Quantum random number (QRNG) beacons distinguish themselves from classical counterparts by providing intrinsic unpredictability originating from the fundamental laws of quantum mechanics. Most demonstrations have focused on certifiable randomness generators to guarantee the public that their genuineness is independent from imperfect implementations. These efforts however do not benefit applications where multiple distrusted users need a common set of random numbers, as they must rely on the honesty of beacon owners. In this paper, we formally introduce a design and proof-of-principle experiment of the first consensus protocol producing QRNs in a decentralized environment (dQRNG). Such protocol allows N number of participants contribute in the generation process and publicly verify numbers they collect. Security of the protocol is guaranteed given (N-1) dishonest participants. Our method is thus suited for distribute systems that requires a bias-resistant, highly secure, and public-verifiable random beacon.

INTRODUCTION

Quantum information science is fundamentally changing the way sensitive information is distributed, shared, utilized, and perceived. While quantum key distribution has been a global focus of research and development in this field, there are many other areas where quantum physics principles can be applied to solve previously prohibitive problems and create significant impacts. Among them is the generation and distribution of shared resources amongst distrusted parties over decentralized, unsecured networks.

Suppose Alice, Bob, Dave, and Charles, who are four validators in a blockchain, need to choose one consensus leader among themselves to create the next block in a secure and fair manner, without relying on any centralized randomness source. Suppose a lottery service must prove to claimants that the process of declaring winners is purely random and not controlled by any central entity. The common ground of such scenarios is, in the absence of a trusted third party, how a group of mutually distrusted participants can agree on some genuine random choices for a public settlement among themselves [1]. This is the problem of constructing a random number generator in a decentralized environment (dRNG) where RNs are not only unbiased, unpredictable, tamper-resistant, but also publicly verifiable. Post-election auditing, creating public parameters for cryptographic protocols, gambling and lotteries services, and private preserved messages [4] are some applications requiring dRNGs. In recent years, with the booming of blockchain applications such as distributed computing platforms, smart contracts, and Proof-of-Stake (PoS) based consensus algorithms [5, 6], dRNG has significantly become more demanding. With the rapid rise in the field of quantum random number generators (QRNGs), one might wonder why not exploit the fundamentally unpredictable randomness from quantum process [7, 10, 11, 12, 13, 14] to build a beacon for dRNG. In the past years, most QRNs research focused on creating either a high speed or a self-tested, device-independent generator. In the first scheme, QRNGs are built by fully characterized devices, offering up to Gbits speeds, comparatively as high as classical pseudo-random number generators (PRNGs), to accommodate numerical simulations or gaming service applications [15, 16]. Meanwhile, the latter QRNGs type serves mostly cryptography purposes by delivering rather slower speeds but self-tested capability in which quantum randomness can be monitored and verified without depending on any trusted physical implementations [17, 19]. Nevertheless, these demonstrations benefit only individual users that have physical access to collect QRNs directly from their own devices. To solve the problem of dRNGs, a users must remotely receive numbers broadcasted from a centralized source of quantum randomness. Because there is no way to verify if those outcomes truly originated from quantum process or whether they are protected in transit, the randomness of these numbers become subjective. Beacon users have no choice but to trust the honesty of QRNG manufacturers, or rely on verification from a third party and security of the data transfer process. These two considerable approach however do not solve the problem of dRNG. From the point of view of dRNG users, a QRNG remains a centralized source of randomness that they send request and receive digital RNs, therefore, it only meets the technical trust of users but not the social trust. dRNGs users still have to count on the honesty of some third party. For example, in 2018, NIST built a QRNG beacon where randomness is certified by the impossibility of superluminal signals. This brought RNG beacon to the next level where users no longer worry of adversarial activities on measurement settings, measurement devices, or randomness seeds. Nevertheless, the loophole Bell tests must be isolated from hackers and customers [20], forcing public users to put trust in the beacon owners. Clearly, a crucial element...
is missing in current QRNG technology: not all or no RNs users participate in verifying the randomness publicly. Building on these observations, we recognize an urgent need to unlock more potential of quantum properties into solving these problems. In this paper, we propose the first dQRNG protocol that, on-demand, produces one or one set of random numbers used for decision-making on a matter involving $N$ parties. Our protocol is motivated from both quantum and classical cryptography. We show that this novel decentralized quantum random number generator (dQRNG) protocol is:

- **Genuine**: no participant has control over the final outcomes.
- **Flexible in randomness distribution**: without post-processing, QRNs follow arbitrary probability distribution specified and agreed upon by all $N$ parties.
- **Decentralized**: anyone is able to participate in the QRN generation process.
- **Publicly verifiable**: all $N$ parties have the power to verify the genuineness of the outcome RN(s).
- **Highly tolerant**: outcomes are unpredictable if at least one party is honest among $N$ parties.
- **Quantum secured**: immune to quantum computer attacks.

