# Designing Quantum-Safe Network Architectures via Certified Randomness Distribution

Filip-Ioan Ceară[1,2] and Diana Maimuţ[1,2]

[1] Advanced Technologies Institute, Bucharest, Romania
{filip.ceara,diana.maimut}@dcti.ro
[2] University of Science and Technology POLITEHNICA Bucharest
060042 Bucharest, Romania

**Abstract.** We propose an alternative design for quantum-safe network architectures, based on distributing certified randomness generated by different trust-level QRNGs. Our modular methodology aggregates high-trust entropy from upper network layers with local contributions, enhancing the security guarantees of locally available random sequences. Leveraging the physical guarantees of QRNGs and symmetric-key cryptographic algorithms for node authentication, our approach enables immediately deployable, quantum-safe infrastructures. Such a paradigm bypasses the practical security limitations of current QKD-based solutions, while avoiding massive key sizes and latency inherent to asymmetric PQC implementations.

**Keywords:** quantum random number generator, distributed randomness, quantum-safe networks, post-quantum cryptography

## 1 Introduction

### 1.1 Context and Motivation

We introduce a novel approach to implementing quantum-safe networks, as an alternative to the *de facto* standard, based on Quantum Key Distribution (QKD) equipments. Inherently resistant to quantum cryptanalysis, quantum networks represent a specialized auxiliary system to classical architectures [47]. Their structure consists of a hybrid framework that layers quantum components over existing classical infrastructure. The physical layer comprises standard telecommunication fiber or satellite links to transmit quantum information. To overcome distance-dependent channel attenuation, current topologies rely on trusted nodes. The hardware layer is supported by a parallel classical control channel for network synchronization and a key management system delivering the resulting cryptographic keys.

Despite its theoretical promise for information-theoretic security (such as the no-cloning theorem), according to very recent position papers from various prominent organizations such as NATO or NSA [40, 41], QKD remains hindered by functional limitations and insufficient maturity from a security perspective,

restricting its viability largely to specialized niche applications. Similar views started to be publicly exposed since a couple of years ago by top security agencies [15]: QKD requires dedicated physical infrastructure and trusted relay nodes over extended distances, which substantially increases deployment complexity while failing to meet the scalability and operational demands of modern networks.

Proving, modeling, and evaluating QKD systems is a highly complex task, as it requires balancing abstract, mathematically rigorous security proofs with practical concerns such as device imperfections [33,44]. Consequently, the scarcity of standardized protocols, coupled with unresolved vulnerabilities regarding side-channel and physical implementation attacks [7], underlines that QKD is not ready yet to supersede classical key agreement or emerging Post-Quantum Cryptography (PQC) frameworks. While very promising research directions regarding different QKD design paradigms were recently published [27], the results are merely theoretical. All in all, as the concept of "unconditional security" can be seen as overrated and oversold in the quantum networking world, our proposal is a notable alternative.

On the other hand, the PQC transition fosters long-term security in a quantum environment, yet migrating legacy cryptosystems to newly NIST[3] standardized public-key primitives [37, 38] presents serious practical challenges. This process demands significant hardware upgrades and extensive system redesigns. Furthermore, asymmetric PQC is characterized by massive key sizes and high computational overhead, introducing severe communication latency that makes them impractical for high-throughput, low-latency network applications.

To bypass these operational barriers, this paper introduces a novel paradigm for quantum-safe networks by avoiding computational penalties. Instead of relying on heavy lattice-based public-key primitives, we demonstrate that optimal network efficiency can be achieved by utilizing symmetric-key quantum-resistant cryptographic algorithms (inherently more resilient against known quantum adversaries). We stress that our approach offers an immediately deployable alternative for modern networks. Concretely, we leverage the physical guarantees of certified randomness generated by Quantum Random Number Generators (QRNGs). We integrate this high-entropy randomness with well-established symmetric-key based Authenticated Key Exchange (AKE) protocols designed for optimal speed, specifically drawing upon [3]. By combining the certified entropy of QRNGs with symmetric primitives, our framework requires significantly fewer changes to existing architectures compared to the PQC migration process. In a nutshell, we provide an efficient proposal for securing distributed networks in the quantum era without sacrificing throughput or expanding the hardware attack surface.

### 1.2 Related Work

Given that, to the best of our knowledge, the current paper is the first work addressing entropy aggregation from different trust assumption QRNGs, having as a central piece the concept of *certified randomness* (not to be mistaken

[3] National Institute of Standards and Technology

for verifiable randomness), we further discuss, at a concept level, distributed randomness as it appeared in the literature in previous contexts.

