temporary trust.

The case of a node that is completely trusted (and therefore does not read or reveal any communication that they pass on) is trivial and therefore not considered in this work.

In general, we can consider any node or combination of nodes in the network to be an adversary. Additionally, several nodes could conspire together, in which case they can be considered a dishonest single adversary:

(a) **A collective adversary**, c_a. When c_a or fewer nodes in the network are malicious with or without additional external eavesdroppers, they form a single collective adversary with a bound of c_a users. This concept has been introduced in Ref. [18].

III. SECURE INAUGURAL AUTHENTICATION-TRANSFER PROTOCOL

QKD offers provable security [1], ensuring that messages cannot be intercepted or decoded. However, it requires both a quantum channel and an authenticated classical channel, which requires that the users preshare a key. Additionally, the classical hardware or people sending the message can be compromised. A National Cyber Security Centre white paper discussing QKD highlights the dependence on authentication as a major flaw [16].

Thus, the most practical way to deploy a single QKD link (between sites A and B) in the field is for both QKD devices to be initiated with the same one-time authentication key. Then two trusted teams of people must accompany the clean uncompromised hardware to the installation sites A and B and commission them. However, installing a new user into an existing quantum network of n users will require one team to accompany the new hardware and n teams to install new authentication keys with all the existing users.

Here, we present an alternative, where deploying a new node requires substantially less effort. The inaugural authentication key is sent through a trusted intermediary node that has a secure connection to the desired end users. To prevent a man-in-the-middle attack, the node could be monitored in person, but this is necessary only for a short period of time (as illustrated in Fig. 6). Alternatively, as discussed in Sec. III E, a new user only needs inaugural authentication keys to be securely sent to two users in order to be able to initialize a QKD scheme with any other user of the network (assuming that the underlying network is fully connected).

A. Background

Authentication schemes are a method of producing a tag, dependent on the message, and a preshared key. This demonstrates that the sender's message is unchanged and has been sent by the correct user. All classical communication in a QKD protocol must be authenticated; the communication can be public but must not be tampered with by a malicious party [19].

All QKD protocols assume that the users already share authentication keys. For two parties, this is trivial; the users can meet in person to verify their identities and exchange initial keys. However, in a large network, the required number of preshared authentication keys grows quadratically. Besides being impractical, it reduces the functionality of a future quantum Internet if communication is only secure between "known" users.

The Wegman-Carter (WC) authentication protocol [20], which is the most widely employed, uses hash functions. All classical communication between Alice and Bob is "signed" with the appropriate tag and an adversary Eve can only replace the message with her own if she is able to guess an appropriate tag. The authentication is compromised when Eve is able to find another message for which she is able to guess a tag, given her knowledge of a previous message-tag pair. The probability of this happening is given by ϵ, which parametrizes the insecurity.

When using the WC scheme, it is provably impossible for an adversary to have a higher probability of success than ϵ. Therefore, an arbitrary level of security can be achieved, depending on the length of the initial shared key. However, reuse of the same key can pose a security risk [21] and therefore a certain proportion of the key generated in a round of QKD is used in the authentication of further rounds. Section V A further considers the parameters of the WC scheme in a likely experimental implementation.

B. Security of authentication with one trusted node

The distribution of inaugural authentication keys via a third party requires that significant trust be placed in an intermediary node. Here, we demonstrate that such trust need not be permanent. Instead, we show that the intermediary may be restricted to a short window in which to perform an impostor attack. Once this opportunity has passed, the intermediary has no advantage over an arbitrary eavesdropper and standard QKD is sufficient to provide security.

Consider the following three-party scenario. Alice and Bob have both been individually conducting QKD protocols with Chloe. Additionally, Alice and Bob are connected by a quantum channel but initially do not share a key that can be used to authenticate their classical channel. Since Alice and Bob can communicate securely with Chloe, she can be used as a trusted third party, to securely distribute an initial authentication key k_{Auth}. This key is used, following the WC scheme, to authenticate Alice's and Bob's classical communication channel. Alice and Bob can now perform a QKD protocol to grow a new secret key k_{AB}.

This new key is secure against an arbitrary eavesdropper who does not have access to k_{Auth}. However, Chloe could use her knowledge of k_{Auth} to falsify the classical communication during the QKD protocol and perform an impostor attack. If Alice and Bob were to continue using k_{Auth} to authenticate their classical communication, then Chloe would always retain the ability to conduct such an attack. In this case, Alice and Bob must trust Chloe for the entire duration of their communication.

Instead, Alice and Bob can use their new key k_{AB} to authenticate their classical communication. If Chloe can be trusted for the time taken to generate k_{AB}, then Chloe's knowledge of k_{Auth} provides no information about k_{AB}. From this point on, Chloe has no advantage over an arbitrary eavesdropper. It is therefore only necessary to trust Chloe during the distribution of k_{Auth} and for the time it takes for Alice and Bob to generate k_{AB}. After this point, even if Chloe were to become malicious, Alice's and Bob's communication remains secure.

This can be used to communicate the initial key using the secure inaugural authentication-transfer protocol (SIAT):

(1) A trusted node (Chloe) sends the authentication key to two users who wish to communicate (Alice and Bob).

(2) The shared key is used to authenticate Alice's and Bob's channel—once this round of QKD has finished, trust in Chloe ends (the length of the QKD round can be adjusted appropriately based on how long this is considered to be possible).
(3) The key produced in this round is used to authenticate further rounds of QKD.

An important aspect of this protocol is that it is decentralized, allowing users to build connections without the intervention of a network authority (which is of benefit compared to previously suggested methods [22]). Nevertheless, this can be combined with a network authority (using this as the trusted node in every case) if required. However, in the case of this protocol and the protocols discussed in further sections, the topology of the network must be known.

Finally, Ref. [23] describes the security of authentication given partial knowledge of the key. This is not unlikely due to data leakage and could happen if Chloe's data storage is partially compromised. This could potentially lead to an attack in which Eve is able to break the authentication system. However, until she has gathered enough information on the key to be able to falsify messages with a low probability of being detected, this does not lead to knowledge of the key k_{AB} produced. Therefore, if the length of each round is sufficiently short, Alice and Bob are able to generate a new authentication key before Eve is able to eavesdrop. Furthermore, Ref. [23] describes several possible preventions of an attack. Therefore, it is reasonable to assume that WC authentication has the security described.

C. Multiple trusted nodes

Consider the case in which, instead of a single intermediate node that is used to share the initial authentication key between Alice and Bob, there are n distinct paths (of any length) between them and Alice can send a bit string k_i through each node, with $k_{\mathrm{Auth}} = k_1 \oplus k_2 \oplus \cdots \oplus k_n$, where $\oplus$ is defined as bitwise sum modulo 2 (XOR). As shown in Ref. [18], if any number $m < n$ are malicious (but do not publicize their part of the key), they still do not have access to the authentication key, as each k_i provides no information on the total key. If all parties collude, they act as a single malicious party with knowledge of the full k_{Auth}, as above. Reference [18] further considers the case of this network being corrupted by collective adversaries, as defined in Sec. II, in which up to c_a nodes conspire and act as a single common adversary. If the corrupted nodes do not publicly publish their keys (and so keep the key to themselves), it is shown that $c_a + 1$ nonoverlapping paths between users guarantees authenticity (there is guaranteed delivery of unchanged classical messages or a notification of failure). Thus, $c_a + 2$ disjoint paths are needed in the case that the corrupted nodes are dishonest and broadcast the communication publicly.

This protocol would require active use of all of the nodes in the multiple nonoverlapping paths (MNOPs). However, this does not necessarily mean that every node in the network must be used. Any node that is unavailable—for example, due to being in use for other operations or being considered completely untrustworthy—can be discounted from the MNOPs as long as an appropriate topology can be formed without them.

D. Practicality of partially trusted nodes

Trusted nodes are a security risk but their advantages can often outweigh these concerns. Nevertheless, it is impossible to be 100% certain about an intermediary node. If each node is assigned a risk factor, then we can use MNOPs to mitigate this risk. If some probability of insecurity can be tolerated, the insecurity of the initial key is as follows.

Let every node be characterized (in agreement between Alice and Bob or taking the lowest trust value for each) by the partial trust probabilities T_j (as defined in Sec. II). That is, they are honest with probability T_j, although they may read any information to which they have access. With probability $1 - T_j$, they are completely dishonest and may share information publicly. Consider an initial key k_{Auth} to be shared between users Alice and Bob, using several disjoint paths that are labeled i, each with nodes labeled j, as shown in Fig. 1 (similarly to the multiple paths shown in Ref. [18]). The probability that the key is compromised (can be read by a node) is

$$P_{\mathrm{comp}} = \prod_i \left(1 - \prod_j T^{ij}\right) + \sum_k \left[\prod_{j'} T^{k,j'} \prod_{i,i\neq k} \left(1 - \prod_j T^{ij}\right)\right], \quad (1)$$

where k runs over the set of j. The first term represents the scenario that at least one node in every path is dishonest (equivalent to the key in that path being published) and the second term describes the scenario in which at least one node in every path but one is dishonest (meaning that the nodes in the remaining path can infer the key).