We provide a general framework of how to implement our protocol into current classical and quantum network infrastructure. Along the way, we demonstrate experimentally dQRNG with four participants and report the robustness of the protocol by showing randomness testing results, probability distribution verification, and fairness for users in QRNs generation process.

## RESULTS

### Previous works

To our best knowledge, there is currently no quantum protocol for dRNG. Thus, we would like to briefly explain how classical techniques generate RNs over a decentralized network before presenting our quantum solution. Many classical consensus protocols have been developed and implemented in recent years [21–23]. A popular approach is producing RNs based on a commit-reveal scheme [24]. In the commit phase, each participant submits the hash of their own secret RN. In the reveal phase, all participants announce the RNs of their choices so that others can compare with the encrypted RN they sent earlier. The final RN is the outcome of a previously agreed upon function combining all secret RNs from the entire pool. A challenge this method faces is that as each participant publicizes their RNs, the last participant to reveal has an advantage of knowing the resulting RN ahead of time. Therefore, the protocol could fail because the last revealer either manipulates the output RN or refuses to submit their secret RN, thus terminating the whole RN generation process. To avoid this, RNs are fed into a verifiable delay function (VDF) with the assumption that the malicious participant does not possess specialized hardware capable of cracking the inherently slow hash computation of the VDF [25]. Other directions for dRNG are built upon threshold digital signature methods where RNs are obtained by amalgamating majority of random signatures from participants [26]. These methods rely on public-private keys scheme and require distributed keys generation process, hence, are vulnerable against quantum computer attacks [27].

### Model

![FIG. 1: Decentralized quantum random number generator (dQRNG) architecture](image)

All communication layers of dQRNG are implemented in parallel corresponding to those in classical communication [28]. Figure 1 depicts quantum related layers of dQRNG. Physical layers are identical to quantum network where there must be an entanglement source, private quantum channels between communication parties, and entanglement verification setup. Detailed schematic illustration of dQRNG physical layers are described in figure 2a and 2b. We apply a model that has been reported in many previous studies as a solution for large-scale quantum networks, in which entangled photons are
passed through a beamsplitter of $N$, distributing to $N$ nodes such that each party shares photon pairs with every other party (total of $P_{num} = \frac{N(N-1)}{2}$ number of possible channel pairs) \cite{2}. Each communication node must have local measurement and data processing devices such as single photon detectors and classical information processors. Transport layer in classical communication is co-used in this protocol for transmission of QRNs between nodes. Unlike quantum network architecture, our protocol can be built without the assumption of secured authentication. Consensus algorithm of dQRNG is operated on classical application layer. All parties in the communication pool are synchronized to a common reference signal of period $T$ to keep track order of their detected photon state. Such a reference can be supplied through the same optical communication channels to distribute the entangled photons (see figure 2b and 2c).

Construction

We first introduce the notations necessary for reading this protocol. Let $N$ be the number of participants, and $k$ be $(N-1)$. This pool would like to agree on a list of random numbers $R$ of length $l$ within a bounded range $B$. Depending on the application, parameter $\mu$, representing the probability of being generated of a random numbers, is decided publicly in advance \cite{30,31}. Parameter $l$ indicates reference identification number or pulse index representing the time slot number of a detected photon. $R^c$ is a list of classical true random numbers generated by participants’ choices, $R^q$ is a list of QRNs generated by quantum process, and $R^{qc}$ is a list representing RNs produced from the dQRNG protocol. $F$ is a function, such as XOR operation or summation operation, agreed ahead of time by participants such that $R = F(R^c)$.