*Distributed Randomness for Decentralized Systems.* Distributed randomness became a core component in decentralized systems, providing unbiasability, unpredictability, and public verifiability against a resource-bounded adversary. To eliminate traditional randomness generation vulnerabilities (*e.g.* single point of failure), modern protocols leverage mechanisms like *distributed randomness beacons* (DRBs), ensuring that no single party or malicious coalition can manipulate or foresee the output. By shifting the trust assumption from an isolated entity to a distributed network, beacons guarantee that the generated entropy is both cryptographically secure and auditable by any network participant in real time. Protocol initialization typically leverages *distributed key generation* (DKG) to establish a joint public key across an $n$-party network without relying on a trusted dealer, enforcing a $t$-out-of-$n$ threshold structure. To achieve deterministic execution and non-interactive verification, modern frameworks utilize *verifiable unpredictable functions* (VUFs) and threshold variants of pairing-based cryptography. Furthermore, to counteract front-running, rushing attacks, and selective abort strategies, modern architectures integrate *verifiable delay functions* (VDFs) and *publicly verifiable secret sharing* (PVSS).

The authors of [9] classify distributed randomness into provider selection and output generation, emphasizing that a robust combining method must preserve uniformity as long as at least one source is honest. Specifically, [17] demonstrates that relying on a single entropy source exposes protocols to a *private beacon attack*, where a corrupt participant can predict future outputs at no cost. In practice, modern frameworks like HydRand [45] and AsyRand [51] implement these distributed requirements via iterative hash-chain constructions. This execution pattern can be uniformly expressed as $R_e = H(R_{e-1}, \sigma_e)$, where $R_e$ denotes the beacon output for epoch $e$, $H(\cdot)$ represents a cryptographic hash function, and $\sigma_e$ is the epoch-specific randomness contribution satisfying the necessary multi-party entropy bounds. Within this framework, $\sigma_e$ corresponds to a threshold-reconstructed secret $g^s$ in AsyRand, whereas in HydRand, it denotes an accumulated network entropy component combined with the previous state.

*Hierarchical Distributed Randomness Constructions.* The hierarchical paradigm formalized in [39] introduces a *random bit generator construction* (RBGC) based on a tree topology. In this architecture, a root generator leverages an initial entropy source to seed child nodes, which subsequently propagate seeds down to non-root generators. Mirroring the multi-party robustness goals of decentralized beacons, this standard relies on combining independent entropy sources via exclusive-or operations after appropriate conditioning. This constraint ensures that the combined output preserves the entropy of the strongest underlying source, forcing an adversary to compromise all constituent inputs to gain control over the system's randomness.

*Aggregated Entropy in Operating Systems.* Currently, both Linux and Windows operating systems use entropy pools. The Linux Kernel 5.17+ utilizes a single BLAKE2s [1] input pool to absorb environmental noise. Once 256 bits of initial entropy are credited, a ChaCha20-based [2] fast cryptographically secure pseudorandom number generator (CSPRNG) is seeded. To ensure *forward secrecy*[4], Linux employs *fast-key-erasure*, immediately overwriting the ChaCha20 key with generated output after fulfilling a request. The system is continuously reseeded asynchronously to guarantee *backward secrecy*[5]. On the other hand, Windows 10+ implements a Fortuna-inspired [13,14] multi-pool accumulator using SHA-512 [35]. To prevent attackers from predicting the internal state before sufficient unknown entropy accumulates, it utilizes an exponential $3^k$ *reseed schedule*[6]. This pooled entropy feeds an AES-CTR-DRBG-type generator, compliant with [36].

### 1.3 Structure of the Paper

In Section 2 we introduce notations and discuss basic information regarding quantum random number generators. In Section 3, we detail our proposed quantum-safe architectures, with a special focus on novel flavors of randomness distributions. Concretely, we present mathematical formalizations of key concepts such as the *entropy capsule* and address security concerns. We implement proof-of-concept use cases and compare the results against standardized post-quantum lattice-based solutions in Section 4, where we also recall the general case authenticated key exchange protocol of [3], as we instantiate it for our benchmark. We conclude in Section 5.

*Notations.* During the following, $\wedge$ denotes the logical and function, while $\|$ represents string concatenation.

