Quantum consensus networks

Peter P. Rohde[1,*]
[1] *BTQ Technologies*

I. QUANTUM CONSENSUS NETWORKS

Quantum consensus networks (QCNs) have quantum-enabled nodes, whose goal it is to form consensus on the generation of certifiable quantum randomness, an important resource in cryptography and numerous other applications.

As quantum hardware is costly compared to classical hardware it is expected that few networks will be quantum-enabled. However, they may exploit the quantum randomness provided by dedicated QCNs acting as *quantum random oracles* (Sec. I.C) to inject entropy into their own global keys using *entropy addition* (Sec. I.B), effectively enabling classical DCNs to achieve quantum random consensus assignment.

We consider two approaches for quantum random number generation (QRNG):

- Quantum key distribution (QKD): requires quantum communication but no quantum computation (Sec. I.D).
- Interactive proofs of quantumness: require quantum computation but no quantum communication (Sec. I.E).

As these are two-party protocols, every instance may be associated with a graph edge between the respective nodes. Random numbers associated with edges may be spoofed if both nodes collude, bypassing the need for expensive quantum resources. However, so long as at least one contributing QRN is genuine, combining them under bit-wise XOR yields a collective QRN source.

Independent of the underlying two-party QRNG protocol, QCNs operate as follows:

1. All nodes execute two-party QRNG with all other nodes, a total of $n(n-1)$ QRNG rounds.
2. All n nodes commit their versions of their QRNG outcomes with all other $n-1$ nodes.
3. A corresponding *compliance graph* is implied by the committed data. This may be realised by any entity observing the published data.
4. A graph reduction algorithm eliminates graph edges associated with inconsistent QRNG outcomes, followed by eliminating nodes not connected by a majority of edges.
5. The resultant graph is a unique fully-connected subgraph representing the unanimous-majority outcome (Sec. I.D.1).

A. Entropy sources: quantum vs. classical

* Entropy rate

$$H(\mathcal{X}) = \lim_{n\to\infty} H(\mathcal{X}_n|\mathcal{X}_{n-1},\ldots,\mathcal{X}_1), \tag{1.1}$$

where $H(X|Y)$ is the conditional entropy of X given knowledge of Y.

TO DO:

* What is quantum random vs classical random.
* Correlations over repeats?

For iid (quantum)

$$H(\mathcal{X}) = H(\mathcal{X}_n) \tag{1.2}$$

B. Entropy addition

The bit-wise XOR operator is a strictly non-entropy-decreasing function. For binary random variables,

$$\mathcal{X}_i \to \{0,1\}, \tag{1.3}$$

with probability distributions,

$$\mathcal{X}_i \sim \text{Ber}(p_i) : \; p_i = p_i(0) = 1 - p_i(1), \tag{1.4}$$

the individual binary Shannon entropies are given by,

$$H_2(\mathcal{X}_i) = -p_i \log_2 p_i - (1-p_i)\log_2(1-p_i), \tag{1.5}$$

where,

$$0 \leq H_2(\cdot) \leq 1, \tag{1.6}$$

the entropy of the random variable given by their bit-wise XOR,

$$\mathcal{X} = \bigoplus_i \mathcal{X}_i, \tag{1.7}$$

is lower-bounded by the maximum entropy of the contributing random variables,

$$\max_i(\{H_2(\mathcal{X}_i)\}) \leq H_2(\mathcal{X}) \leq 1. \tag{1.8}$$

Hence, a random source derived from multiple sources via entropy addition is at least as random as any of them. Consequently, if any single contributing source is a genuine QRNG, so too will be their the combined source.

Note that hash functions do not exhibit the entropy addition property of the bitwise XOR operation and cannot be employed in this context.