We assume the existence of a quantum network infrastructure, as described in the previous section, connecting all communication nodes. All participants publicly agree upon a common photon measurement basis. We do not assume any necessary authentication between nodes nor existence of any encryption and decryption, given quantum or classical method, for data transferring during the protocol operation. A set of random numbers $R$ is generated in five steps:

Reveal & Consensus 1. Each participant generates and publicizes their own list of random values of length $m$ ($m >> l$) called $R^c_i = \{r^c_{i,m}\}$ and probability weight $\mu_i$ (i indicates participant such as Alice (A), Bob (B), Charles (C), Dave (D), etc; this parameter is optional) $r^c_{i,m} \in N$, $r^c_{i,m} \leq B$. List of all $R^c_i$ are combined to create $R^c = R^c_A \cup R^c_B \cup R^c_C \cup R^c_D \ldots \cup R^c_N = \{r^c_{m,c}\}$, where $m_c = m \times N$. A probability density function (pdf) $C$ is constructed by combining probability weight $\mu_i$.

Quantum measurements. Entangled photon pairs are sent to all participants. Each participant then performs photon measurement privately to obtain their collapsed photon states $R^q_i = \{r^q_{i,m,t}\}$ with corresponding indices $I_i = \{ind_{i,m,t}\}$ ($mt$ is the number of detected photons. In practice, length $mt$ is different for each node depending on the loss of the quantum channel and measurement noise, $mt >> m >> l$).

Quantum results reveal. Measurement results of photon states with corresponding indices are publicly revealed by each participant. Quantum entanglement verification is performed by extracting detected photon results previous step from $P_{num}$ number of channel pairs, $R^q_{ij} = R^q_i \cap R^q_j = \{r^q_{i,m,t}\} \cap \{r^q_{j,m,t}\} = \{r^q_{ij,m,t}\}$ with corresponding list of indices $I_{ij} = I_i \cap I_j$. In the ideal case, $mt_{ij} = mt_i = mt_j$, however, $mt_{ij} < mt_i$ and $mt_j$ due to quantum channel loss and measurement noise. For example, for pair Alice and Bob,
correlated list of $R^q_{AB} = R^q_{A} \cap R^q_{B} = \{r^q_{A,m_{l,q}}\} \cap \{r^q_{B,m_{l,q}}\}$ with corresponding list of indices $I_{AB} = I_{A} \cap I_{B} = \{ind_{A,m_{l}}\} \cap \{ind_{B,m_{l}}\} = \{ind_{AB,m_{l}}\}$.

Similarly, using the same measurement results, probability distribution verification is performed to compare with pdf C.

**Consensus 2.** A list of QRNs is obtained by appending lists of correlated photon measurement results $R^q_{ij}$ from all $P_{num}$ channel pairs, forming $R^q = R^q_{AB} \cup R^q_{AC} \cup R^q_{AD} \cup R^q_{BC} \cup R^q_{BD} \cup R^q_{CD} = \{\{r^q_{m}\}\}$, for all $P_{num}$ and $I = I_{AB} \cup I_{AC} \cup I_{AD} \cup I_{BC} \cup I_{BD} \cup I_{CD} = \{\{ind_{m}\}\}$, for all $P_{num}$ where $m' = \sum_{P_{num}} m_{ij}$.

Quantum randomness is combined with users’ randomness by first publicly perform $I \cap R^c$ to assemble a new list of indices called $I_{final} = \{ind_{m'}\} \cap \{r^c_{m}\}$. Then, the final list of random numbers is filtered so that only values with corresponding indices in $I_{final}$ remain, called $R_{q} = \{r^c_{m}\}$. It can be seen that $R_{q}$ are selected as subset of QRNs harvested in the quantum process by consensus happened before quantum measurements were operated, and therefore, still are QRNs.

**Computing output.** To obtain the list of $R$ random numbers of length $l$ as desired, an aggregation function $F$ (e.g modular addition) is used to calculate publicly and can be confirmed by any participants if needed, $R = F(\{r^c_{m}\})$.

**Analysis**

**Correctness:** For as long as at most $k$ participants are malicious, meaning at least $1$ participants follow the protocol honestly, the results will still be unbiased and unpredictable, the protocol will be executed and all participants will agree upon the final result when the protocol is completed.

**Proof.** Assume the case of four participants Alice, Bob, Charlie, and Dave where they need to agree on one truly random number ($l = 1$) in the range from 0 to 6, given each of number is equally likely to be selected. Alice is the only one being honest ($k = 3$). We proceed the dQRNG protocol to show that the final result is still random and inacculable.