There is also the case in which the paths overlap and cross at a particular node. When considering a collective adversary bounded by c_a, this effectively reduces the number of paths. Applied to Eq. (1), the cross-point node can be considered as occupying a position in multiple paths (say, i and i') so that, for example, $T^{ij} = T^{i'j} = T^{xx}$, where xx labels the cross-point. Additionally, we must account for the case in which the cross-point node collates the information from several otherwise trusted channels

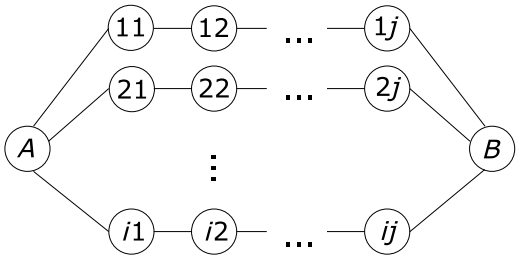

FIG. 1. An example of multiple nonoverlapping paths (MNOPs), demonstrating the labeling of the nodes in MNOPs between A and B.

and at least one node on all other channels is dishonest. This modifies Eq. (1) by adding an additional term $T^{xx}\prod_{\{i\}/x}\left(1-\prod_j T^{ij}\right)$ for each cross-point.

E. Scalability of initial key distribution in a quantum network

The ability to verify the identity of users is necessary to build a secure quantum network. Therefore, whenever a new user joins a network, they must physically receive a secret key (k_{Auth}) that they share with all other users in the network. Thus for a network of n users, this must be done at least $n-1$ times. However, for the new user to be able to have authenticated communication with all of the other users of the network, they need to share a secure initial key with each desired end user and so that number increases depending on the trust in different users, the available resources, and the network topology. As this may require a considerable amount of cost and effort, the number of preshared keys should be minimized.

Consider the scenario where the user being added to the network has full trust in all of the members of the network. Two disjoint paths are needed between the users wishing to communicate, as discussed in Sec. III C. This prevents intermediate nodes having access to k_{Auth} if they become malicious. In the case of a fully connected network with $n>2$, this means that two keys need to be distributed for each new user (as there will be at least two disjoint paths to every other user), so the number of preshared keys required increases as $2n-3$.

As discussed previously, in the case where nodes collude or are corrupted by a collective adversary of c_a nodes, there must be c_a+2 disjoint paths between two users wishing to communicate to distribute the shared key (with the other nodes being trusted). Within a fully connected graph where each user wishes to communicate with every other user, the number of preshared keys (n_k) therefore scales as

$$n_k=\frac{(c_a+2)(c_a+1)}{2}+(n-c_a-2)(c_a+2), \qquad (2)$$

for $n>c_a+1$. Given that c_a scales sublinearly in n, n_k is linear in n, in comparison to the naive solution, which would scale quadratically in n.

A full derivation of this equation is given in Sec. B 1 of the Appendix.

For other topologies, this number will depend on the availability and length of MNOPs that can be found in the network. However, any other topology can be considered a subnetwork of a fully connected network. We assume that the appropriate physical infrastructure exists, so that a fully connected network could first be created with the given number of preshared keys. Then, for any other topology, only some of these secure connections, once established, would need to be used. Therefore, Eq. (2) can be considered an upper bound for any topology.

This only considers the situation in which there are sufficient physical connections to allow this—for more complex topologies with regard to both the physical infrastructure and communications requirements, further calculation would be necessary; however, the given principles [namely Eq. (4) and the SIAT protocol] still hold.

IV. OPTIMAL KEY RATES USING KEY FLOODING WITH TRUSTED NODES

A. Background

Flooding is a classical routing protocol for transmission of information through a multihop network [24]. This strategy has recently been employed in quantum information theory to show that multipath quantum (and private) capacities of a quantum network can greatly outperform corresponding single-path capacities [17]. When a flooding protocol is used in the communication between two users of a network, the source broadcasts a data packet to every user to which it is connected. Each intermediary node then manipulates and outputs the incoming packets on every possible outgoing link except for the one(s) it arrived from. This continues for every node on the network except the receiver. In this way, every link in the network is used exactly once and multiple paths through the network are used in parallel.

Such a protocol has a number of benefits. Each intermediary node does not need to know the full topology of the network, merely the nodes with which it share links. Additionally, flooding protocols are very robust. As long as at least a single path exists between the two end users, communication between them will occur. Flooding is therefore a general protocol that may be applied in any network and we make no assumption on the network topology in describing the protocol, except that it is known.

In general, to implement flooding in a given network, there exists more than one routing strategy. This is best seen when the network is represented by a simple graph $\mathcal{N}(V,E)$, where the set of vertices V represents the users of

the network and the set E represents the edges. Two vertices, $\mathbf{v_i}$ and $\mathbf{v_j}$, are connected by an edge, $(\mathbf{v_i}, \mathbf{v_j}) \in E$, if the corresponding users share a connection. Flooding uses every edge in the network exactly once, so a given flooding routing strategy $\mathbf{p_i}$ corresponds to assigning an orientation to every edge in the network that represents the direction of information flow. For the case of two users communicating, in which one acts as a source and the other as a sink, there is the additional requirement that all the edges from the source are orientated positively and that the edges into the sink are orientated negatively, as is typical in a flow network. Orientation of the edges transforms the simple graph $\mathcal{N}$ into a directed graph $\mathcal{N}_{p_i}$, which represents the ith possible flooding routing strategy on the network. This is depicted for a four-user "diamond" network in Fig. 2. Since each possible directed graph corresponds to a routing strategy and each intermediary edge can be orientated in one of two possible directions, there are $2^{(|E|-|E_s|)}$ possible routing strategies for flooding, where $|E|$ is the total number of edges and $|E_s|$ is the number of edges connected to the source or sink.

In the case of a QKD network, some adaptations must be made. Since we use keys shared between users in the network as one-time pads, there is clearly a maximum amount of information that may be securely transmitted between two users. Therefore, rather than broadcast the full amount of received information, an intermediary node only outputs as much as may be communicated securely to each of its neighboring nodes. Nonetheless, we make use of each edge of the network exactly once and the intermediary users only require knowledge of which users they share keys with and the lengths of those keys.

This strategy of quantum secure flooding is a purely cryptographic formulation of the flooding protocol designed in Ref. [17] to lower bound the multipath quantum and private capacity of a quantum network. The idea is that once entanglement or secret bits are shared between the nodes of a quantum network, the operations of entanglement swapping or key composition (one-time pad) can be combined with an optimal multipath routing strategy [17]. As we can expect from a multipath protocol, quantum secure flooding provides advantages over single-path strategies for communication on a QKD network. In the case that the intermediary nodes are fully trusted, flooding may be used to increase the end-to-end key rate, as we demonstrate experimentally in Sec. V B. In the case that the intermediary nodes are only partially trusted, flooding may instead be used to reduce the risk associated with using these nodes. We demonstrate the latter scenario in Sec. V D.

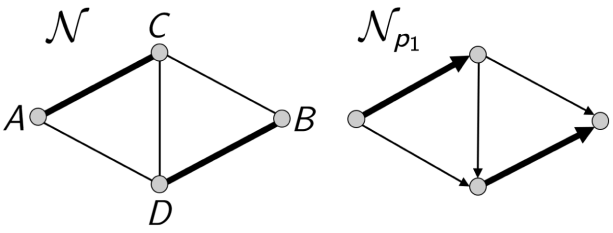

FIG. 2. A comparison of the simple graph representing the network and the directed graph representing a flooding protocol performed on it. The left image depicts the graph of the four-user network $\mathcal{N}$ comprised of end users A and B and two intermediary nodes C and D. A seeks to communicate with B via a flooding protocol. The thickened lines represent higher-throughput connections between A and C and B and D. The first flooding protocol $\mathbf{p_1}$ converts this graph into the directed graph $\mathcal{N}_{p_i}$, which is depicted on the right. The second possible flooding protocol $\mathbf{p_2}$ simply corresponds to reversing the orientation of the edge connecting C and D.

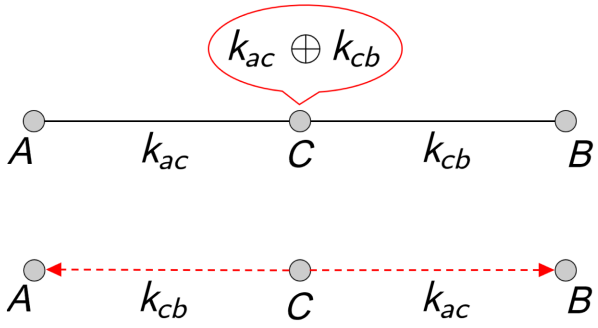

FIG. 3. Simple key flow. The top line shows the intermediary node C announcing the combined key $k_{ac} \oplus k_{cb}$, which is shown in the top diagram. A and B may decode the keys k_{cb} and k_{ac}, respectively. Therefore, knowledge of k_{cb} (k_{ac}) flows to A (B) via k_{ac} (k_{cb}). This is depicted in the bottom diagram.

B. Linear chain

Any multipath protocol may be considered as multiple single-path routing protocols taking place simultaneously. It is therefore natural to first consider how two users may share a key using a single-path protocol.

The simplest network with a single path between two users is a three-user chain network. We consider, as an example, three users—Alice (A), Bob (B), and a trusted intermediary node (C)—in which Alice and Bob do not directly share a key but, instead, both share a key with the intermediary node. For simplicity, we consider the length of the keys k_{ac} and k_{cb} to be equal. The intermediary node makes a public announcement of the bit-wise sum modulo 2 of the two keys $k_{ac} \oplus k_{cb}$. Alice and Bob can decode each other's key by again performing bit-wise addition of their own key with the announced combined key. Knowledge of k_{ac} flows from C to Bob via k_{cb} and correspondingly knowledge of k_{cb} is passed to Alice via k_{ac}. This is illustrated in Fig. 3.

We now make the following observations. Provided that the intermediary node is trusted, the announcement is made correctly and an adversary only has access to the combined key $k_{ac} \oplus k_{cb}$, which provides no information about either of the individual keys. However, Alice and Bob may not concatenate the two keys to generate a shared key of twice the length, as the announcement provides sufficient information to determine one key from the other. It is also clear that if the original keys are of different lengths, then the maximum length of secure key that Alice and Bob may share is equal to the length of the shorter key.