* peter@peterrohde.org; https://www.peterrohde.org

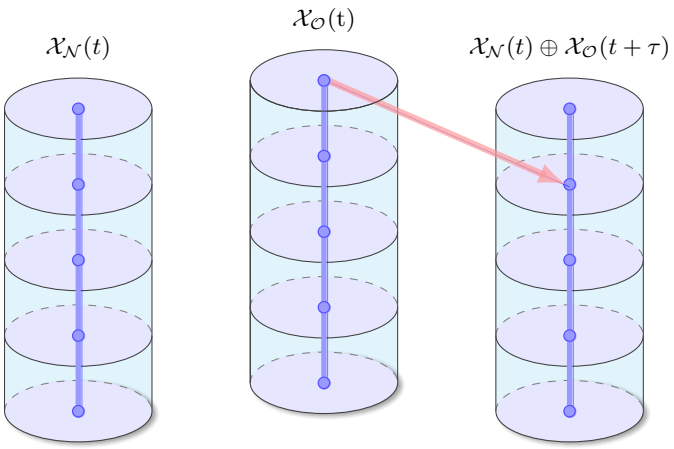

Figure 1: Quantum random oracles. A classical DCN establishing hash-based secure random source $\mathcal{X}_{\mathcal{N}}(t)$, where t denotes time, may add entropy from a QCN acting as oracle for quantum random source $\mathcal{X}_{\mathcal{O}}(t)$. To maintain security $\mathcal{X}_{\mathcal{N}}$ must be established in advance of $\mathcal{X}_{\mathcal{O}}$, achieved by adding entropy from a QRN outcome at pre-agreed future time $t+\tau$, $\tilde{\mathcal{X}}_{\mathcal{N}}(t) = \mathcal{X}_{\mathcal{N}}(t) \oplus \mathcal{X}_{\mathcal{O}}(t+\tau)$.

C. Quantum random oracles

Let $\mathcal{O}(t)$ denote a dynamic set of contributing random oracles at time t. We define a collective random bitstream,

$$\mathcal{X}_{\mathcal{O}}(t) = \bigoplus_{i \in \mathcal{O}(t)} \mathcal{X}_i(t), \tag{1.9}$$

whose combined entropy is bounded by,

$$\max_{i \in \mathcal{O}}(\{H_2(\mathcal{X}_i)\}) \leq H_2(\mathcal{X}_{\mathcal{O}}) \leq 1. \tag{1.10}$$

A classical DCN may observe a QRNG oracle and add its entropy $\mathcal{X}_{\mathcal{O}}$ to its own global key. For this to be secure it is required that the DCN's own global key, $\mathcal{X}_{\mathcal{N}}$, be committed prior to the availability of the external entropy source,

$$\tilde{\mathcal{X}}_{\mathcal{N}}(t+\tau) = \mathcal{X}_{\mathcal{N}}(t) \oplus \mathcal{X}_{\mathcal{O}}(t+\tau), \tag{1.11}$$

where $\tilde{\mathcal{X}}_{\mathcal{N}}$ is the network's oracle-modulated global key, and $\tau > 0$ is a pre-agreed future point in time, subsequent to commitment of the networks initially established global key, $\mathcal{X}_{\mathcal{N}}$.

D. Quantum key distribution (QKD)

Quantum key distribution (QKD) (**??**) enables the secure establishment of shared randomness between two parties with information theoretic security. While ordinarily utilised for secret key exchange, here we exploit not the secrecy of shared randomness but its inability to be spoofed under honest execution of the protocol.

Assuming the existence of a quantum internet (**?**) capable of arbitrary point-to-point entanglement routing, all node-pairs (i,j) have access to an indefinite supply of maximally-entangled Bell pairs,

$$|\Psi\rangle_{i,j} = \frac{1}{\sqrt{2}}(|0\rangle_i |0\rangle_j + |1\rangle_i |1\rangle_j), \tag{1.12}$$

requiring full $O(n^2)$ quantum communications connectivity.

For the nth copy of $|\Psi\rangle_{i,j}$ both nodes independently and privately choose measurement bases,

$$b_i(n), b_j(n) \in \{0,1\}, \tag{1.13}$$

where $b=0$ denotes the Pauli-Z basis and $b=1$ the Pauli-X basis, and record their associated measurement outcomes,

$$m_i(n), m_j(n) \in \{0,1\}. \tag{1.14}$$

The subset of measurement outcomes where both parties operate in the same basis defines a shared random bit-string,

$$s_{i,j} = \{m(n) \,|\, b_i(n) = b_j(n)\}_n. \tag{1.15}$$

These post-selected bit-strings correspond identically to those provided by the E91 (**?**) QKD protocol. The BB84 protocol (**?**) can be similarly employed with the interpretational difference that for one party b and m denote encoding, for the other measurement.

Physical and implementation errors reduce the otherwise perfect measurement correlations between nodes measuring in the same basis resulting in inconsistent shared strings. However, privacy amplification (**?**) may be used to reduce an imperfect random bit-string to a shorter one with higher entropy using classical post-processing.