**Reveal & Consensus 1.** Each participant Alice (A), Bob (B), Charles(C), Dave(D) generates and publicizes their own list of random values of length $l$, let’s make $m = 10$:

$$R^c = \{r^c_{A,1 \rightarrow 10}\}, R^c = \{r^c_{B,1 \rightarrow 10}\},$$

$$R^c = \{r^c_{A,1 \rightarrow 10}\}, R^c = \{r^c_{D,1 \rightarrow 10}\},$$

$$R^c = \{r^c_{C,1 \rightarrow 10}\}, R^c = \{r^c_{C,1 \rightarrow 10}\},$$

$$R^c = \{r^c_{D,1 \rightarrow 10}\}, R^c = \{r^c_{D,1 \rightarrow 10}\},$$

$$R^c = \{r^c_{1 \rightarrow 40}\}, m_c = 40,$$

$$C = 1$$

**Quantum measurements.**

$$R^q = \{r^q_{A,m_{l,q}}\}, R^q = \{r^q_{B,m_{l,q}}\},$$

$$R^q = \{r^q_{C,m_{l}}\}, R^q = \{r^q_{D,m_{l}}\},$$

$$(m_{l} \gg 10 > > 1).$$

**Quantum results reveal.** Alice reveals her results honestly while Bob, Charles, and Dave do not. The three of them might cheat without knowing there are other cheaters in the pool. They could alter their results partially or entirely with artificial data. This method, however, does not favor the cheaters in anyway. Quantum entanglement verification easily displays them because their announced results must be compared with each others and with Alice.

In another possible scenario, all three cheaters might consult each other and assemble set of data such that they can pass the quantum entanglement verification with each other in public. At the same time, they still embedded some honest results to verify with Alice. In such situation,

$$R^q = \{r^q_{A,C,B} (non-deterministic),$$

$$R^q = \{r^q_{A,C} (non-deterministic),$$

$$R^q = \{r^q_{A,D} (non-deterministic),$$

$$R^q = \{r^q_{B,C,D} (deterministic),$$

$$R^q = \{r^q_{B,D} (deterministic),$$

$$R^q = \{r^q_{C,D} (deterministic),$$

with corresponding indices behave in the same manner $I_{AB}, I_{AC}, I_{AD}$ are non-deterministic; $I_{BC}, I_{BD}, I_{CD}$ are deterministic.

**Consensus 2.** $R = R^q_{AB} \cup R^q_{AC} \cup R^q_{AD} \cup R^q_{BC} \cup R^q_{BD} \cup R^q_{CD}$ and therefore, it’s a combination of random numbers extracted from quantum process and falsified ones made up by Bob, Charles, and Dave. Likewise, $I = I_{AB} \cup I_{AC} \cup I_{AD} \cup I_{BC} \cup I_{BD} \cup I_{CD}$ is a combination of random indices and fabricate ones. Assume in the first step, $R^c_{B}, R^c_{C}, R^c_{D}$ are agreed and arranged by Bob, Charles, and Dave before they proclaim, $R^c$ still carries private randomness comes from Alice. Thus, the outcome indices of $I_{final} = I \cap R^c$ remains unpredictable. By post-selecting only numbers with indices $I_{final}$ in $R^q$, $R_{q} = \{r^c_{m}\}$ is composed from mixture of quantum random numbers and arbitrary subset of predetermined numbers.

**Computing output.** As a consequence, computing $R = F(\{r^c_{m}\})$ returns an unpredictable number.

Now let’s examine some common attacks on quantum networks and how dQRNG protocols are immune against them.

**Digital emulation attack.** The quantum results reveal step occurs over the classical network, it is vulnerable to digital emulation attack. Without the access to the quantum channel, an adversary can pretend to be one of the participant because the consensus happens in the application layer of the classical channel. Observing others revealing their results first, he then fabricates $R^q_i = \{r^q_{i,m_{l}}\}$ with corresponding indices $I_i = \{ind_{i,m_{l}}\}$ to emulate that he has matching measurement results with every other participants. This malicious action is rather meaningless because it neither
FIG. 4: Experimental setup to demonstrate four parties performing random number generation consensus helps him know the final random numbers result before hand nor manipulate them.

**Man-in-the-middle attack (MITM).** Since we do not assume any secure quantum authentication requirement, an outside adversary can perform MITM attack and take the identity of that node. In this case, the adversary gains full access to quantum measurement results and has the choice to publicize any type of results he wants. His dishonest reveal, if he chooses to do so, does not affect the unpredictability of the final results as proved previously. Participants do not hide their quantum measurement results and corresponding indices, therefore, knowing these information ahead of time does not give the adversary any advantages either.