The natural extension to n trusted intermediary nodes also reduces to the above example. Each node learns all the keys used in announcements along the chain and a general adversary only has access to the bit-wise sum of any of these keys. Thus, while Alice and Bob learn all the keys along the chain and these keys remain secure, they can only use one key, which must be predetermined. Therefore, the maximum length of secure key that can be generated between the two end users is equal to the length of the shortest key in the chain and corrupting a single user gives an adversary access to all the keys in that chain. Thus, given partial trust in each of the intermediaries, more users increases the probability of insecurity. We can instead increase the security and rate of the protocol by introducing multipath strategies in more complex networks, as we now detail.

C. Quantum secure flooding

The quantum secure-flooding protocol is the same regardless of whether the end users wish to increase their end-to-end communication rate or the security of their communication. In both cases, the end users share multiple secure keys via multiple paths (i.e., via different sets of other partially or fully trusted nodes). The difference arises at the end of the protocol, when the end users privately either concatenate keys (in order to increase the rate) or XOR keys (to increase security).

These scenarios are discussed in Secs. V B and V D, respectively. Throughout this section, we assume that the communication is between the end users, Alice (the source) and Bob (the sink). It is important to note that, in this protocol, it is necessary for Alice to have complete knowledge of the current network topology and available and/or unused key rates between users.

As discussed in Sec. IV B, keys are passed through the network by announcing them XORed with other network keys. Since a key may only be used once as one-time pad, an intermediary node should not simply pass all of their received keys out to every other node with which they share a connection. Instead, an intermediary node must split its received keys only among the users with which it wishes to communicate.

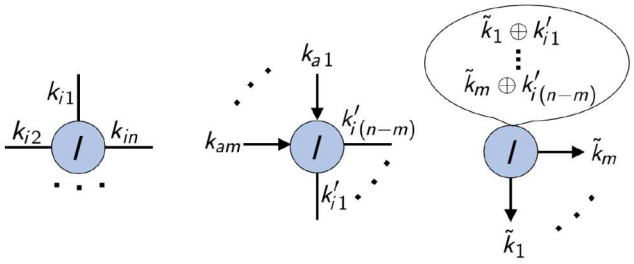

FIG. 4. The procedure for splitting keys during quantum secure flooding. The left diagram depicts the initial neighborhood of user I, which shares keys with n other users. The middle diagram shows the user when it is contacted during step (4) of the quantum secure-flooding protocol. The user receives m keys and has $n-m$ remaining keys. The right diagram shows the node having split its keys according to the received ordering and announcing these keys XORed with the remaining $n-m$ keys. Since all of the remaining keys are used, this corresponds to the case in which the total length of keys received is greater than the length of the remaining keys.

To illustrate this, we consider as an example an intermediary node I that shares keys with n other nodes as illustrated in Fig. 4. The intermediary receives a set of m keys $\{k_{a1}, \ldots k_{am}\}$ with corresponding lengths $\{l_{i1}, \ldots l_{im}\}$, either directly from the source or decoded from another intermediary's announcement. The intermediary therefore has $n-m$ keys that have not yet been used, nor are keys shared with the source. The intermediary node privately concatenates all of its received keys into the combined key $\tilde{k} = k_{a1}||\cdots k_{am}$. The node also receives an ordering for their remaining $n-m$ keys, resulting in the ordered set $\{k'_{i1}, \ldots k'_{i(n-m)}\}$. Each of these keys also has a corresponding length $\{l'_{i1}, \ldots l'_{i(n-m)}\}$. The intermediary node then splits $\tilde{k}$ into separate keys $\tilde{k}_i$ with lengths equal to l'_i until the entire key is used or all the m unused keys have matching length keys. I then announces each of these keys XORed with the corresponding unused key k'.

In more detail, the steps of the protocol are the following:

(1) The optimal flooding protocol is calculated by Alice as detailed in Appendix A 4. In general, this can be achieved in $\mathcal{O}(|V||E|)$ time, for a network with $|V|$ nodes and $|E|$ edges [44].
(2) Alice contacts the first intermediary node, sending them an ordered list of output keys.
(3) The intermediary user splits the keys that they share with Alice according to the ordered list they have received. They announce these keys XORed with their remaining keys in accordance with the received ordering.
(4) Alice contacts the next intermediary node and sends them an ordered list of output keys. They privately decode any keys they can from previous

announcements. They concatenate these keys with any keys that they shared directly with Alice and then split the resulting key in accordance with the ordered list that they have received. These keys are announced XORed with their remaining keys in accordance with the received ordering.

(5) Step (4) is repeated for all the required intermediary nodes in the protocol.

Figure 5 illustrates the optimal quantum secure-flooding protocol applied to an idealized network shown in Fig. 2.

Calculation of the optimal flooding protocol requires full knowledge of the network topology. It may be possible for malicious users to convince others that actually overlapping paths do not overlap. We restrict our work here to the case in which the topology can always be determined. We note that all users in our network share entanglement; thus they can distinguish between a direct link with another user as opposed to a link via a trusted node. This may enable users to cooperatively and securely map out the network topology despite the presence of some malicious users.

V. PRACTICAL APPLICATIONS OF SIAT AND KEY FLOODING

A. Adding a new user to the quantum network

To show the feasibility of the SIAT protocol for adding a new user, we demonstrate it applied to the example of a previous experiment of an eight-user quantum network, with the data shown in Ref. [3].

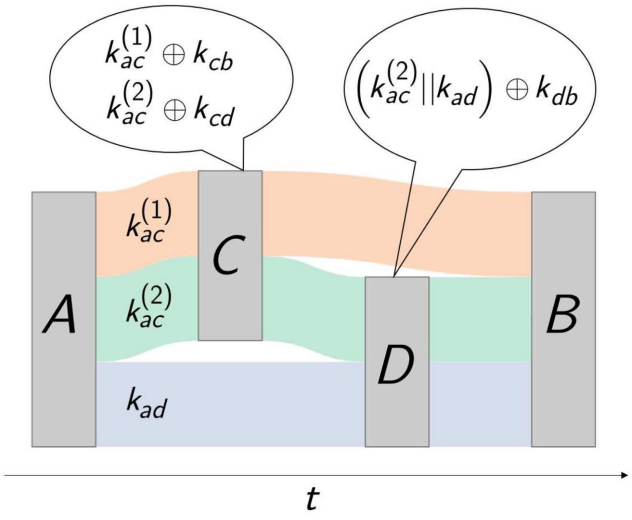

FIG. 5. Key flow during flooding. The diagram depicts the time at which announcements are made and the flow of knowledge of A's keys through the network depicted in Fig. 2. Each color depicts the flow of knowledge of one of A's keys (after splitting). Node C splits their longer key k_{ac} into two keys of equal length, $k_{ac}^{(1)}$ and $k_{ac}^{(2)}$. They then publicly announce $k_{ac}^{(1)} \oplus k_{cb}$ and $k_{ac}^{(2)} \oplus k_{cd}$, causing $k_{ac}^{(1)}$ to flow to B and $k_{ac}^{(2)}$ to flow to node D. Node D concatenates $k_{ac}^{(2)}$ and k_{ad} and announces $(k_{ac}^{(2)} || k_{ad}) \oplus k_{db}$, causing $k_{ac}^{(2)}$ and k_{ad} to flow to B.

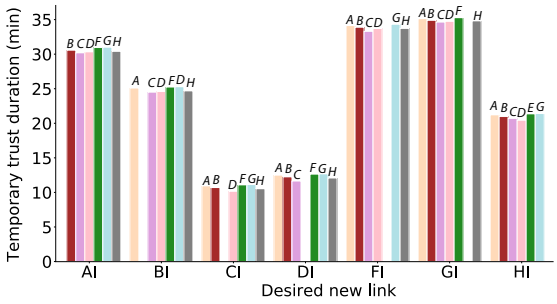

FIG. 6. We show that initial authentication, while adding a new user to the network, can be established in less than 35 min. The figure corresponds to a demonstration in which a fully connected network of seven users exists and a new user—I—wishes to join the network using the SIAT protocol. The time in any other user must be trusted, when used as an intermediary node in order to connect I to different end users is shown. The end users are shown on the x-axis and intermediaries are denoted by different colours and labelled above the bars. The data used are taken from Ref. [3].

The amount of classical communication required in order to produce a bit of key varies between experiments (as opposed to the factor of 2 in ideal QKD). In this experiment, there are up to 10 000 detection events for each bit of secure key, which are labeled with 64 bits of time tagging and 2 bits for the basis choice. This is increased to 72 bits to label each event when considering error-correction data, so an estimate for the amount of classical communication required is 720 000 bits per bit of key. For an insecurity of 10^{-9}, with 10% of the key reused each round for authentication, this means that successive QKD rounds should each produce 50 563 bits, as shown in Sec. B 2 of the Appendix.

The example case is considered in which a new user is added to the network and wishes to form a secure connection with each other user of the network. In this example, this will be considered to be the eighth user (I). If each of the users (A, B, C, D, F, G, and H) is used as a trusted intermediate node, Fig. 6 shows the amount of time needed to initialize a secure authenticated channel with any other user. This shows that given the simplest scheme, the maximum amount of time a user needs to be trusted is less than 35 min. Therefore, the length of time needed to initialize the authentication in a quantum network should not be considered a significant disadvantage.

However, this scheme can be improved further by utilizing the previously discussed flooding protocol.