## 2 Quantum Random Number Generators

Random number generators (RNGs) can be viewed as an essential building block in cryptography. To an extent, we can directly link the security of a cryptographic algorithm to the source of entropy it relies on. A high-level, simplistic classification of RNGs dissociates between true random number generators (TRNGs), pseudo-random number generators (PRNGs) and hybrid flavors of the two previously mentioned classes. However, depending on the chosen reference, corresponding notions may differ [24,36,39]. Note that for security applications, CSPRNGs are considered.

A special class of TRNGs consists of QRNGs. Such generators can be considered one of the most mature *quantum technologies* [18], as per the current state of the *second quantum revolution* [22]. Such generators exploit the intrinsic uncertainty specific to quantum measurements to generate random variables.

[4] previously generated bits are impossible to recover
[5] future bits should be impossible to predict
[6] pool 0 is used every reseed, pool 1 every $3^{\text{rd}}$, pool $k$ every $3^{k\text{-th}}$

Random sequences can be easily obtained by sampling the resulting random variables, but for security guarantees, one must model the connection between the obtained random variables and the quantum system that was employed in the construction of the device.

Besides classifying QRNGs based on the quantum process being harnessed, another relevant classification, in the context of security applications, considers the relationship between random variables and physical phenomena. Thus, we distinguish between three types of QRNG families:

1. *Device-Dependent* (DD) or, alternatively, *trusted* (for more information, see Appendix A);
2. *Semi-Device-Independent* (Semi-DI) (for more information, see Appendix B);
3. *Device-Independent* (DI) [18,31] (for more information, see Appendix C).

The assessment of a QRNG to one of the three classes is done based on a *trust model*, *i.e.* a mathematical description of the employed random generation mechanism. Such a model considers: (1) the quantum physical process, (2) the variance of the physical parameters of the additional components and (3) the adversarial side-information assumptions.

*Remark 1.* A QRNG's level of assurance is inversely related to the feasibility of its implementation in an existing infrastructure. The maximum level is associated to DI QRNGs, but constructing a valid such generator implies solving complex physical and engineering problems[7]. However, even if deploying a DD QRNG can be considered easier[8], it implies trusting the theoretical model of the manufacturer.

We define the certification of a QRNG by means of a quantitative statement about the behavior of the QRNG under a specifically stated trust model. The BSI[9] standard [24] explicitly states that RNGs must provide a stochastic description of the entropy source while in an evaluation process. Both [34] and [24] rely on *min-entropy* as a security parameter used for the description of the resulting output sequences. Min-entropy quantifies the probability of the most likely outcome and, therefore, captures the worst-case guessing probability of an adversary.

**Definition 1.** *For a classical random variable $X$ with distribution $p_X(x)$, we define min-entropy as*

$$H_{\min}(X) = -\log_2\Big(\max_x p_X(x)\Big)\,. \tag{1}$$

If an adversary has no side information, the maximum value of the guessing probability is $p_{\text{guess}}(X) = 2^{-H_{\min}(X)}$. In security analyzes, the adversary may have access to supplementary information $E$. For a classical-quantum state $\rho_{XE}$, conditional min-entropy is connected to the adversary's optimal measurement by

$$H_{\min}(X|E) = -\log_2 p_{\text{guess}}(X|E). \tag{2}$$

[7] *e.g.*, implementing loophole CHSH inequality verification
[8] less operational requirements
[9] Bundesamt für Sicherheit in der Informationstechnik

While min-entropy quantifies the unpredictability of the generated outcomes and is used as a metric that evaluates the quality of the generated sequences, statistical test suites, *e.g.*, [34], assess their observable statistical properties.

*Remark 2.* Statistical test suites, *e.g.* [34], are merely guidelines for the quality of the generated bit strings, representing only a necessary condition for evaluating an RNG, and not itself a security guarantee. Therefore, statistical testing serves as a complementary validation mechanism rather than an independent security guarantee.

In addition to security guarantees, QRNG implementations can be evaluated in terms of their generation rates. The raw sampling rate represents the number of measurement outcomes acquired per second, while the certified entropy rate quantifies the amount of conditional min-entropy generated per unit time under the assumed trust model. The extracted bit rate represents the rate of random bits obtained after applying a randomness extractor, whose parameters depend on the certified entropy and the required security guarantees. Consequently, the performance of a QRNG must be assessed in relation to its underlying security model, as a high raw sampling rate does not necessarily imply a high rate of certified random bits.