**Results**

Here we realize the dQRNG protocol in setups purposing for different applications. Table I records parameters used and measured in each experiment. The first type of scenario occurs when a pool of parties repeatedly shares a new random string for cryptography protocols such as private messages or multi-party computation[32]. In particular, we create a case where four parties must generate an 8-bit string periodically. We use pseudo-random number generators (PRNG) to simulate choices of $R^c_i$ list from each party. The rest of the dQRNG steps are executed following the raw data collected from the experiment whose details are explained in section Methods. To examine the uniformity of the dQRNG, we run protocol 100 times with multiple random numbers string are produced each time. Figure ?? indicates the average probability of generating a certain 8-bit value calculated from 100 experiment runs comparing to the theoretically probability. The second type of dQRNG use case happens in scenario requiring a random fair voting mechanism.

| Case | 1 | 2 |
|------|---|---|
| N    | 4 | 4 |
| R    | 8-bit (binary) value | Number represents A, B, C or D |
| B    | [1, 256] | [0, 3] |
| 1    | 1 | 1 |
| $\mu_i$ | N/A | $\mu_A = \mu_B = \mu_C = \mu_D = 0.25$ |
| $R^c_i$ | generated by PRNG | generated by PRNG |
| m    | 8388607 | 8388607 |
| $R^q_{ij}$ | collected from quantum measurements | collected from quantum measurements |

TABLE I: For this typical example, users A, B, C, and D received $m_A = 1029054$, $m_B = 964265$, $m_C = 952180$, and $m_D = 1019553$ photons respectively. This parameter will vary each time the protocol is run. Other parameters that will vary are user-input random number list $R_c$, matching pairs $R^q_{ij}$ with indices $I_{ij}$ where $i,j = A,B,C,D$.
FIG. 5: Case 1: dQRNG protocol is used when four parties must together repeatedly generate an 8-bit random number after a certain time interval for cryptography purposes. The protocol is run 100 times with various amount of 8-bit random numbers produced each time. Typical results of the probability that an 8-bit random number (total 256 possible values) is generated by dQRNG protocol with four participants is recorded. Comparing to the theoretical probability of $\frac{1}{256} = 0.0039$, experimental results fluctuate between 0.0032 and 0.0047 (this gives errors less than 20.5%).

Specific examples could be a winner must be decided in a lottery or a validator must be selected to create the next block in a blockchain protocol. In our experimental demonstration, a pool of four parties must decide who is chosen with the condition that each party equally likely has chance of being chosen ($\mu = 0.25$). Similar to case 1, PRNGs are used to generate list $R_c$ in step 1 of dQRNG while quantum measurement outcomes are used for rest of the steps. We run our protocol 100 times and recorded the average probability results in Figure ??.

DISCUSSION

The invention of distributed applications and blockchain have promised to bring security, fairness, and transparency to digital technologies, business solutions, as well as socioeconomic and political structure by enabling consensus decision in a global scale. Nonetheless, this relies on dRNG and heavily depends on keeping it conspiracy-resistant using current classical communication techniques. It will be deleterious when a fast enough quantum computer breaks the current dRNG and from there, vandalizes this decentralized-everything idea completely. One might argue, a natural approach to quantum proof dRNG could be applying quantum cryptography protocols such as quantum keys distribution, quantum secret sharing, or quantum digital signatures to protect data transfer or signatures encryption [33, 34]. However, in trying to make a scalable and applicable dRNG protocol, we create and present a quantum version that is simple, lightweight. Our dQRNG avoids the cumbersome procedures in quantum cryptography including keys distillation, quantum error correction, or slow key rates, but still holds the quantum advance in being provably random and information-theoretic secure. At the same time, comparing to classical domain, our protocol operates without colluding given only one honest party while requiring no encryption algorithm, and thus, reduces the communication complexity between parties. Furthermore, with the arbitrary probability distribution QRNs generation feature, our protocol outperforms others by avoiding an extra step of complex mathematics and not wasting random numbers when transforming QRNs from uniform distribution to others.

FIG. 6: Case 2: dQRNG protocol is used when four parties must together select a winner among themselves for some voting protocol purposes. The protocol is run 100 times with one party is chosen each time. Comparing to the theoretical probability of 0.25 experimental results only fluctuate less than 5.28% showing the protocol is unbiased.

to protect data transfer or signatures encryption [33, 34].