1. Consensus protocol

Associating QKD bit-strings $s_{i,j}$ with graph edges we assume a certification function,

$$f_{\mathrm{QKD}}(s_{i,j}) \to \{0,1\}, \tag{1.16}$$

which evaluates `true` if $s_{i,j}$ passes a certification test for randomness. We define a QKD compliance graph with edge-inclusion based on the validity of the respective QKD bit-strings,

$$\mathcal{G}^{(\mathrm{comp})}_{\mathrm{QKD}} : e_{i,j} = f_{\mathrm{QKD}}(s_{i,j}). \tag{1.17}$$

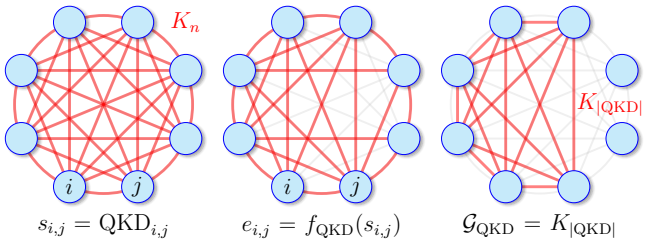

Figure 2: QKD proof-chain for shared quantum randomness. Amongst n nodes with full $O(n^2)$ quantum communications connectivity given by the complete graph K_n, every node executes a QKD protocol with every other node, associating a shared random bit-string $s_{i,j}$ (received by i from j) with every edge. Bit-strings passing a QRN certification function $f_{\mathrm{QKD}}(s_{i,j})$ define the edges of a subgraph, where certification acts as an implied vote of honesty. Maintaining only vertices connected to a majority of other nodes followed by eliminating those not connected to every other, we obtain a complete subgraph $\mathcal{G}_{\mathrm{QKD}}$ reflecting the unanimous majority.

Letting the QKD compliance of nodes be,

$$\mathcal{G}^{(\mathrm{comp})}_{\mathrm{QKD}} : v_i = \textsc{Majority}\left(\bigcup_{j\neq i} e_{i,j}\right), \tag{1.18}$$

the reduced graph now only contains nodes whose associated QKD bit-strings $s_{i,j}$ are majority valid.

Additionally requiring unanimity demands finding a fully-connected subgraph, or clique[1], achieved by eliminating all vertices in $\mathcal{G}^{(\mathrm{comp})}_{\mathrm{QKD}}$ not connected by an edge to every other node,

$$\mathcal{G}_{\mathrm{QKD}} : v_i = \begin{cases} 1 & \text{if } |u_i \in \mathcal{G}^{(\mathrm{comp})}_{\mathrm{QKD}}| = |\mathcal{G}^{(\mathrm{comp})}_{\mathrm{QKD}}| - 1 \\ 0 & \text{otherwise} \end{cases}. \tag{1.19}$$

The fully connected $\mathcal{G}_{\mathrm{QKD}} = K_{|\mathcal{G}_{\mathrm{QKD}}|}$ subgraph now represents the accepted subset of QKD-compliant nodes under consensus. The associated collectively established shared random bit-string is defined as,

$$s(\mathcal{G}_{\mathrm{QKD}}) = \bigoplus_{v_i, v_j \in \mathcal{G}_{\mathrm{QKD}}} s_{i,j}. \tag{1.20}$$

Although nodes could commit post-processed QKD strings obtained following post-selection and privacy amplification, this requires interaction between respective nodes. In the interests of maintaining a broadcast-only communications interface nodes may simply commit their raw unprocessed strings (b and m) from which the associated QRNs s are implied under a network-agreed post-processing function $f_{\mathrm{PP}}(\vec{b}, \vec{m}) \to \vec{s}$.

[1] While the MaxClique problem of finding the largest cliques in a graph is known to be **NP**-complete in general, here we are not finding maximal cliques, affording an efficient solution.

E. Interactive proofs of quantumness

An interactive proof of quantumness (**??**) comprises two parties, a *prover* and a *verifier*, where the goal is for the prover to prove to the verifier that they have honestly executed a quantum implementation of some function $f(\cdot)$ that cannot be spoofed by classical simulation. The verifier has only classical resources and both parties may classically communicate.