## 3 Distributed Randomness with a Quantum Flavor

To harness the capabilities of existing QRNGs, we propose an architecture allowing randomness[10] to accumulate and be distributed across multiple network levels. In this way, high-quality randomness generated by less practical sources can be outsourced to less costly QRNGs, while each node preserves its local entropy contribution, as well as its own trust assumptions. Thus, the network's role is allow each node to strengthen its local randomness state (with contributions from other nodes operating under different assurance assumptions), rather than replacing it.

Our solution is inspired by the mechanism employed for synchronization distribution in classical networks [20,21], in which atomic clocks[11] are at the top of the hierarchy. At lower levels, local clocks receive the periodic signal generated by the nodes above as reference. Following up, we propose an architecture organized as hierarchical layers, according to the trust model of the employed QRNGs. Higher-assurance generators are placed in the upper layers and periodically provide randomness contributions to lower layers composed of semi-DI and trusted QRNGs. Each receiving node aggregates the randomness sent from upper layers, supplementing it with its own contribution. Hence, the resulting randomness can be locally used or further propagated towards the next layer.

We base our construction on the fact that randomness generation, confidentiality, and authentication represent different security flavors. Although these

[10] generated under different trust models
[11] having the lowest jitter of all possible clock sources

objectives interact during the randomness distribution throughout the network, we analyze them separately. Therefore, we propose, as illustrated in Figure 1, an architecture consisting of three logically independent layers: (1) the entropy distribution layer, (2) the authentication layer, and (3) the secure-transport layer.

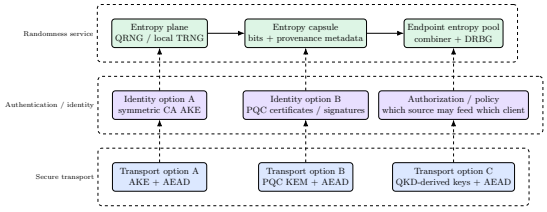



Fig. 1: Our Proposed Layered Architecture

Such a separation allows the randomness-distribution mechanism to remain independent of the specific cryptographic infrastructure used to authenticate nodes or protect network communication.

### 3.1 The Entropy Distribution Layer

The QRNG trust model is an attribute of the entropy distribution layer. The differences between a DD, semi-DI and DI QRNG do not interfere with the internals of the algorithms ensuring confidentiality and (entity) authentication.

In our proposed architecture, at each level, randomness is (1) received from the layer above and (2) combined with the locally generated entropy. This construction allows the network to preserve the contribution of a higher-assurance node while maintaining the independence of the local entropy generation mechanism.

For a node $N_i$ and a given time $t$, we denote the randomness received from the upper layer by $R_i$, and the entropy generated by the local QRNG by $X_i$. We model a stateless aggregating mechanism, as follows:

$$\mathsf{Accumulator}_i = (\mathsf{Setup}_i, \mathsf{Local}_i, \mathsf{Contribute}_i)\,. \tag{3}$$

The function $\mathsf{Setup}_i$ validates the inputs of the current execution, generates the local entropy contribution, and constructs a transient object that contains the entropy-bearing material and its conservative entropy budget. The function $\mathsf{Local}_i$ conditions this material by means of a cryptographic hash function for local computational use. Finally, $\mathsf{Contribute}_i$ applies an explicit *randomness extractor* when the node provides a new randomness contribution to another node or layer. The transient object created by $\mathsf{Setup}_i$ is discarded when the current execution has finished.

We denote the network object used to transport randomness contributions and additional metadata as an *entropy capsule* and is denoted $\mathcal{EC}$. At a given moment $t$, we define $\mathcal{EC}_t$ as

$$\mathcal{EC}_t = (v_t, ID_t, Class_t, e_t, q_t, \tau_t, R_t, \ell_t, h_t^R, \mathcal{H}_t, \Delta_t). \tag{4}$$

We refer the reader to Appendix D for a detailed description of each parameter.

During the following, we formalize the stateless mechanism underlying the entropy-distribution layer.

### 3.1.1 Stateless Accumulator

At each node, the accumulator validates the incoming $\mathcal{EC}s$ according to a local policy, combines accepted remote contributions with fresh local entropy under a conservative entropy budget, and derives randomness either for local use or for subsequent distribution without maintaining persistent entropy-bearing state.

Let $s$ be a security parameter and let $pp_i$ denote the public parameters and configuration of the accumulator associated with node $N_i$.

We consider

$$pp_i = (ID_{N_i}, \mathcal{P}_i, s, H, \mathrm{Ext}), \tag{5}$$

where $\mathcal{P}_i$ is a local entropy policy, $H$ is a cryptographic hash function used for local conditioning, and Ext denotes the selected randomness extractor. These parameters contain no entropy-bearing secret state.

