Anonymous CKA with linear cluster states

J. de Jong, F. Hahn, J. Eisert. N. Walk, A. Pappa

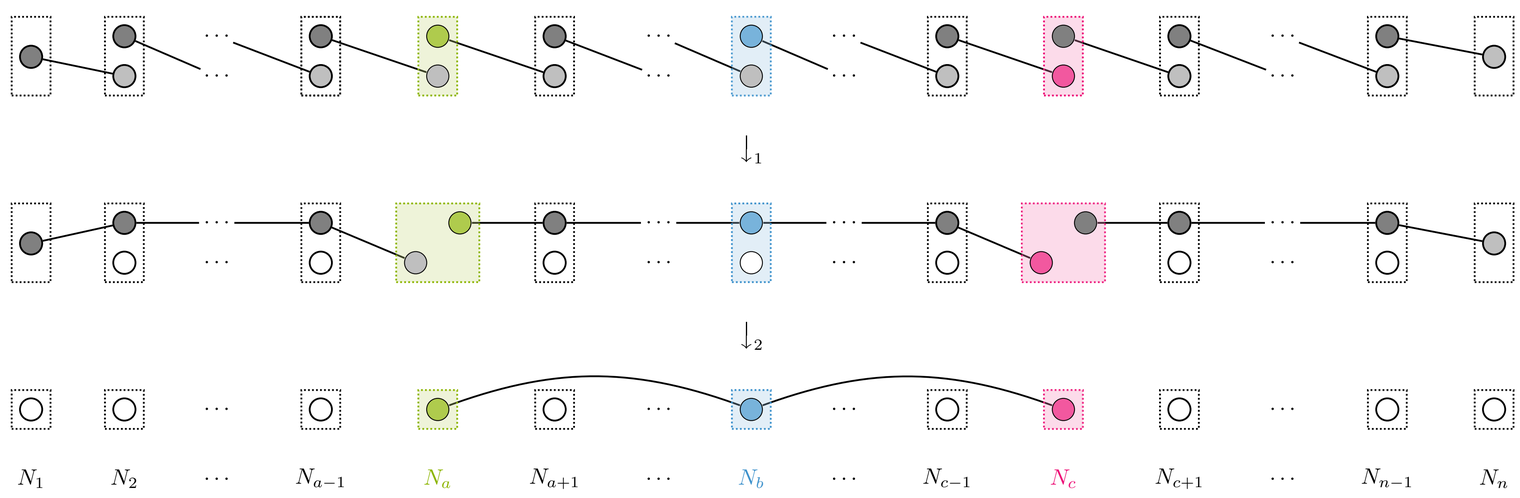
Abstract

ACKA aims to obtain a secret and secure key between a subset of parties in a network, while keeping their identity hidden. In comparison to earlier work where full-network multipartite entanglement is necessary, here we assume a nearest-neighbour line architecture where only bi-partite quantum links exist, and provide a protocol for 3 anonymous parties to create key; we provide full finite-key analysis and simulations for various noise levels.

Anonymous Conference Key Agreement

The goal of *conference key agreement*[1] (CKA) is to generate a secret key between multiple parties (*the participants*) within a larger network. Additionally, this can be performed as *anonymous* CKA, where the identities of the participants keeps hidden from the rest of the network (the *non-participants*). ACKA has been proposed with both bipartite[2] and multipartite entangled quantum resources[2,3] distributed over a fully connected network. Here, we follow a more realistic approach where we consider a network of nodes $\{N_i\}$ in a nearest-neighbour linear configuration, where as an initial resource every node shares an EPR pair (i. e. $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$) with their neighbours. After running the protocol, three parties sitting anywhere in the linear network – *Alice* (N_a), *Bob* (N_b) and *Charlie* (N_c) – obtain a secret and secure key while not divulging their identity. During the protocol, three linear cluster states are created from the initial resources; subsequently, a GHZ $= \frac{1}{\sqrt{3}}(|000\rangle + |111\rangle)$ state is extracted between the three participants. This state is either verified by them, or is used for key generation. During classical post-processing both error correction is performed to obtain a perfectly correlated and secret key. Full finite-key analysis is given, which is largely based on previous tools[4] but includes adaptations to keep the identity of the participants hidden. We provide the finite key rate as a function of the total number of network uses and provide simulations for a multitude of noise rates within the network.