METHODS

We realize our dQRNG protocol by constructing a simple experimental model comprised of four participants depicted in Figure 4. We first apply the same technique of generating QRNs with arbitrary probability distribution as we introduced in our previous work to create programmable QRN [35]. The difference is that the dQRNG protocol requires broadcasting QRNs to all participants, thus an entangled photons generation and distribution process must follow afterwards. The procedure is described as follow: first, intensity of an 1559.67 nm continuous wave (CW) laser is modulated using an
FIG. 7: Consider the independent timing jitters of the SPD (33 ps) and the TDC (20 ps) at each communication node, photon arrival times collected are rounded to 250 ps per bin-width to compensate the overall timing jitter of all nodes in the network. Joint photon arrival time plots which present correlated QRNs of a typical channel pair are showed in (a), (b), and (c) following uniform, gaussian, and rayleigh probability distributions.

FIG. 8: Left figure: Temporal cross-correlation between photon arrival times of each channel pair in different probability distributions. From left to right are uniform, Gaussian, and Rayleigh distributions. Total coincidence photon pairs are collected over 60s interval delaying within 500 ps around the maximum peak. Right figure: Accidental coincidence counts with photon arrival times rounded to 250 ps binwidth.

electro-optical modulator (EOM) before coupled into a PPLN waveguide for second-harmonic generation (SHG) at 779.8 nm; next, the SHG light is coupled into another identical PPLN waveguide for generating entangled photon pairs via spontaneous-parametric-down-conversion (SPDC) process; and lastly, a system of three 50/50 fiber beamsplitters are used to randomly transmit photons into four different nodes. Note that quantum entanglement verification is an essential procedure for dQRNG protocol. SPDC is ubiquitous for many entanglement-based quantum networks demonstrated thus far. At each node, photons are passed through fiber polarization controller (FPC) before detected by superconducting nanowire single photon detector (SNSPD). In this experiment, using a time-to-digital converter (TDC), we record the photon arrival times compared to a network synchronized 10 MHz reference signal thus harvest the high dimensional temporal mode of each single photon detection. Moreover, the dQRNG protocol is scalable to multiple degrees of freedom in photonics states (ie. polarization, spectral, and spatial modes) for photon efficient generation of QRNs. Data are then transferred to a local machine which is connected to others in the classical network to perform consensus tasks in the application layer.

In the experimental demonstration of our dQRNG protocol, three different pump shapes including CW, Gaussian, and Rayleigh pulses are sent into the SPDC source, showing the versatility in probability distribution of our QRNs. Correlated QRNs of each channel pair \( R_{ij} \) must satisfy the designated distribution to pass the quantum results reveal step. Figure 5 displays joint QRNs between a typical channel pair when photon arrival times are rounded to 250 ps binwidth. Figure 6 (left) describes the time resolved coincidence peaks of all available channel pairs AB, AC, AD, BC, BD, and CD with correspond-
ing CW, Rayleigh, and Gaussian SPDC pump pulses, respectively. Discrepancies in coincidence counts between channel pairs are due to the imperfection of beamsplitters as well as contributions by different detection efficiency and timing jitters of each SPD channel. Using one SPDC source to distribute photon pairs to the entire network reduces the coincidence count rate or correlated QRNs rate. The dQRNG produces one or one small set of RNs at a time, thus, even though the average \( R_{ij} \) rate we achieve is rather low, it does not affect the performance of the protocol. With the same time bin’s width, coincidence to accidental coincidence ratios (CARs) of all possible channel pairs is described in Figure 8b. Security threshold of dQRNG protocol depends on CAR to decide when the SPDC process should end. For example, assuming CAR threshold is 50 for gaussian pump shape, photon pairs generation continue until all channel pairs achieve this value. From our experiment data, this means Alice, Bob, Charles, and Dave each has length \( m_{A} = m_{B} = m_{C} = m_{D} = \), respectively. It can be seen that \( m_{i} >> I = 1 \) in this case. Lastly, following consensus 2 step, we append all \( R_{ij} \) into one list \( R \) and examine this sequence using National Institute of Standards and Technology (NIST) SP 800-22 Statistical Test Suite [30]. Results of a typical QRNs sequence in uniform distribution is presented in figure 9, showing our QRNs pass these test suites with high confidence level. FIG. 9: NIST results of typical 63182 correlated QRNs of uniform probability distribution. QRNs are harvested from 250 ps binwidth photon arrival times and then QRNs are converted into 8-bit representation of 505456 bits and inputted into the NIST test suite. P-values greater than significance level 0.01 certifies this sequence passes the randomness threshold rigorously for standard cryptography purposes.

