Pan-European Quantum Internet Hackathon

Quantum cheque protocol

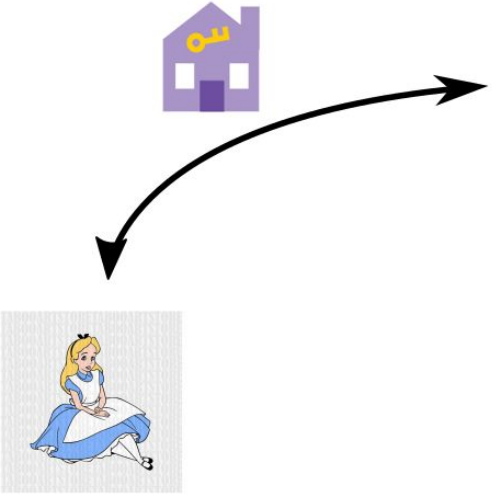
Team Quantum Winter:
Nicolas GAUDE, Michel NOWAK, Rhea PAREKH, ChinTe LIAO, Hop DINH, Amar MOUFFOK

Outline

1. Introduction
2. GENERATE
3. SIGN
4. VERIFY
5. Conclusion

GENERATE

Alice goes to the bank (QKD)



bob

alice

charlie

GENERATE

bob

alice

charlie

GENERATE

bob

alice

bob gives the checkbook to alice

charlie

SIGN

bob

alice

charlie

alice signs the cheque

bob

alice

charlie

alice sends the cheque to charlie

VERIFY

bob

alice

charlie

charlie submits the cheque to the bank

VERIFY

bob

alice

charlie

the bank verifies and cashes out the check

GENERATE

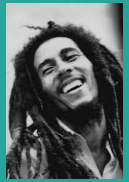
- QKD BB84 protocol
- GHZ states preparation

Protocol

A B C

Use BB84 to have a shared key between the Alice and Bank

BB84 protocol



- generate random bit string and random basis

qubits to Bob →

-measure qubits in Bob's basis

← Bob's basis to Alice

- compare Bob's basis to her basis

- form a key based on her basis and matched basis.

The index of the match →

- filt out measurement results based on matched basis, and form a key

Protocol

A B C

1. Bank generates n GHZ state

GHZ states preparation

$$\left|\phi^{(i)}\right\rangle_{\mathrm{GHZ}} = \frac{1}{\sqrt{2}}\left(\left|0^{(i)}\right\rangle_{A_1}\left|0^{(i)}\right\rangle_{A_2}\left|0^{(i)}\right\rangle_{B} + \left|1^{(i)}\right\rangle_{A_1}\left|1^{(i)}\right\rangle_{A_2}\left|1^{(i)}\right\rangle_{B}\right)$$

Protocol

B

2. Bank sends group A and C to Alice

A C

SIGN

- The one way function
- Bell states measurement

Protocol

B

3. Using keys like shared key, amount of money, database_id and randomised salt parameter, Alice produces a unique key and generates a state from the quantum one way function using this unique key

OWF A C

The one way function

$$\left|\psi^{(i)}\right\rangle = f(k||\mathrm{id}||r||M||i)$$

A quantum one way function is defined as,

$$\Psi : k \times |0\rangle^{\otimes n} \rightarrow |\psi_k\rangle \,,$$

where $k \in \{0, 1\}^*$ and $|\psi_k\rangle$ is a n-qubit quantum state, such that,

– Ψ is easy to compute, i.e. there exists a polynomial-time algorithm that can evaluate $\Psi(k, |0\rangle^{\otimes n})$ and outputs $|\psi_k\rangle$,
– Ψ is hard to invert, i.e. given $|\psi_k\rangle$, it is difficult to compute k

The one way function

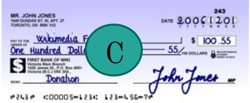
```
def one_way_function(conn, BB84_key, db_id, r, M):
    owf_state = qubit(conn)
    owf_key = bin(BB84_key)[2:] + bin(db_id)[2:] + bin(r)[2:] + bin(M)[2:]
    owf_key = int(abs(hash(str(owf_key))))
    # p1 , p2, p3 are prime numbers , so coprimes
    # thus rotation X(key%p1) and Y(key%p2) and Z(key%p3) are independant
    p1 = 33179
    p2 = 32537
    p3 = 31259
    owf_state.rot_X(owf_key%p1%256)
    owf_state.rot_Y(owf_key%p2%256)
    owf_state.rot_Z(owf_key%p3%256)
    return owf_state
```

Protocol

B

4. Alice performs a Bell state measurement on the states OWF and A, thus collapsing the states and sending the information to the entangled B and C states. C state is now the cheque.