B. Optimal key generation using trusted nodes

The flooding protocol described in Sec. IV C can be used to increase the end-to-end secure key rate. In the case that all the intermediary nodes are trusted (which means

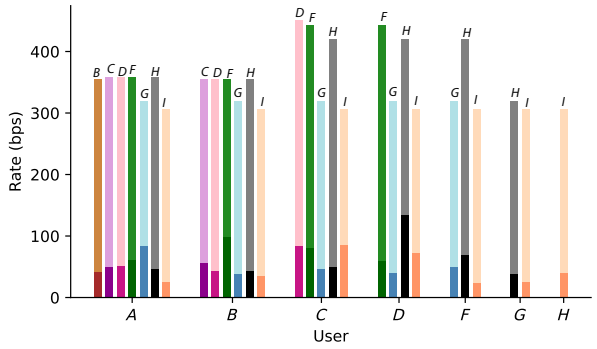

FIG. 7. Improved secure key rates with end-to-end key flooding on our eight-user quantum network test bed. The direct key rates (bold colors) and the flooding secure key rates (lighter colors) are shown between every pair of end users in the experiment (labels above the bar indicate the user connected to). Using the flooding protocol, a fully connected network can be made exactly equivalent to an access network with improved rates, where an optical switch chooses which end users can communicate at any instant.

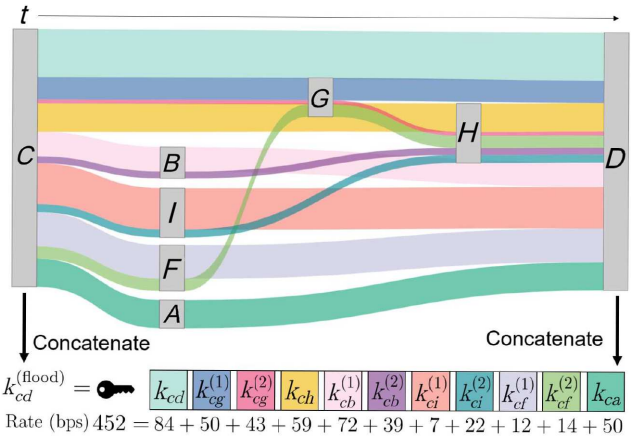

FIG. 8. The optimum end-to-end key flooding protocol between end users C and D on our eight-user quantum network test bed. The figure demonstrates an optimal flooding protocol between users C and D corresponding to the data used in Fig. 7. For clarity, the public announcements made by the intermediary users are suppressed and shown instead in Sec. B 3 of the Appendix. The width of the bars indicates the length of the keys ultimately shared between C and D. The end users privately concatenate the shared keys to produce $k_{cd}^{(\text{flood})}$ with length equal to the sum of the lengths of the shared keys. This achieves the optimal flooding rate shown in Fig. 7.

that we are not restricted to nonoverlapping paths), the two end users may concatenate all the flooded keys into a new longer key.

We use data from the quantum network test bed described in Appendix A 1 to demonstrate the increase in the end-to-end key rate between all possible pairs of users. Data from all users are collected every 20 min and the entire QKD postprocessing is performed, including error correction and privacy amplification. Flooding is performed using the final secure keys. We note that it may also be possible to perform flooding on the raw or sifted keys before privacy amplification. This may result in a higher end-to-end key rate due to the nonlinearity of the key rate with the quantum bit error rate (QBER). Further work is required to consider all the security implications and study the potential improvements. Such a study should also account for inefficiencies during the implementation of error correction and privacy amplification. Figure 7 shows the optimal flooding rate for each possible pair of end users compared with the direct rate between them for a single 20-min block. On average, we are able to demonstrate an approximately sevenfold increase in the key rate on our quantum network test bed. Figure 8 shows the corresponding optimal flooding protocol between the users Chloe (C) and Dave (D). Further details are provided in Sec. B 3 of the Appendix.

C. Enhancing security using flooding

In an arbitrary network, suppose that there are MNOPs between two end users A and B; then, the end users can choose to use the flooding protocol to maximize the secure key rate. However, this is not the only optimization they can perform. Here, we show that flooding can also be used to improve the security between the end users.

As discussed in Sec. IV B, a path through a network can be treated as a linear chain. All of the intermediary nodes in the chain learn the secure key that is passed from one end user to the other. The communication is secure along the chain as long as all the intermediary nodes are trusted. If we now consider a path i with N such intermediary nodes, each partially trusted with a trust value T_j, then the probability that the communication along the chain is secure is $T_i = \prod_j^N T_j$. Using single-path routing strategies, the maximum security that can be achieved is simply the security of the most secure single path: $t = \max_{\forall i'} T_{i'}$.

However, the two end users may improve the security of their final key by first performing a flooding protocol and receiving a set of n keys $\{k_1, k_2, \ldots k_n\}$. Each of these keys corresponds to a path through the network used in the flooding protocol and may be assigned a trust as described above. The two end users may now privately XOR all of these keys, resulting in the final key $k_{AB} = k_1 \oplus k_2 \oplus \cdots k_n$. In the case that all of the paths used by the flooding protocol are nonoverlapping, the overall trust value t can be derived from Eq. (1) and is given by

$$t = 1 - \left(\prod_i^n (1 - T_i) + \sum_k T_k \prod_{i\ i \neq k}^n (1 - T_i) \right). \quad (3)$$

This follows immediately from the fact that for nonoverlapping paths, the protocol is compromised if either all or

all but one of the paths contains a dishonest intermediary node. The more general case in which some of the paths overlap is also discussed in Sec. III D.

The above process results in a final key with length equal to the minimum length of the XORed keys. However, it is also possible to consider the scenario in which the end users wish to improve both the security and the rate. In this case, after the flooding protocol, the end users partition their set of keys into subsets, indexed by x, each comprising at least three keys, which are subsequently XORed. Each of the resulting keys is then concatenated into a final key. This concatenation further reduces the trust in the final key, since all of the combined keys must now individually be secure. In the nonoverlapping case in which there are m subsets, each comprising n' keys such that $n'm = n$, the final trust value is given by

$$t = \prod_{x}^{m}\left[1 - \left(\prod_{i}^{n'}(1 - T_{ix}) + \sum_{k} T_{kx}\prod_{i\ i\neq k}^{n'}(1 - T_{ix})\right)\right]. \tag{4}$$

We consider as an example the case of a network with $N + 2$ users. The network is fully connected except that users A and B do not share a connection. Each of the shared keys between the users is assumed to have the same length (which we normalize to 1) and each of the N intermediary users has the same partial trust T. The optimal flooding protocol (either for optimizing the key rate or security) consists of N nonoverlapping paths of the form $A \to I \to B$, where I is an intermediary user. In this scenario, Eq. (4) simplifies to the following:

$$t = \left[\sum_{k=0}^{n'-2}\binom{n'}{k}T^{n'-k}(1 - T)^{k}\right]^{m}. \tag{5}$$

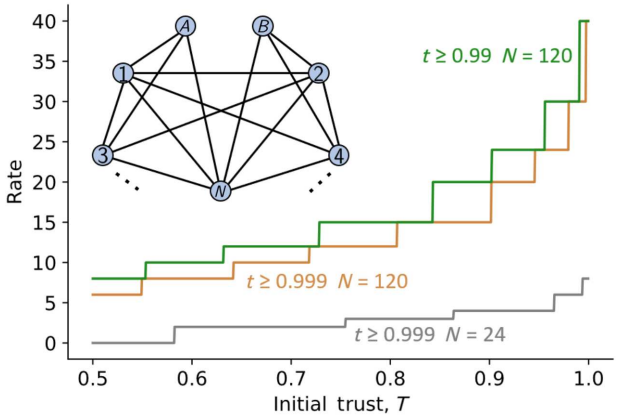

FIG. 9. The trade-off between trust and the rate. The figure corresponds to the hypothetical scenario in which all of the shared keys have the same length (normalized to 1) and all N intermediary nodes have the same trust T. The lines show the maximum rate achievable such that the total trust t is greater than or equal to a given value. The inset shows the $N + 2$ user network, which is fully connected except that there is no connection between A and B.

Figure 9 demonstrates the trade-off between the rate and trust for such networks with a variety of network sizes and required trust values.

D. Enhancing security in a realistic quantum network

The previous example of implementing the SIAT protocol (as illustrated in Fig. 6) assumes a completely trusted third party. In the case of using multiple partially trusted nodes, flooding can be used (as highlighted previously) to increase the security. It is possible to use flooding to implement the SIAT protocol, in the case that the intermediate nodes are not fully trusted, as we now show.

Consider the example scenario shown in Fig. 10. Ivan has been on the network for a while without needing to communicate with Alice but has been exchanging QKD keys with several other nodes over the network. Ivan publicly announces all (except those he wishes to keep secret for whatever reason) nodes with which he shares established QKD links (i.e., a preauthenticated classical channel and a quantum channel). Alice does the same. In this example, they mutually identify six nodes that they have in common. Alice and Ivan both have perceptions of how much they personally trust each of these six nodes. These partial-trust values for both Alice and Ivan are shown in Table I.

Through the following steps, we can implement a combined SIAT and flooding protocol (i.e., "flood the SIAT protocol") thereby increasing the security of the SIAT protocol:

(1) Alice and Ivan exchange an inaugural authentication key, via the SIAT protocol, using each of the six mutual peers. They then XOR these keys to ensure that the inaugural authentication is secure as long as at least two of their mutual peers remains honest (this uses a relaxed definition of "honest" as previously described—we note that security in our protocol would be guaranteed with a minimum of one ideally trustworthy node). This establishes the dashed link in Fig. 10.
(2) Alice and Ivan share their trust tables (i.e., their partial trust values in each of the other nodes). This can be done publicly or privately using the newly established secure channel.
(3) The minimum of the two trust values assigned by Alice or Ivan for an intermediary node is chosen as the partial trust for that node.
(4) Alice and Ivan agree upon a minimum end-to-end trust value; say, 99.25%.