*Accumulator Initialization.* Our proposed construction is stateless: initialization creates no entropy pool or persistent randomness state. Each execution uses only current contributions: root nodes obtain fresh local QRNG entropy and intermediate QRNG nodes may combine accepted remote with local entropy, while nodes without a physical source begin with zero local entropy and provide output only after sufficient accepted entropy is received. Public initialization values are never credited as randomness. Replay state, such as $\mathrm{last}_i[ID]$, belongs to capsule acceptance rather than the accumulator and may be maintained locally or provided by an authenticated anti-replay transport mechanism.

*Entropy Limitation.* The incoming $\mathcal{EC}$ is verified based on a policy check routine. Each receiving node $N_i$ applies a local policy $\mathcal{P}_i$ to validate an entropy capsule before using it.

We define the policy

$$\mathcal{P}_i = \left(\mathcal{S}_i^{\mathrm{allow}}, \mathcal{C}_i^{\mathrm{allow}}, \mathcal{V}_i, \ell_i^{\min}, \ell_i^{\max}, \Delta_i^{\max}, h_i^{\min}, h_i^{\max}\right), \tag{6}$$

where $\mathcal{S}_i^{\mathrm{allow}}$ represents the set of allowed identities, $\mathcal{C}_i^{\mathrm{allow}}$ represent the set of permitted trust classes and $\mathcal{V}_i$ refers to the capsule's version. The received payload length should be between $[\ell_i^{\min}, \ell_i^{\max}]$, while the $\Delta_i^{\max}$ denotes the maximum accepted staleness; $h_i^{\min}$ represents the min-entropy lower bound, while $h_i^{\max}$ denotes the upper bound of remote entropy the current node is willing to accept.

For a capsule $\mathcal{EC}_t$ having payload $R_{i,t} \in \{0,1\}^{\ell_{i,t}}$, we define:

$$\begin{aligned}\mathsf{PolicyCheck}(\mathcal{EC}_{i,t}, \mathcal{P}_i) \equiv \big(ID_t \in \mathcal{S}_i^{\mathrm{allow}}\big) \wedge \big(Class_t \in \mathcal{C}_i^{\mathrm{allow}}\big) \wedge \big(v_t \in \mathcal{V}_i\big) \\ \wedge \big(\mathcal{H}_t = \mathsf{OK}\big) \wedge \big(|R_{i,t}| = \ell_{i,t}\big) \wedge \big(\ell_i^{\min} \le \ell_t \le \ell_i^{\max}\big) \\ \wedge \big(t_{\mathrm{now}} - \tau_t \le \min\{\Delta_t, \Delta_i^{\max}\}\big) \wedge \big(h_t^R \ge h_i^{\min}\big) \wedge \big((e_t, q_t) > \mathsf{last}_i[ID_t]\big).\end{aligned} \tag{7}$$

The remote entropy of an accepted capsule is bounded by the claimed value, the payload length, and the receiver's local policy:

$$\widehat{h}_{i,t}^{R} = \min\left\{h_t^R, \ell_t, h_i^{\max}\right\}. \tag{8}$$

Hence, we define the context-verification procedure as

$$\mathsf{CheckContext}\left(\mathcal{EC}_{i,t}, \mathcal{P}_i, \mathsf{st}_i\right) = \begin{cases} \left(\text{Accept}, \widehat{h}_{i,t}^{R}\right), & \text{if } \mathsf{PolicyCheck}(\mathcal{EC}_{i,t}, \mathcal{P}_i) = 1, \\ \left(\text{Reject}, 0\right), & \text{otherwise.} \end{cases} \tag{9}$$

The current processing index is incremented only after the capsule has been accepted for current execution. If several capsules are available, the same verification is applied independently to each of them. The accepted set is denoted by $\mathcal{A}_{i,t} \subseteq \mathcal{EC}_{i,t}$. An architecture-specific policy may additionally require a minimum number or a particular diversity of accepted sources before $\mathsf{Setup}_i$ is allowed to continue.

After the remote capsules have been accepted, node $N_i$ generates a fresh local contribution, denoted by $X_{i,t} \in \{0,1\}^{m_{i,t}}$. We assume that the trust model and security analysis of the local QRNG establish $H_{\min}(X_{i,t} \mid E_{i,t}) \geq h_{i,t}^{X}$, where $E_{i,t}$ denotes the adversarial side information relevant for the current execution. For a shared combining service without a local QRNG, we set $X_{i,t} = \perp$ and $h_{i,t}^{X} = 0$. Similarly, a root QRNG can execute the combiner with $\mathcal{A}_{i,t} = \emptyset$ and use only the fresh local contribution.