[1] D. Wasserman, Let them eat chances: Probability and distributive justice, Economics and Philosophy 12 (1996).
[2] A. K. Lenstra and B. Wesolowski, A random zoo: sloth, unicorn, and trx, Cryptology ePrint Archive, Report 2015/366 (2015), https://eprint.iacr.org/2015/366.
[3] T. Baignères, C. Delerablée, M. Finiasz, L. Goubin, T. Lepoint, and M. Rivain, Trap me if you can – million dollar curve, Cryptology ePrint Archive, Report 2015/1249 (2015), https://eprint.iacr.org/2015/1249.
[4] J. van den Hooff, D. Lazar, N. Zeldovich, and M. A. Zaharia, Vuvuzela: scalable private messaging resistant to traffic analysis (Association for Computing Machinery (ACM), 2015) pp. 137–152.
[5] V. Buterin, A next generation smart contract and decentralized application platform Tech. Rep. (2014).
[6] P. Gazi1, A. Kiayias, and D. Zindr, Proof-of-Stake Sidechains Tech. Rep. (2018).
[7] C. Gabriel, C. Wittmann, D. Sych, R. Dong, W. Maurer, U. L. Andersen, C. Marquardt, and G. Leuchs, A generator for unique quantum random numbers based on vacuum states, Nat Photon 4, 711 (2010).
[8] Y. Shi, B. Chng, and C. Kurtsiefer, Random numbers from vacuum fluctuations, Applied Physics Letters 109, 041101 (2016).
[9] E. O. Samsonov, B. E. Pervushin, A. E. Ivanova, A. A. Santev, V. I. Egorov, S. M. Kynev, and A. V. Gleim, Vacuum-based quantum random number generator using multi-mode coherent states (2020), 2004.06552 [quant-ph].
[10] Q. Zhou, R. Valivarthi, C. John, and W. Tittel, Practical quantum random-number generation based on sampling vacuum fluctuations, Quantum Engineering 1, e8 (2019), https://onlinelibrary.wiley.com/doi/pdf/10.1002/que2.8.
[11] B. Qi, Y.-M. Chi, H.-K. Lo, and L. Qian, High-speed quantum random number generation by measuring phase noise of a single-mode laser, Opt. Lett. 35, 312 (2010).
[12] H. Guo, W. Tang, Y. Liu, and W. Wei, Truly random number generation based on measurement of phase noise of a laser, Phys. Rev. E 81, 051137 (2010).
[13] Y.-Q. Nie, H.-F. Zhang, Z. Zhang, J. Wang, X. Ma, J. Zhang, and J.-W. Pan, Practical and fast quantum random number generation based on photon arrival time rel-
ative to external reference. Applied Physics Letters 104, 051110 (2014) [http://dx.doi.org/10.1063/1.4863224]

[14] F. Xu, J. H. Shapiro, and F. N. C. Wong, Experimental fast quantum random number generation using high-dimensional entanglement with entropy monitoring. Optica 3, 1266 (2016)

[15] Z. Z. Y, H. W. Y, S, and G. H, 6 gbps real-time optical quantum random number generator based on vacuum fluctuation, Rev Sci Instrum (2019).

[16] K. Kim, Massively parallel ultrafast random bit generation with a chip-scale laser, Science 371, https://doi.org/10.1126/science.abc2666 (2021).

[17] T. Lunghi, J. B. Brask, C. C. W. Lim, Q. Lavigne, J. Bowles, A. Martin, H. Zbinden, and N. Brunner, Self-testing quantum random number generator, Phys. Rev. Lett. 114, 150501 (2015)

[18] M. Avesani, D. G. Marangon, G. Vallone, and P. Villegas, Source-device-independent heterodyne-based quantum random number generator at 17 gbps, Nature Communications 10.1038/s41467-016-0019-0 (2018).

[19] L. Y, Z. Q, L. M.H, G. J.Y, Z. Y, Bai.B, Zhang.W, Liu.W.Z, Wu.C, Yuan.X, L. H, Munro.W.J, W. Z, You.L, Zhang.J, Ma.X, Fan.J, Zhang.Q, and Fan.J.W, Device-independent quantum random-number generation, Nature 10.1038/s41586-018-0558-4 (2018).