TABLE I. A table demonstrating the use of flooding to improve key rates and security in a realistic network scenario. Key rate data are taken from our eight-user fully connected quantum test bed and the trust values for the intermediary nodes from Fig. 10. All possible partitions of the six intermediary nodes that can simultaneously boost security and rate are shown. Of these, the row highlighted in green provides the best improvement to both security and end-to-end key rate. *A* and *I* will also combine their keys obtained via a direct link to further improve the throughput.

First Subset		Second Subset		Combined	
Intermediaries	Rate (bps)	Intermediaries	Rate (bps)	Trust (%)	Rate (bps)
B C D	34.29	F G H	25.31	99.42	59.60
B C F	25.31	D G H	39.57	99.25	64.88
B C G	34.29	D F H	25.31	99.42	59.60
B C H	34.29	D F G	25.31	99.42	59.60
B D F	25.31	C G H	39.57	99.34	64.88
B D G	34.29	C F H	25.31	99.48	59.60
B D H	34.29	C F G	25.31	99.34	59.60
B F G	25.31	C D H	39.57	99.34	64.88
B F H	25.31	C D G	48.33	99.48	73.65
B G H	34.29	C D F	25.31	99.34	59.60

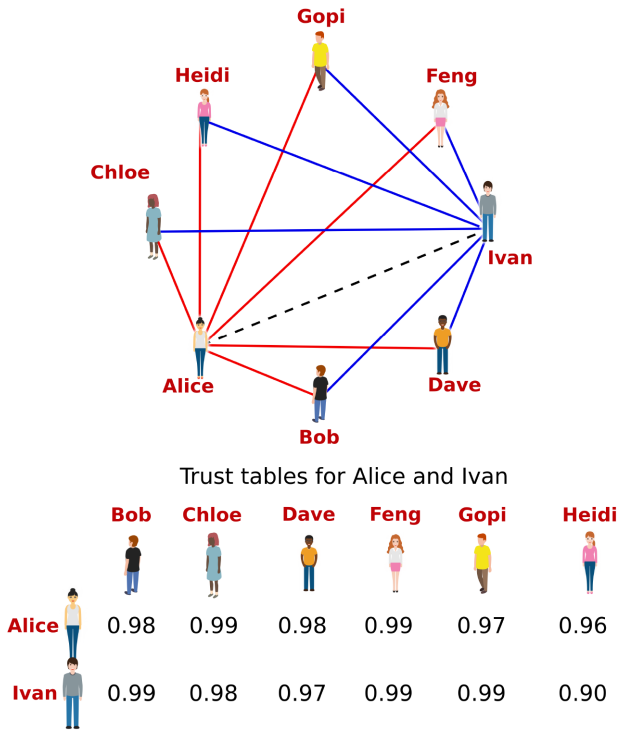

FIG. 10. An example scenario where Alice (A) does not have a preshared authentication key with Ivan (I). Each user has several preshared keys with various other users in the network and the amount of trust Alice or Ivan is willing to place in other users is given in the trust table. All links not shown in this example are currently being utilized for other purposes and are thus not available.

(5) The key flooding protocol is implemented and keys from multiple sets of MNOPs are XORed together until they meet or exceed the minimum end-to-end trust value. If multiple sets of MNOPs can produce keys that exceed the desired trust value, then these keys are concatenated together. A simple algorithm (see Appendix A 4) is used to evaluate every possible combination of MNOPs to ensure the best possible final end-to-end key rate given the desired trust threshold.

Using this realistic example, we can see that flooding and the SIAT protocol can easily be combined in order to maximize the efficiency and security capabilities of a quantum network. Possible final key rates between end users, and their trust in the security of the key, are shown in Table I for the different ways of separating the six mutual peers into two subnetworks.

VI. CONCLUSION

The building of national or international quantum networks with several users is a labor-intensive and costly process. Therefore, it is important to ensure that any network we build can easily be expanded. Furthermore, the use of trusted nodes in a quantum network is a convenient and effective way to build long-distance quantum communication networks. However, the risk that an individual trusted node is compromised remains, so placing absolute trust in any one intermediary node is a serious security flaw. Many quantum networks and communications techniques therefore seek to eliminate trusted

nodes. This can be achieved by using measurement-device-independent QKD [25,26]. resorting to twin-field techniques [27,28] (potentially overcoming the secret key capacity for repeaterless QKD protocols [29]). using entanglement distribution [30,31]. or by using quantum memories or repeaters [32,33]. However, these solutions are not always applicable.

We demonstrate a set of algorithmic solutions to growing quantum networks that, provided that the network is large enough and sufficiently well connected, can be implemented the same way regardless of the size of the network. Following work in Ref. [18], we use the concept of partially and temporarily trusted nodes, employing MNOPs to mitigate the associated security risks.

The SIAT protocol uses trusted nodes to distribute authentication keys to initiate new links, removing the need to physically transport these keys. It shows that, in the case of a well-connected network, only $\mathcal{O}(n)$ key stores are required, as opposed to $\mathcal{O}(n^2)$. It is compatible with a peer-to-peer-like referral scheme or with centralized authorities. We experimentally implement this protocol on our eight-user quantum network test bed and show that, using just one trusted intermediary node, the SIAT protocol takes between 10 and 35 min to execute. The SIAT protocol can be used in conjunction with classical and/or postquantum authentication protocols such as those implemented in Ref. [22]. When using any protocol based on computational security to perform the initial authentication, the probability that such an algorithm can be broken, within the time taken for authentication, upper bounds the value of the partial trust placed in that node.

The SIAT protocol is combined with a flooding protocol, showing how best to calculate which MNOPs to use in any network. We show that the larger the network, the larger is the gain in security from using MNOPs. The same techniques can be used to improve the end-to-end security and/or the key-generation rate between any two end users. In many practical scenarios, as illustrated by the examples we provide, users can choose a good compromise that improves both the security and the final key rate. Furthermore, the flooding protocol can be used to optimize the network performance by using idle network resources to boost network throughput. Additionally, by deploying the protocol on our fully connected network, the requirement that an end user know the full network topology and keys lengths is mitigated. So long as there are sufficiently few dishonest users, monogamy of entanglement allows the end users to detect any false reporting of the network topology. In a real-world implementation, users could choose to share information about how much they trust other users openly, via encrypted private channels or using quantum secure anonymous protocols (as in Ref. [34]). Recently, it has been suggested that redundant QKD devices, within each individual node, can help overcome certain security concerns [35]. Our flooding protocol can also be used together with such redundant devices to further optimize both the security and the key rate.

Together, these two protocols represent a means for quantum networks to be deployed with ease and grow organically according to end-user requirements. These protocols are most effective in large densely connected quantum networks where several available MNOPs are likely to exist. The SIAT and flooding protocols presented here can be used in conjunction with most types of QKD protocols (including continuous-variable implementations) and hardware as long as MNOPs exist. In several cases, trusted nodes are the most effective solution and access to these nodes or the amount of memory they have for key stores can be very limited. Quantum communication cube satellites are a good example. They are a cheap and effective way to link quantum networks across the globe and almost all such efforts use the satellite as a trusted node [36–38]. When building satellite constellations, physical access to all optical ground stations to share an initial authentication key is impractical. These protocols would allow authentication transfer between satellites, increase trust levels, and help to optimally route key-generation traffic based on link availability, achieving optimal end-to-end performance for global quantum networks [39,40]. Further refinements to these protocols, and secure key storage, would provide complete security solutions for quantum communication networks.

ACKNOWLEDGMENTS

The research leading to this work has received funding from United Kingdom Research and Innovation's (UKRI) Engineering and Physical Science Research Council (EPSRC) Quantum Communications Hub (Grants No. EP/M013472/1 and No. EP/T001011/1) and equipment procured by the Quantum Photonic Integrated Circuits (QuPIC) project (EP/N015126/1). We also acknowledge the Ministry of Science and Education (MSE) of Croatia, Contract No. KK.01.1.1.01.0001. We acknowledge financial support from the Austrian Research Promotion Agency (FFG) ASAP12-85 project and the SatNetQ 854022 project. S.P. acknowledges support from the European Union via "Continuous Variable Quantum Communications" (CiViQ, Grant Agreement No. 820466). N.R.S. was funded by the EPSRC through the Quantum Engineering Centre for Doctoral Training, EP/SO23607/1. A.F. was funded by the EPSRC via a Doctoral Training Partnership, EP/R513386/1. We would like to thank Thomas Scheidl for help with the software used to run the original experiment and Mohsen Razavi and Guillermo Currás Lorenzo for their help in proving the security of the implementation of the original network experiment.

N.R.S. and S.K.J. conceived the authentication protocol, A.F. and S.P. implemented the flooding protocol. N.R.S., A.F., S.K.J., D.A., and S.P. explored the relationships between these two protocols and the use cases. N.V. and B.L. wrote the postprocessing software. S.K.J., D.A., S.W., M.L., S.P.N., B.L., Z.S., M.S., and R.U. performed the quantum network experiment. S.K.J., S.P., and J.G.R. supervised the project and contributed to the ideas. S.K.J. was the team leader. The paper was written by N.R.S., A.F., D.A., and S.K.J. and all authors read and contributed to improving it.

APPENDIX A

1. Our quantum network test bed

We use our entanglement-based quantum network with eight users [3], spread amongst multiple university buildings, to implement and test new protocols. The quantum network architecture is best understood when divided into different layers of abstraction as shown in Fig. 11. The "physical layer" is comprised of hardware necessary to generate, distribute, and detect the entangled states, thus forming the actual infrastructure. In this layer, our implementation only requires one single fiber between each user and the (de)multiplexed source, while in the logical or connection layer, the topology naturally forms a fully connected graph between all possible pairs formed by the users in the network. We use one source of polarization-entangled photon pairs and a combination of standard telecom dense wavelength-division multiplexers (DWDMs) together with in-fiber beam splitters (FBSs) in order to distribute bipartite entangled states between all eight users.