While such protocols are not known in general for arbitrary $f(\cdot)$, they have been described in the context of a restricted class of functions known as trapdoor claw-free functions.

1. Trapdoor claw-free (TCF) functions

Trapdoor claw-free functions (TCF) are a class of cryptographic, 2-to-1, one-way functions,

$$f_{\mathcal{I}}(x) \to w, \tag{1.21}$$

which are classically efficient to evaluate in the forward direction, but for which it is hard to find simultaneously satisfying inputs $\{x_0, x_1\}$ (the 'claw') mapping to the same output,

$$f(x_0) = f(x_1) = w, \tag{1.22}$$

where $x \in \{0,1\}^n$ and $w \in \{0,1\}^{n-1}$ are bit-strings.

Here, $\mathcal{I}$ denotes a problem instance derived from a secret (the trapdoor). If the secret is known, finding claws $\{x_0, x_1\}$ is classically efficient for any w. Since $f(\cdot)$ is easy to evaluate in the forward direction, verifying solutions is classically efficient, and the problem of claw-finding by definition resides in the complexity class **NP**.

2. The LWE problem

A candidate TCF is the lattice-based learning with errors (LWE) problem (**???**). This problem is believed to be post-quantum, where the associated claw-finding problem lies outside of **BQP**, the class of problems efficiently solvable by quantum computers[2].

For matrix,

$$A \in \mathbb{Z}_q^{m\times n}, \tag{1.23}$$

[2] An alternate number-theoretic TCF based on Rabin's function has been described (**??**). Since here the complexity of inverting the trapdoor reduces to integer factorisation this candidate TCF is vulnerable to quantum attack via Shor's algorithm (**?**), making it less applicable in the assumed context of universal quantum computation.

and vectors,

$$x, y, s, e \in \{0,1\}^n, \tag{1.24}$$

related by,

$$y = A \cdot s + e, \tag{1.25}$$

under modulo q arithmetic where q is prime, a TCF may be constructed as,

$$f_{\mathcal{I}}(b, x_b) = \lfloor A \cdot x + b \cdot y \rceil, \tag{1.26}$$

where $b = \{0,1\}$ is a single bit and claws are related by,

$$x_0 = x_1 + s. \tag{1.27}$$

Here, $\mathcal{I} = \{A, y\}$ specifies the problem instance derived from the secret trapdoor $\mathcal{T} = \{s, e\}$ secretly held by the verifier, enabling efficient classical claw-finding and verification if known.

Since $f(x) \to w$ is classically efficient to evaluate in the forward direction, it is easy to find a w for which a single satisfying input x is known. The challenge lies in finding simultaneously satisfying pairs of inputs, believed to be hard for both classical and quantum computers.

3. Interactive proof protocol

Taking a cryptographic TCF function, $f_{\mathcal{I}}(x) \to w$, an interactive proof of quantumness may be implemented as follows:

1. The verifier specifies a problem instance $\mathcal{I}$, without revealing the associated secret $\mathcal{T}$ from which it was derived.

2. The prover prepares a uniform superposition of all length-n bit-strings x via a Hadamard transform,

$$|\psi_H\rangle = \hat{H}^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle . \tag{1.28}$$

3. Evaluating $f_{\mathcal{I}}(x)$ into an output register yields[3],

$$|\psi_{\mathcal{I}}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f_{\mathcal{I}}(x)\rangle ,$$

which may be efficiently prepared using a quantum circuit with,

$$O(n^2 \log^2 n), \tag{1.29}$$

gate count (**?**).

4. The prover measures the output register, obtaining measurement outcome w which is communicated to the verifier. Measuring w collapses the x-register onto the equal superposition of associated satisfying pre-images,

$$|\psi_w\rangle = \frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle) |w\rangle . \tag{1.30}$$

As the x-register was initialised into a uniform superposition over all length-n bit-strings and the TCF is a 2-to-1 function, w is sampled uniformly at random.

5. The verifier specifies a random measurement basis in which the prover should measure qubits in the x register, where $b = \{0 \equiv \hat{Z}, 1 \equiv \hat{X}\}$ correspond to the respective Pauli measurement bases.