[20] P. Bierhorst, E. Knill, S. Glancy, Y. Zhang, A. Mink, S. Jordan, A. Rommal, Y.-K. Liu, B. Christensen, S. W. Nam, M. J. Stevens, and L. K. Shalm, Experimentally generated randomness certified by the impossibility of superluminal signals, Nature 556, 10.1038/s41586-018-0019-0 (2018).

[21] P. Schindler, A. Judmayer, N. Stifter, and E. Weippl, Hydrand: Efficient continuous distributed randomness, in 2020 IEEE Symposium on Security and Privacy (SP) (2020) pp. 73–89.

[22] T. Nguyen Van, T.-D. Le, T. Nguyen-Anh, M.-P. Nguyen, T. Nguyen-Van, M.-Q. Le-Tran, Q. Le, H. Pham, and K. Nguyen-An, A system for scalable decentralized random number generation (2019).

[23] E. Syta, P. Jovanovic, E. K. Kogias, N. Gaillly, L. Gasser, I. Khoffi, M. J. Fischer, and B. Ford, Scalable bias-resistant distributed randomness, Cryptology ePrint Archive, Report 2016/1067 (2016), [https://eprint.iacr.org/2016/1067]

[24] Randao: Verifiable Random Number Generation, Tech. Rep. (2017).

[25] D. Boneh, J. Bonneau, B. Bunz, and B. Fisch, Verifiable delay functions, Cryptology ePrint Archive, Report 2018/601 (2019), [https://eprint.iacr.org/2018/601] pdf

[26] A. Skidanov. Near Protocol Randomness Beacon Tech. Rep. (2019).

[27] B. Libert, M. Joye, and M. Yung, Born and raised distributively: Fully distributed non-interactive adaptively-secure threshold signatures with short shares, Proceedings of the Annual ACM Symposium on Principles of Distributed Computing 645 (2014)

[28] V. Cerf and E. Cain, The dod internet architecture model, Comput. Networks 7, 307 (1983).

[29] X. Liu, X. Yao, R. Xue, H. Wang, H. Li, Z. Wang, L. You, X. Feng, F. Liu, K. Cui, Y. Huang, and W. Zhang, An entanglement-based quantum network based on symmetric dispersive optics quantum key distribution, APL Photonics 5, 076104 (2020) [https://doi.org/10.1063/5.0002595]

[30] C. T. Nguyen, D. T. Hoang, D. Nguyen, D. Niyato, H. Nguyen, and E. Dutkiewicz, Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities, IEEE Access 7, 85727 (2019).

[31] F. Sales, Blockchain without Waste: Proof-of-Stake, The Review of Financial Studies 34, 1156 (2020) [https://academic.oup.com/rfs/article-pdf/34/3/1156/3624599/klua075_supplementary_data.pdf]

[32] C. Baum, I. Damgard, and C. Orlandi, Conference on Security and Cryptography for Networks 8642, 175 (2014).

[33] M. Hillery, V. Bužek, and A. Berthiaume, Quantum secret sharing, Physical Review A 59, 1829–1834 (1999)

[34] R. Collins, R. Amiri, M. Fujikawa, and et al., Experimental demonstration of quantum digital signatures over 43 db channel loss using differential phase shift quantum key distribution, Scientific Report 7, https://doi.org/10.1038/s41598-017-03401-9 (2017).

[35] L. Nguyen, P. Rehain, Y. M. Sua, and Y.-P. Huang, Programmable quantum random number generator without postprocessing, Opt. Lett. 43, 631 (2018)

[36] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, N. Heckert, J. Dray, and S. Vo, A statistical test suite for random and pseudorandom number generators for cryptographic applications (2001).

[37] R. F. W. Coates, G. J. Janacek, and K. V. Lever, Monte Carlo simulation and random number generation, IEEE Journal on Selected Areas in Communications 6, 58 (1988)

[38] F. James and L. Moneta, Review of high-quality random number generators, Computing and Software for Big Science 4, 10.1007/s41781-019-0034-3 (2020).

[39] M. O. Rabin, Transaction protection by beacons, Journal of Computer and System Sciences 27, 256 (1983)

[40] Anu quantum random numbers

[41] Random.org true random numbers service