Graphical overview



Graphical representation of the protocol, which exist of two steps plus post-processing. During **the first step**, almost all parties merge their qubits together to create three *linear cluster* states. **The second step** involves all non-participants measure their qubits under an identity-agnostic basis pattern, so that the participants obtain a state LC-equivalent to the GHZ state. **Subsequently**, the participants correct their state and perform either `Verification` of the state or `KeyGeneration` to obtain raw key. **Finally**, the participants perform anonymity-sustaining post-processing, including error-correction and privacy-amplification.

1. State preparation

In the **first subprotocol**, all nodes $\{N_i\}$ except for N_a and N_c perform *Bell state projections* to create three linear cluster states from the initial Bell pairs.

All previously mentioned nodes N_i perform:

- Receive *correction* bit o_{i-1} and apply Z on top qubit conditionally
- Perform CZ between two qubits, measure top qubit in σ_x basis and record outcome o_i
- Send o_i to next node

The other nodes perform steps to hide their identity.

2. GHZ extraction

In the **second subprotocol**, the non-participants $\{N_i\} \setminus \{N_a, N_b, N_c\}$ measure their leftover qubits in an alternating σ_x-σ_y pattern. When all measurements are finished, everyone announces their outcome; the participants announce random bits to hide their identity. The resulting state for the participants is now LC-equivalent to the GHZ state.

3. Measurements and post-processing

In the **third subprotocol**, the participants use some pre-shared key to coordinate their measurements in either the σ_z-basis for `KeyGeneration` or the σ_x-basis for `Verification`; first, they rotate their qubits under the necessary corrections so that they obtain the proper state. The previous two protocols are repeated L times, where $k << L$ rounds are for verification and $L - k$ for keygeneration. If the fraction of faulty verification rounds doesn't exceed a pre-determined Q_{tol}, the participants perform error-correction and privacy amplification. To keep their identity hidden, all communication is OTP-encrypted using a pre-shared key.

References

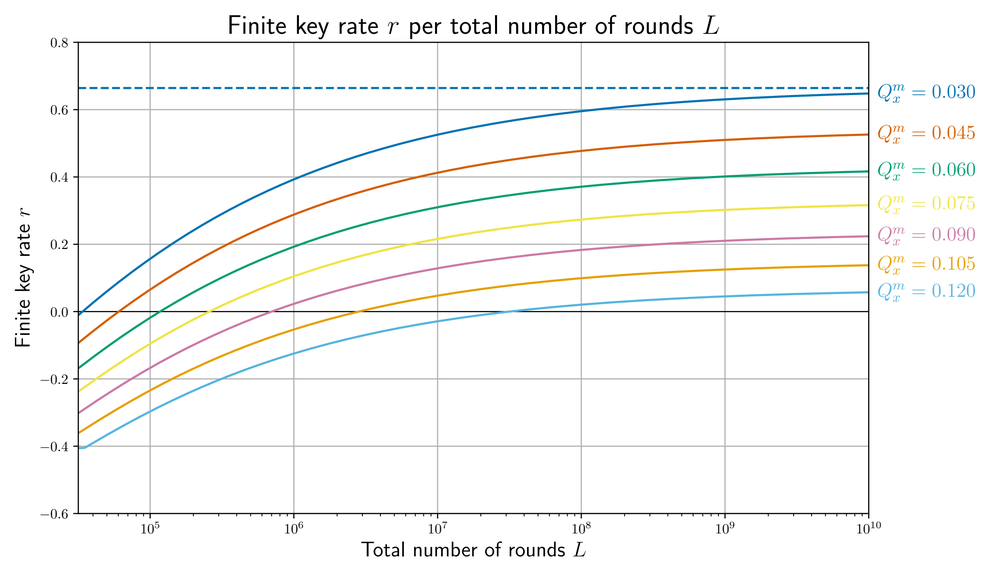
[1] G. Murta et al., "Quantum conference key agreement: A review", Advanced Quantum Technologies 3.11 (2020)

[2] F. Grasselli et al., "Robust Anonymous Conference Key Agreement enhanced by Multipartite Entanglement", ArXiv preprint (2021)

[3] F. Hahn, J. de Jong and A. Pappa, "Anonymous Quantum Conference Key Agreement", PRX Quantum (2020).

[4] F. Grasselli et al. "Finite-key effects in multipartite quantum key distribution protocols", New Journal of Physics, (2018)

Finite key rate



$$r = (1-p)\left[1 - h_2\left(Q_{\text{tol}} + \mu\left(\frac{\varepsilon_{\text{s}} - \varepsilon}{2}\right)\right) - h_2(Q_z)\right] - h_2(p) + \frac{1}{L}\left(\log_2\left(\varepsilon^2\varepsilon_c\right) - 2\right),$$

Link to pre-print