OWF A



C

Bell Measurement

$$\left|\phi^{(i)}\right\rangle = \left|\psi^{(i)}\right\rangle \otimes |\phi\rangle_{\mathrm{GHZ}}$$
$$= \frac{1}{2}\left\{\left|\Phi^{+}\right\rangle_{A_1} (\alpha_i |00\rangle_{A_2B} + \beta_i |11\rangle_{A_2B})\right.$$
$$+ \left|\Phi^{-}\right\rangle_{A_1} (\alpha_i |00\rangle_{A_2B} - \beta_i |11\rangle_{A_2B})$$
$$+ \left|\Psi^{+}\right\rangle_{A_1} (\beta_i |00\rangle_{A_2B} + \alpha_i |11\rangle_{A_2B})$$
$$\left. + \left|\Psi^{-}\right\rangle_{A_1} (\beta_i |00\rangle_{A_2B} - \alpha_i |11\rangle_{A_2B})\right\}$$

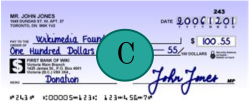
Protocol

B

5. Alice sends the cheque to Charlie

Protocol

B



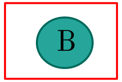
C

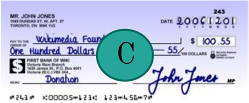
6. Charlie submits cheque to the bank

VERIFY

- Local corrections and measurement of Bank’s qubit group
- The swap test

Protocol

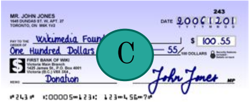




7. Bank performs local corrections on B group and measures it with Hadamard bases to collapse the state and reflect the information on the cheque

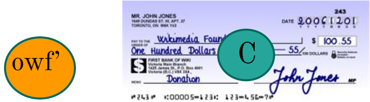
Protocol

owf’



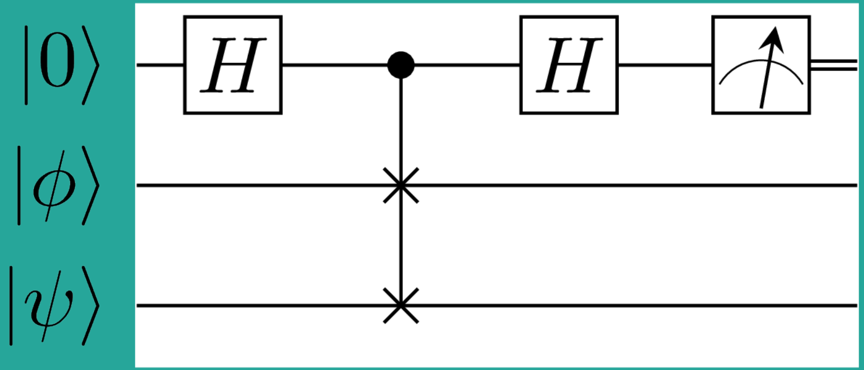
8. Using supplementary information like shared key, amount of money, database id, the Bank generates the unique id back and recreates the owf’ state using this unique id

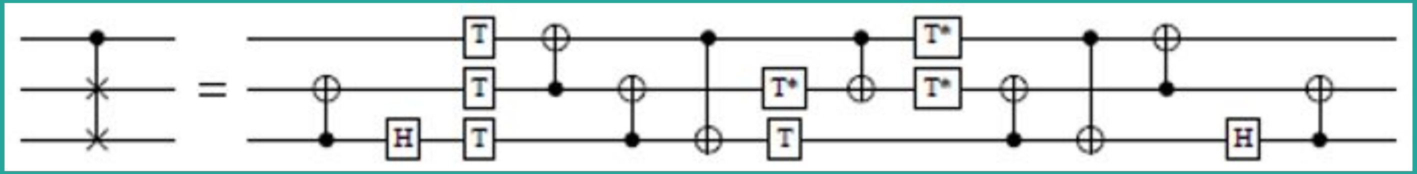
Protocol



9. Finally a SWAP test is performed between owf’ and C. If the SWAP test passes, the cheque is accepted. Otherwise it the cheque is denied and the protocol is aborted.

