Interpretation of Symbols in shor-preskill.scr

In this document, interpretation of symbols in BB84 and the EDP-based protocol, which are discussed by Shor and Preskill [1], is introduced. BB84 and the EDP-based protocol are formalized as qCCS processes on the basis of our previous work [2]. Readers can find the script `scripts/shor-preskill.scr` in the package.

In the script, Alice's, Bob's and Eve's quantum variables are appended with `_A`, `_B` and `_E` respectively for readability. The exception is that `EVE_2[r_B]` is initially the state of Eve's variable but she will be able to send it to Bob through `c2?r_B` because `c2` is public. This intuitively means that arbitrary quantum state that an adversary has prepared can be sent to Bob through public channel.

The EDP-based protocol employs CSS quantum error-correcting code (QECC), which is constructed from two classical linear codes C_1, C_2. CSS QECC can be parameterized with $x \in C_1$ and $y \in C_2$. We write $CSS^{x,y}(C_1, C_2)$ for CSS code parameterized x and y that employ codes C_1 and C_2.

1 Interpretation of Symbols in the EDP-based Protocol

Density Operators

- Alice first prepares EPR pairs. Let quantum variables q and r be of the length `n`, where `n` interpreted as an arbitrary natural number n. `EPR[`q, r`]` is interpreted as EPR pairs $(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|)_{q,r}^{\otimes n}$.
- `RND[`q`]` is interpreted as $(|0\rangle\langle 0| + |1\rangle\langle 1|)_q^{\otimes n}$.
- `Z[`q`]` is interpreted as $|0\rangle\langle 0|_q^{\otimes n}$.
- `EVE`, `EVE1` and `EVE2` are arbitrarily interpreted. They express quantum states that are prepared by the adversary. `EVE` is one for a quantum variable with length `m`, where `m` is interpreted as an arbitrary natural number m. `EVE1` and `EVE2` are ones for quantum variables with length `n`.

Superoperators

- `hadamards[`q, r, s`]` randomly performs Hadamard transformation to qubit-string q, r according to a bitstring s which serves as a seed of randomness.
- `shuffle[`q, r, s`]` randomly shuffles the position of qubit-string q, r according to the randomness s.
- `copy2n[`q, r`]` copies the value of q with length `2n` to r, where q is supposed to be assigned a classical value. `copyN[`q, r`]` and `copy1[`q, r`]` are for quantum variables with length `N` and `1`.
- `measure[`q`]` is the projective measurement of q.
- `abort_alice[`q, r, s`]` compares two bitstrings q and r, and sets the value 0 to a bit s if the difference between q and r is lower than the threshold, else sets the value 1 to s.
- `css_projection[`q, r, s`]` converts the halves of EPR pairs q to a random $CSS^{x,y}(C_1, C_2)$ codeword, where parameters x, y are also determined randomly. The value of x and y are stored in r and s.

- `css_decode[`q, r, s`]` decodes q as $CSS^{x,y}(C_1, C_2)$ codeword when the value of r and s are x and y.
- `unshuffle[`q, r, s`]` is the inverse of `shuffle[`q, r, s`]`.
- `css_syndrome[`q, r, s, u, v`]` calculates the error syndrome of q as a codeword of $CSS^{x,y}(C_1, C_2)$ when r and s have the value x and y, and stores the syndrome in u and v.
- `css_correct[`q, r, s`]` is error correction with the syndorme stored in r, s.

1 Interpretation of Symbols in BB84

BB84 employs classical codes C_1 and C_2 which correspond to $CSS^{x,y}(C_1, C_2)$ in the EDP-based protocol.

Density Operators

- Alice first prepares two same random bitstrings. This initial state is represented by `PROB[`q, r`]` with q for Alice and r for Bob, which is interpreted as $(|00\rangle\langle 00| + |11\rangle\langle 11|)_{q,r}^{\otimes n}$.
- `RC1[`q`]` is interpreted as $\sum_{u \in \mathrm{C1}} |u\rangle\langle u|$.
- `RC2[`q`]` is interpreted as $\sum_{v \in \mathrm{C2}} |v\rangle\langle v|$.

Superoperators

- `syndrome[`q, r`]` calculates the error syndrome of q using as a codeword in C_1 and store the syndrome to r.
- `correct[`q, r`]` corrects errors of q with the syndrome r.
- `key[`q`]` calculates with respect to C_2 the coset of the value that is an element of C_1 and stored in q.

References

[1] P. W. Shor and J. Preskill. Simple proof of security of the bb84 quantum key distribution protocol. *Phys. Rev. Lett.*, 85(2):441–444, Jul 2000.

[2] T. Kubota, Y. Kakutani, G. Kato, Y. Kawano, and H. Sakurada. Application of a process calculus to security proofs of quantum protocols. *Proceedings of WORLDCOMP/FCS2012*, Jul 2012.