The multiplexing strategy serves the purpose of fully interconnecting eight users while only using 16 wavelength channels with eight FBSs, thus optimizing the transmission and entangled-state fidelity per channel. Every user is provided with a polarization-analysis module that performs a passive-basis choice using a bulk beam splitter (BS), a half-wave plate (HWP), a polarization beam splitter (PBS) and two single-photon detectors [3]. This enables every user to measure in the horizontal- or vertical-polarization basis or in the diagonal- or antidiagonal-polarization basis when the photons go through the long path with an HWP. Our eight users, referred to as Alice (A), Bob (B), Chloe (C), Dave (D), Feng (F), Gopi (G), Heidi (H), and Ivan (I), are comprised of two subnets of four users, where each subnet uses wavelength multiplexing to fully interconnect its members—A, B, C, and D. The use of FBSs on each of those wavelength channels allows us to duplicate the subgroup, creating another set of four fully interconnected users—F, G, H, and I.

Finally, we require two additional pairs of wavelengths to connect the remaining links between users, AF, BG, CH, and DI, across the two subnets. Any pair of users in the network can perform its own standard BBM92 protocol [41], where all detected photons from other possible users can be considered as background noise. By choosing a narrow coincidence window (typically approximately 130ps) which can be optimized in postprocessing, one can ensure that this noise only increases the QBER marginally. With this setup, we can generate secure keys between all 28 possible combinations of paired users. Some extra wavelengths remain, allowing us to form four additional links increasing the secret key rate for some selected users.

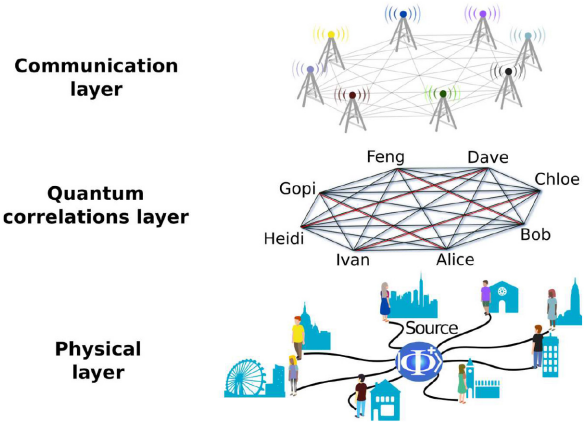

FIG. 11. The network layers. The physical layer represents the physical hardware, including the photon pair source and the optical components necessary to multiplex and distribute the photons toward the users, together with their detection-analysis modules. The quantum correlation layer represents the sharing of entanglement between users and the communication layer is where the measurement results are processed to implement QKD, flooding, and/or the SIAT protocols.

2. The network service provider

We use data from the experiment described in Ref. [3]. Our network architecture relies on a quantum network service provider (QNSP) to distribute bipartite polarization-entangled states to users with polarization analysis modules (PAMs). The QNSP is comprised of an entangled-photon-pair source employing a Sagnac scheme and a multiplexing unit (MU). The Sagnac source consists of a 5-cm-long magnesium-oxide-doped periodically poled lithium niobate (MgO:ppLN) bulk crystal, with a poling period of 19.2 μm, which is pumped by a cw laser emitting at 775.1085 nm in both directions inside the Sagnac loop. The input-output of this loop is defined by a dichroic mirror and a PBS, allowing for diagonally polarized pump light to split and propagate the horizontally (vertically) polarized part anticlockwise (clockwise) inside. A HWP after the PBS transmission port is used to set the pump light in the anticlockwise to vertical and also allows for

the $|V_s\rangle|V_i\rangle$ signal and idler photon pairs generated through type-0 spontaneous parametric down-conversion in the other direction to become $|H_s\rangle|H_i\rangle$. This makes it possible for both contributions in the loop to coherently recombine at the PBS and exit isolated from pump light by the dichroic mirror. The $|\Phi^+\rangle$ Bell state generated spans 32 channels in the *C* band. The distribution of these 32 channels (as defined by the International Telecommunication Union in G.694.1) in each of the fibers is achieved by our MU, consisting of a set of eight standard 50:50 fused couplers with insertion loss below 3.4 dB, together with 16 add-or-drop thin-film DWDMs showing 0.5 dB insertion loss and a channel spacing and nominal full width of 100 GHz. Fiber polarization controllers (FPCs) are used to ensure that the reference frame of polarization in the source is (nearly) identical to that of the PAM.

3. Data generation and analysis

The data are generated during the operation of the quantum network over a span of several days. The duration of each data collection run is limited for logistical reasons and/or the hold time of the cryostat used for the detectors (approximately 18 h). At the beginning of each data collection run, we polarization neutralize all fibers. For fibers that carry several wavelengths, they are neutralized at their central wavelength. We also tune the state produced by the source to minimize the QBER for one of the connections. Signals from all 16 detectors are collected by a Swabian Instruments time tagger and these data are stored. While processing the data, each user's counts are extracted from the data file and stored in eight separate files. These are then processed to generate the final key rates. The rates used here account for finite key effects.

We process the data in 20-min blocks, the first few seconds of data of each block being used to compute the optimal coincidence window to use for that block.

We use the Wegman-Carter authentication scheme [20] to estimate the amounts of authentication key needed and the average key rates generated by our real network over 18 h to simulate the running of these protocols.

All data from this publication are stored for at least 10 years on the University of Bristol's Research Data Storage Facility (RDSF). The processed data for the findings in this paper are available publicly from the RDSF. The set of raw data consisting of time-tag files is too large to host publicly and is available from the corresponding author on request.

4. Calculating the optimal flooding protocol(s)

The network can be represented by a simple graph where two vertices $\mathbf{v_i}$ and $\mathbf{v_j}$ are connected by an edge $(\mathbf{v_i}, \mathbf{v_j}) \in E$ if they share a key k_{ij}. Each edge is assigned a capacity that corresponds to the length of the key shared between the users per unit time (i.e., the key rate). In the case of our quantum network test bed, this corresponds to the length of key generated between two users in each 20-min block. This gives rise to a symmetric capacity matrix $\mathcal{C}$, where $\mathcal{C}_{ij}$ is the capacity of the edge connecting the vertices i and j.

Given two parties who wish to generate a key across the network, we consider one, Alice, acting as a source and the other, Bob, acting as a sink. As discussed in Sec. IV A in terms of the graph view, a given flooding protocol p_i corresponds to an assignment of an orientation to all of the intermediary edges. This orientation imposes a partial time ordering on the vertices in which a vertex i acts before another j if there is a positively orientated edge $(i \to j)$ connecting them. Orientation of the edges transforms the simple graph $\mathcal{N}$ into a directed graph $\mathcal{N}_{p_i}$. Each possible directed graph corresponds to a set of flooding protocols that are equivalent up to the ordering of the output edges when the key splitting is undertaken (as discussed in Sec. IV C).

Each flooding protocol has a maximum length of secure key that can be generated between Alice and Bob. This length corresponds to the maximum flow between the two vertices representing the parties in the directed graph. Since, in general, each directed graph has its own maximum flow, we refer to the directed graph maximizing the maximum flow as the optimum directed graph and the corresponding maximum flow as the optimum flow. A flooding protocol that achieves the optimum flow is called an optimal flooding protocol.

The maximum flow between two vertices in a directed graph is well known to be related to minimum cuts in the network by the max-flow min-cut theorem [42]. A cut is a bipartition of the vertices of the graph such that the source and sink lie in different sets. The cut set $\tilde{C}$ consists of all edges passing across the cut. Under a multipath routing strategy such as flooding the maximum flow $F_{\max}$ is given by

$$F_{\max}(\mathcal{N}_{p_i}) = \min_{\tilde{C}} \sum_{(i,j)\,\in\,\tilde{C}} \mathcal{C}_{ij}, \tag{A1}$$

which is the sum of the capacities of all the edges passing through the cut with the minimum value.

This can be implemented computationally using the Edmonds-Karp algorithm [43] to find the optimum flooding protocol(s). We first convert the undirected graph representing our network into a directed multigraph in which each edge in the original graph is converted into two edges with opposite orientations. We then perform a breadth-first search of the graph beginning from the source vertex and ending with the sink. By back-tracing, it is possible to find an augmenting path from the source to the sink with available capacity. We then send the maximum possible flow along this path and remove this from the capacity matrix to find the residual graph. This process continues until there are no further paths with available capacity and the flow

is maximized. An optimum flooding protocol (and corresponding optimum directed graph) can be found from the orientation of edges in the augmenting paths. More efficient algorithms such as Orlin's algorithm [44] can reduce the run time to $\mathcal{O}(VE)$, where V and E are the number of vertices and edges, respectively.

The case in which we consider partial trust is slightly more complex, as it may be possible that no optimum flooding protocol satisfies the trust requirements of the end users. However, a different flooding protocol, corresponding to a different directed graph, may satisfy the trust requirements. In this case, it is necessary to predefine the orientation of the edges, converting the graph into a directed graph $\mathcal{N}_{p_i}$. The Edmonds-Karp algorithm may then be used to find the maximum flow for this flooding protocol. The optimum protocol that obeys the end users' trust requirements may then be found by performing the procedure for all possible directed graphs and calculating the end-to-end trust values. We note that it may be more efficient to first calculate which combinations of MNOPs satisfy the trust requirements and then restrict to calculating the maximum flow for directed graphs that contain these paths.