6. When measuring in the $b = 0$ ($\hat{Z}$) basis, the prover randomly measures either $m = x_0$ or $m = x_1$, easily verified by direct evaluation of $f(x) \to w$ and comparison with the prover's previously reported w. When measuring in the $b = 1$ ($\hat{X}$) basis verification succeeds if $m \cdot x_0 = m \cdot x_1$. The verification rules are,

$$\begin{aligned} \hat{Z}\,(b=0): &\quad m = \{x_0, x_1\}, \\ \hat{X}\,(b=1): &\quad m \cdot x_0 = m \cdot x_1. \end{aligned} \tag{1.31}$$

7. The above is repeated some constant number of rounds, independently randomising the measurement basis b at every round.

---

[3] The unitarity of quantum circuits prohibits direct evaluation of classical functions on quantum registers in general,

$$\hat{U}_f |x\rangle \not\to |f(x)\rangle ,$$

where $\hat{U}_f$ denotes a quantum circuit evaluating classical function $f(\cdot)$. Introducing ancillary quantum register $|y\rangle$ affords the reversible classical transformation $(x, y) \leftrightarrow (x, y \oplus f(x))$, which may be implemented unitarily in general,

$$\hat{U}_f |x\rangle |y\rangle \to |x\rangle |y \oplus f(x)\rangle .$$

Considering a single output bit of $f(\cdot)$, $\hat{U}_f$ admits the decomposition,

$$\hat{U}_f = \hat{\Pi}_0 \otimes \hat{I} + \hat{\Pi}_1 \otimes \hat{X},$$

where,

$$\hat{\Pi}_i = \sum_{x \,|\, f(x)=i} |x\rangle \langle x| ,$$

are projectors onto the subspaces of x satisfying $f(x) = i$ (Nb: $\hat{\Pi}_0 + \hat{\Pi}_1 = \hat{I}$, $\hat{\Pi}_0 \cdot \hat{\Pi}_1 = 0$). The unitarity of $\hat{U}_f$ follows, independent of $f(\cdot)$,

$$\hat{U}_f^\dagger \cdot \hat{U}_f = (\hat{\Pi}_0 \otimes \hat{I} + \hat{\Pi}_1 \otimes \hat{X})^2 = \hat{I}.$$

Repeating for all output bits and letting $y = 0$ yields,

$$\hat{U}_f |x\rangle |0\rangle \to |x\rangle |f(x)\rangle .$$

The key observation is that since $\hat{X}$ and $\hat{Z}$ measurements do not commute, it is not possible for the prover to know both measurement outcomes simultaneously and therefore must measure in accordance with the verifier's stated measurement basis to pass verification of a single round. While a single round can be classically spoofed if the measurement basis b is known in advance of announcing w, if unknown, b can only be guessed with a probability of $1/2$. Upon repetition, the probability of correctly guessing all measurement bases scales as $p = 1/2^n$ for n rounds, ensuring asymptotic confidence in the honesty of the prover.

4. Consensus protocol

To incorporate IPQs into the QCN framework we require all nodes to act as both prover and verifier for all other nodes.

In the verifier capacity every node prepares a single random TCF instance for all other nodes to prove. Despite solving the same problem instance their proofs will be distinct.

Following the same approach as with QKD we represent the proofs-of-quantumness via a complete graph with the distinction that as this is an asymmetric protocol the graph is now directed (from prover to verifier) with edges in both directions for every node-pair.

Majority votes as per Eq. (1.18) are now made from the verifier perspective.

The additional synchronous steps required to accommodate IPQs are:

1. Nodes commit a single random problem instance $\mathcal{I}_i$.

2. Nodes execute the quantum problem instance specified by every other node j and commit the obtained $w_{i,j}$.

3. Nodes commit the random measurement bases b_i other nodes will be required to measure in.

4. Nodes complete their quantum computations and commit the obtained measurements $m_{i,j}$.

5. Nodes reveal their secrets $\mathcal{T}_i$.

Assuming a verification function analogous to Eq. (1.16),

$$f_{\text{IPQ}}(\mathcal{I}_i, \mathcal{T}_i, w_{i,j}, b_i, m_{i,j}) \to \{0, 1\}, \tag{1.32}$$

similarly defines a directed IPQ compliance graph,

$$G^{\text{comp}}_{\text{IPQ}} : e_{i,j} = f_{\text{IPQ}}(\mathcal{I}_i, \mathcal{T}_i, w_{i,j}, b_i, m_{i,j}) \to \{0, 1\}. \tag{1.33}$$