Swap test

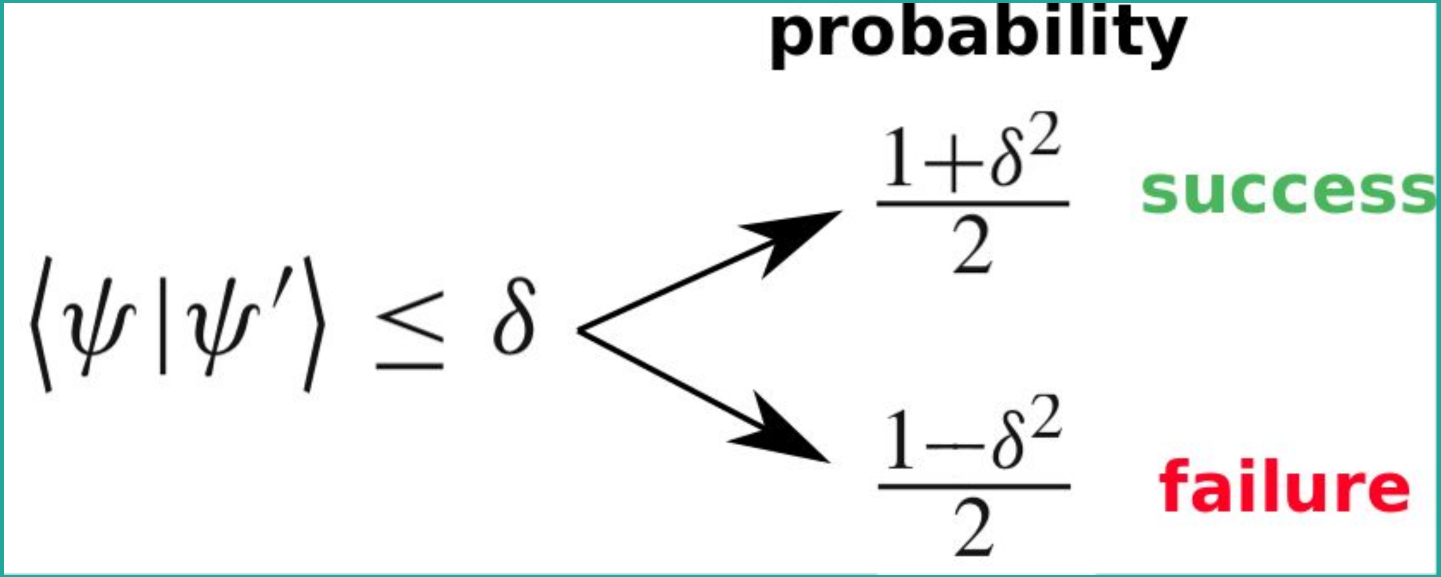




Swap test

```python
from one_way_function import one_way_function
from cqc.pythonLib import CQCConnection, qubit
import random
import tqdm


def T(q):
    # T = RZ(pi/4) * e(i*pi/8)
    q.rot_Z(256//8)
    return


def invT(q):
    # T* == RZ(-pi/4) * e(i*pi/8)
    q.rot_Z(256 - 256//8)
    return


def CSWAP(q0, q1, q2):
    # fredkin implementation from :
    # https://www.mathstat.dal.ca/~selinger/quipper/doc/QuipperLib-GateDecompositions.html
    q2.cnot(q1)
    q2.H()
    T(q0)
    T(q1)
    T(q2)
    q1.cnot(q0)
    q2.cnot(q1)
    q0.cnot(q2)
    invT(q1)
    T(q2)
    q0.cnot(q1)
    invT(q0)
    invT(q1)
    q2.cnot(q1)
    q0.cnot(q2)
    q1.cnot(q0)
    q2.H()
    q2.cnot(q1)
    return

def swap_test(conn, q1, q2):
    # swap_test implementation from :
    # https://en.wikipedia.org/wiki/Swap_test

    # q0 = qubit(conn)
    # q1.cnot(q0)
    # q2.cnot(q0)
    # m = q0.measure()
    # q1.measure()
    # q2.measure()
    # return m

    q0 = qubit(conn)
    q0.H()
    CSWAP(q0, q1, q2)
    q0.H()
    m = q0.measure()

    # collaspse everything after swap_test to avoid :
    # cqc.pythonLib.CQCNoQubitError: No more qubits available
    q1.measure()
    q2.measure()

    return m

```

Swap test


probability
$\langle \psi | \psi' \rangle \leq \delta$
$\frac{1+\delta^2}{2}$
success
$\frac{1-\delta^2}{2}$
failure

DEMO !


QIH