Finally, we note that an optimum flooding protocol is in general nonunique. This nonuniqueness may arise in two ways: at the directed-graph level and at a "key-splitting" level. Different directed graphs with the same capacity matrix may achieve the same optimum flow, in which case there is more than one time ordering of the intermediary nodes that can perform an optimal flooding protocol. Additionally, even for the same directed graph and ordering of intermediaries, there may be more than one optimal protocol. This arises due to the different ways in which keys may be split, as discussed in Sec. IV C.

APPENDIX B

1. Derivation of authentication key scaling - Eq. 2

Equation 2 shows the minimum numbers of keys that must be shared while initiating a fully connected, authenticated network, in which there is the possibility of up to c_a users forming a collective adversary:

$$n_k = \frac{(c_a+2)(c_a+1)}{2} + (n-c_a-2)(c_a+2).$$

As discussed, to ensure the security of the SIAT protocol when sharing authentication keys, there must be c_a+1 disjoint paths between every pair of users that with to communicate (or a direct path). We assume that every pair of users wish to communicate.

For $n > c_a+1$, consider adding a new user. They must have received at least c_a+1 pre-shared keys in order to have authenticated communication with every other user; in fact, they need to receive that many exactly, if the underlying network is fully connected - that is, for each user with whom they share a key, they know that user has an authenticated channel with any other user with whom they may wish to communicate.

When considering the total number of pre-shared keys required, we first need to construct a fully connected network for the initial $n = c_a+1$ users, which requires $\frac{1}{2}(c_a+2)(c_a+1)$ pre-shared keys. Every remaining user needs to share keys with c_a+1 of these users, which means an additional $(n-c_a-1)(c_a+1)$ keys are required. A similar formula can be derived on need for any desired network topology.

2. Calculating trust duration when adding users into the network - Fig. 6

The WC authentication protocol [20], which is the most widely used, makes use of hashing functions which give each message a tag dependent on the message and a pre-shared key.

In the scheme, Alice sends Bob a message m out of a set M of possible messages, and appends the tag $f(m)$, in which f is a function from the set F which maps the set M to the set T of possible tags. In order for Bob to verify that this tag is genuine and the message was sent by Alice, Alice and Bob must have an initial shared key (k_{Auth}) which specifies which member of F is used for the tag. Therefore, the length of k_{Auth} depends on the size of F required.

Say that we would like to implement some protocol to produce a key of length a, and we would like the scheme to be insecure by probability c. Ref. [20] shows that the authentication tag must be in a set of size $|F|$, where $|F| = \frac{2}{c}$. Hence the tag length, b, should be $\log_2(\frac{2}{c})$ bits.

Ref. [20] then shows that the length of the original shared key should have length $s = 4(\log_2(\frac{2}{c}) + \log_2\log_2(d))\log_2(d)$ in which d is the length, in bits, of the message m. We assume that this is proportional to the final key length, a, i.e. this is $d = ga$ for some constant g. Experimentally, this is the amount of classical authenticated data that the two users need to exchange to be able to generate each bit of key. We are then interested in the number of bits of authentication key required per bit of final secret key, for insecurity c and final key length a, i.e. the proportion

$$4\left(\log_2\left(\frac{2}{c}\right) + \log_2\log_2(ga)\right)\log_2(ga)/a.$$

This is the proportion of key produced in each round of QKD that should be used as the authentication key for the following round. This is decided either arbitrarily, or dependent on the length of each round, a; conservatively let's say 10% of each round is reused for authentication.

We then solve $0.1 = 4\left(\log_2\left(\frac{2}{c}\right) + \log_2\log_2(ga)\right)\log_2(ga)/a$ for a to find the ideal number of bits to be produced in each round. For every event detected by a user,

the arrival time is measured by the time tagger and stored in a 64 bit binary format. Further the basis information (1 bit for each detected photon) is also recorded. Lastly, information is exchanged back and forth to perform error correction, status checks, etc. In our experiment this was at most 7 more bits of classical information sent between the users per photon detected. Given the experimentally measured key rates and detection rates, we conservatively estimate that for every every bit of key generated we had at most 10000 individual photon detection events. Thus we chose a value of g of 720 000. In this case, we find each round should be 50 563 bits, and so the key length should be $s = 4\left(\log_2\left(\frac{2}{c}\right) + \log_2\log_2(d)\right)\log_2(d) = 2179$ for our desired valued of c, 10^{-9}.

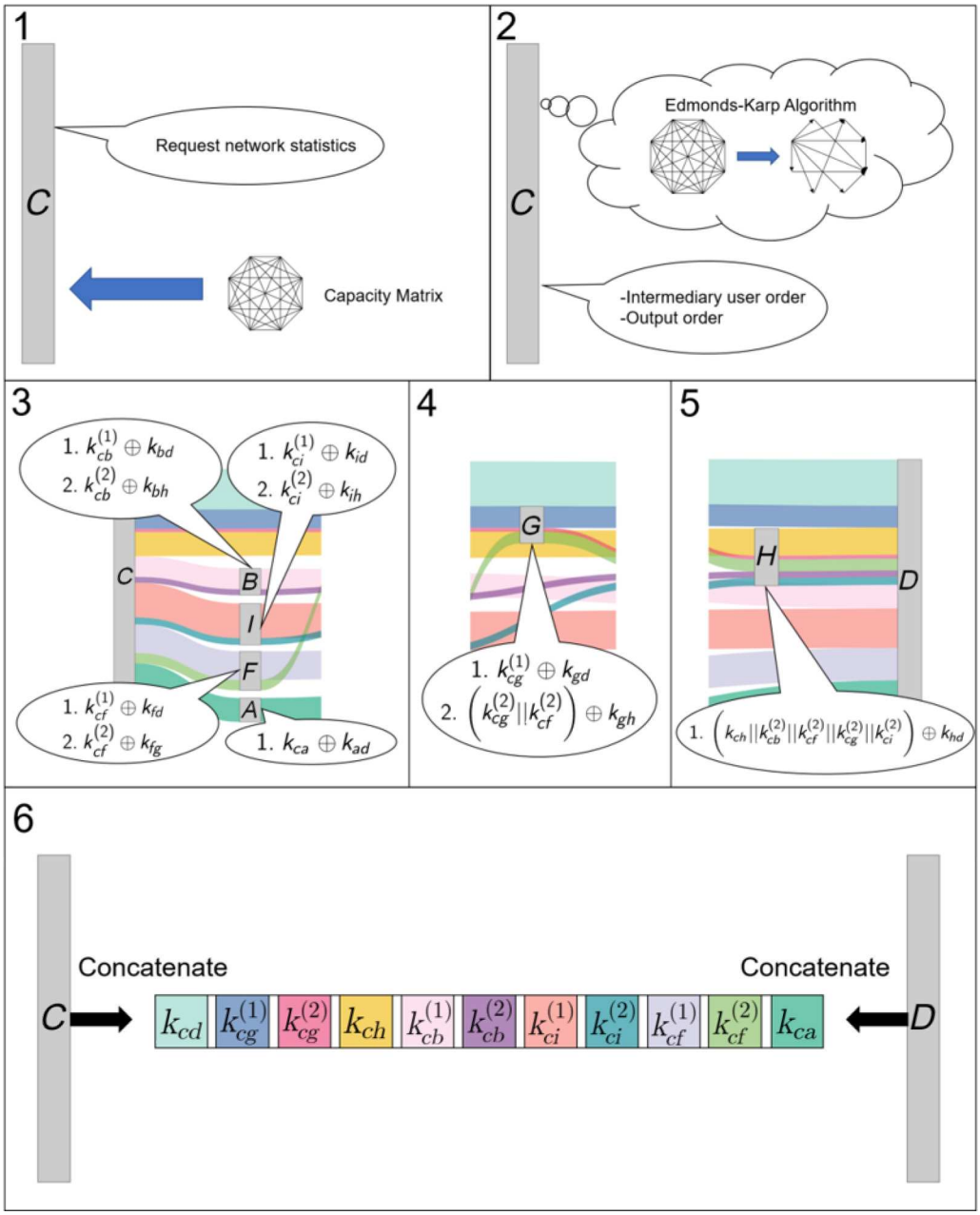

FIG. 12. *Step by step implementation of our flooding protocol on the quantum network test bed*: Panel 1 shows user C requesting network statistics to determine the capacity matrix of the available network resources. Panel 2 shows user C finding an optimal quantum secure flooding protocol using the Edmonds-Karp Algorithm and the capacity matrix. User C communicates to the intermediary nodes their ordering and the order in which to use their remaining keys to output their received keys. Panels 3, 4 and 5 show the announcements made by the intermediary nodes. Panel 6 shows the two end users privately concatenating their shared keys.

Fig. 6 shows the amount of time needed to carry out the SIAT protocol - that is, the amount of time needed (with previously experimentally generated key rates) for the trusted node to send the new user and the desired end user the 2179-bit tag, and then the amount of time needed for the new user and the end user to carry out a full 50 563-bit round of QKD.

For example, let's say the user I would like to make a connection with the user A, and already has a connection with the trusted node B. The mean key rate for connection AB in [3] is 45.94 bits per second, and 34.70 bps for BI. For the first part of the SIAT protocol, the trusted node B sends a 2179-bit tag to A and I simultaneously, which takes $2179/34.70 = 62.80$ s. A and I then carry out a full 50563-bit round of communication. As the mean key rate in communication between A and I is 28.52 bps, this takes 1773 s, and the total required time is 1836 s (≈ 30 minutes). This is represented by the first bar in Fig. 6.

We note that a similar process can be used to add users into any quantum network where the topology allows the end users to share initial authentication keys with at least one common node or set of nodes.

3. Announcements for optimal flooding with fully trusted nodes

Figure 8 shows the optimum end-to-end flooding protocol between users C and D. The announcements made by the intermediary nodes were suppressed in the figure to preserve clarity. Figure 12 shows the full set of announcements made by the intermediary nodes to pass knowledge of keys from C to D. These users concatenate these keys to form the final flooded key $k_{cd}^{(\mathrm{flood})}$ as shown in Fig. 8.

[1] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. L. Pereira, M. Razavi, J. Shamsul Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden, Advances in quantum cryptography, Adv. Opt. Photon. **12**, 1012 (2020).

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

[3] S. K. Joshi, D. Aktas, S. Wengerowsky, M. Lončarić, S. P. Neumann, B. Liu, T. Scheidl, G. Currás Lorenzo, Ž. Samec, and L. Kling, *et al.*, A trusted node-free eight-user metropolitan quantum communication network, Sci. Adv. **6**, eaba0959 (2020).

[4] X. Liu, X. Yao, R. Xue, H. Wang, H. Li, Z. Wang, L. You, X. Feng, F. Liu, and K. Cui, *et al.*, An entanglement-based quantum network based on symmetric dispersive optics quantum key distribution, APL Photonics **5**, 076104 (2020).

[5] Y. Shi, S. M. Thar, H. S. Poh, J. A. Grieve, C. Kurtsiefer, and A. Ling, in *CLEO: Applications and Technology* (Optical Society of America, 2020), p. ATu3S–5.

[6] N. B. Lingaraju, H.-H. Lu, S. Seshadri, D. E. Leaird, A. M. Weiner, and J. M. Lukens, Adaptive bandwidth management for entanglement distribution in quantum networks, (2020), arXiv preprint ArXiv:2010.10369.

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

[8] M. Sasaki, *et al.*, Field test of quantum key distribution in the Tokyo QKD network, Opt. Express **19**, 10387 (2011).

[9] D. Stucki, M. Legre, F. Buntschu, B. Clausen, N. Felber, N. Gisin, L. Henzen, P. Junod, G. Litzistorf, and P. Monbaron, *et al.*, Long-term performance of the SwissQuantum quantum key distribution network in a field environment, New J. Phys. **13**, 123001 (2011).

[10] F. X. Xu, W. Chen, S. Wang, Z. Q. Yin, Y. Zhang, Y. Liu, Z. Zhou, Y. B. Zhao, H. W. Li, and D. Liu, *et al.*, Field experiment on a robust hierarchical metropolitan quantum cryptography network, Chin. Sci. Bull. **54**, 2991 (2009).

[11] S. Wang, *et al.*, Field and long-term demonstration of a wide area quantum key distribution network, Opt. Express **22**, 21739 (2014).

[12] P. Toliver, R. J. Runser, T. E. Chapuran, J. L. Jackel, T. C. Banwell, M. S. Goodman, R. J. Hughes, C. G. Peterson, D. Derkacs, and J. E. Nordholt, *et al.*, Experimental investigation of quantum key distribution through transparent optical switch elements, IEEE Photonics Technol. Lett. **15**, 1669 (2003).

[13] T.-Y. Chen, J. Wang, H. Liang, W.-Y. Liu, Y. Liu, X. Jiang, Y. Wang, X. Wan, W.-Q. Cai, L. Ju, L.-K. Chen, L.-J. Wang, Y. Gao, K. Chen, C.-Z. Peng, Z.-B. Chen, and J.-W. Pan, Metropolitan all-pass and inter-city quantum communication network, Opt. Express **18**, 27217 (2010).

[14] C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer, and H. Yeh, in *Quantum Information and Computation III*, Vol. 5815, edited by Eric J. Donkor, Andrew R. Pirich, and Howard E. Brandt [International Society for Optics and Photonics (SPIE), 2005], p. 138.

[15] X.-Y. Chang, D.-L. Deng, X.-X. Yuan, P.-Y. Hou, Y.-Y. Huang, and L.-M. Duan, Experimental realization of an entanglement access network and secure multi-party computation, Sci. Rep. **6**, 29453 (2016).

[16] National Cyber Security Centre, *Quantum security technologies*, Tech. Rep. (National Cyber Security Centre, 2020).

[17] S. Pirandola, End-to-end capacities of a quantum communication network, Commun. Phys. **2**, 51 (2019).

[18] L. Salvail, M. Peev, E. Diamanti, R. Alléaume, N. Lütkenhaus, and T. Länger, Security of trusted repeater quantum key distribution networks, J. Comput. Ser. **18**, 61 (2010).

[19] C. Bennett and G. Brassard, Quantum cryptography: Public key distribution and coin tossing, Theor. Comput. Sci.—TCS **560**, 175 (1984).

[20] M. N. Wegman and J. L. Carter, New hash functions and their use in authentication and set equality, J. Comput. Syst. Sci. **22**, 265 (1981).

[21] C. Portmann, Key recycling in authentication, IEEE Trans. Inf. Theory **60**, 4383 (2014).

[22] W. Liu-Jun, Z. Kai-Yi, W. Jia-Yong, C. Jie, Y. Yong-Hua, T. Shi-Biao, Y. Di, T. Yan-Lin, and L. Zhen, *et al.*, Experimental authentication of quantum key distribution with post-quantum cryptography, (2020), arXiv preprint ArXiv:2009.04662.

[23] J. Cederlof and J.-A. Larsson, Security aspects of the authentication used in quantum cryptography, IEEE Trans. Inf. Theory **54**, 1735 (2008).

[24] A. S. Tanenbaum and D. Wetherall, *Computer Networks* (Pearson, 2011), 5th ed.

[25] S. L. Braunstein and S. Pirandola, Side-Channel-Free Quantum Key Distribution, Phys. Rev. Lett. **108**, 130502 (2012).

[26] H. K. Lo, M. Curty, and B. Qi, Measurement-Device-Independent Quantum Key Distribution, Phys. Rev. Lett. **108**, 130503 (2012).

[27] M. Minder, M. Pittaluga, G. L. Roberts, M. Lucamarini, J. F. Dynes, Z. L. Yuan, and A. J. Shields, Experimental quantum key distribution beyond the repeaterless secret key capacity, Nat. Photonics **13**, 334 (2019).

[28] 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 **557**, 400 (2018).

[29] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi, Fundamental limits of repeaterless quantum communications, Nat. Commun. **8**, 15043 (2017).

[30] S. Wengerowsky, S. K. Joshi, F. Steinlechner, J. R. Zichi, S. M. Dobrovolskiy, R. van der Molen, J. W. N. Los, V. Zwiller, M. A. M. Versteegh, and A. Mura, *et al.*, Entanglement distribution over a 96-km-long submarine optical fiber, Proc. National Acad. Sci. **116**, 6684 (2019).

[31] S. Wengerowsky, S. K. Joshi, F. Steinlechner, J. R. Zichi, B. Liu, T. Scheidl, S. M. Dobrovolskiy, R. van der Molen, J. W. N. Los, and V. Zwiller, *et al.*, Passively stable distribution of polarisation entanglement over 192 km of deployed optical fibre, npj Quantum Inf. **6**, 1 (2020).

[32] N. L. Piparo, M. Razavi, and W. J. Munro, Memory-assisted quantum key distribution with a single nitrogen-vacancy center, Phys. Rev. A **96**, 052313 (2017).

[33] F. Furrer and W. J. Munro, Repeaters for continuous-variable quantum communication, Phys. Rev. A **98**, 032335 (2018).

[34] Z. Huang, S. K. Joshi, D. Aktas, C. Lupo, A. O. Quintavalle, N. Venkatachalam, S. Wengerowsky, M. Lončarić, S. P. Neumann, and B. Liu, *et al.*, Experimental implementation of secure anonymous protocols on an eight-user quantum network, (2020), arXiv preprint ArXiv:2011.09480.

[35] V. Zapatero and M. Curty, Secure quantum key distribution with a subset of malicious devices, npj Quantum Inf. **7**, 1 (2021).

[36] L. Mazzarella, C. Lowe, D. Lowndes, S. K. Joshi, S. Greenland, D. McNeil, C. Mercury, M. Macdonald, J. Rarity, and D. K. L. Oi, Quarc: Quantum research cubesat—a constellation for quantum communication, Cryptography **4**, 7 (2020).

[37] S. P. Neumann, S. K. Joshi, M. Fink, T. Scheidl, R. Blach, C. Scharlemann, S. Abouagaga, D. Bambery, E. Kerstel, and M. Barthelemy, *et al.*, Q^3 Sat: Quantum communications uplink to a 3U CubeSat—feasibility & design, EPJ Quantum Technol. **5**, 4 (2018).

[38] E. Kerstel, A. Gardelein, M. Barthelemy, M. Fink, S. K. Joshi, R. Ursin, and The CSUG Team, *et al.*, Nanobob: A CubeSat mission concept for quantum communication experiments in an uplink configuration, EPJ Quantum Technol. **5**, 6 (2018).

[39] C. Harney and S. Pirandola, Optimal Performance of Global Quantum Networks, (2021), arXiv preprint ArXiv:2104.10701.

[40] Y. A. Chen, *et al.*, An integrated space-to-ground quantum communication network over 4600 kilometres, Nature **589**, 214 (2021).

[41] C. H. Bennett, G. Brassard, and N. D. Mermin, Quantum Cryptography without Bell's Theorem, Phys. Rev. Lett. **68**, 557 (1992).

[42] L. R. Ford and D. R. Fulkerson, Maximal flow through a network, Can. J. Math. **8**, 399 (1956).

[43] J. Edmonds and R. M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, J. ACM (JACM) **19**, 248 (1972).

[44] J. B. Orlin, Max flows in $O(nm)$ time, or better, Proc. Annu. ACM Symp. Theory Comput. 765 (2013).