Let the accepted set contain $n$ capsules and denote their claimed entropy by $\widehat{h}_{i,t}^{R,(1)}, \ldots, \widehat{h}_{i,t}^{R,(n)}$. We do not assume that the remote contributions are mutually independent or independent of the local contribution. Therefore, their individual entropy claims are not added. Instead, the node $N_i$ applies

$$h_{i,t}^{\text{joint}} = \max\left\{\widehat{h}_{i,t}^{R_1}, \ldots, \widehat{h}_{i,t}^{R_n}, h_{i,t}^{X}\right\}. \tag{10}$$

For a single $\mathcal{EC}$, $h_{i,t}^{\text{joint}} = \max\left\{\widehat{h}_{i,t}^{R}, h_{i,t}^{X}\right\}$. The value $h_{i,t}^{\text{joint}}$ represents a conservative lower bound on the entropy available in the complete input under the adopted side-information model.

*Canonical Context Construction.* For each accepted $\mathcal{EC}_{i,t}^{(k)} \in \mathcal{A}_{i,t}$, we construct

$$C_{i,t}^{(k)} = \mathsf{Encode}\left(v_t^{(k)}, ID_t^{(k)}, Class_t^{(k)}, e_t^{(k)}, q_t^{(k)}, \tau_t^{(k)}, \ell_t^{(k)}, h_t^{R,(k)}, \mathcal{H}_t^{(k)}, \Delta_t^{(k)}\right). \tag{11}$$

The complete context of the current execution is then

$$C_{i,t} = \mathsf{Encode}\left(ID_{N_{i,t}}, C_{i,t}^{(1)}, \ldots, C_{i,t}^{(n)}\right), \tag{12}$$

where $ID_{N_{i,t}}$ represents the identity of the receiving node $N_i$. The fields contained in $C_{i,t}$ represent public protocol metadata and therefore, their provided entropy $h(C_{i,t}) = 0$. Including the receiver's node identity ensures recipient separation

when an upstream node distributes $\mathcal{EC}s$ toward several descendants. The $\mathsf{Encode}$ function can simply be a specific cryptographic hash function.

To avoid ambiguity, the entropy-bearing inputs are framed before they are aggregated. For every accepted remote payload, we define $R^{e,(k)}_{i,t} = \mathsf{length}(\ell^{(k)}_{i,t}) \parallel R^{(k)}_{i,t}$, and, when a local contribution exists, $X^{e}_{i,t} = \mathsf{length}(m_{i,t}) \parallel X_{i,t}$. Function $\mathsf{length}$ should be chosen such that it returns a fixed-length byte representation of its input, using a unique canonical encoding agreed upon by all nodes.

The randomness existing at the current execution is

$$W_{i,t} = R^{e,(1)}_{i,t} \parallel \cdots \parallel R^{e,(n)}_{i,t} \parallel X^{e}_{i,t}, \tag{13}$$

where absent terms are omitted. For a root node, $W_{i,t} = X^{e}_{i,t}$. For an aggregating service without a local QRNG, $W_{i,t}$ contains only accepted contributions.

*Setup Function.* $\mathsf{Setup}_i$ outputs $\omega_{i,t} = \left(C_{i,t}, W_{i,t}, h^{\text{joint}}_{i,t}\right)$. Therefore, $\omega_{i,t} = \mathsf{Setup}_i\left(\mathcal{EC}_{i,t}, P_i, X_{i,t}\right)$. The value $\omega_{i,t}$ is not a persistent state, in the sense that it exists only for current executions and is erased after the local and distribution operations associated with that execution have completed.

*Usage of Local Randomness.* One possible use of $\omega_{i,t}$ is local computational conditioning. We define $K^{\text{local}}_{i,t} = \mathsf{Local}_i(\omega_{i,t})$ as

$$K^{\text{local}}_{i,t} = H_s\left(C_{i,t} \parallel W_{i,t}\right), \tag{14}$$

where $H_s$ denotes a cryptographic hash function whose output has length $s$. The value $K^{\text{local}}_{i,t}$ can be supplied as entropy input to instantiate or reseed operation of a local CSPRNG. The hash function referred in (14) is used only for computational conditioning, not as an information-theoretic extractor. Therefore, $h^{\text{joint}}_{i,t}$ is not assigned directly to $K^{\text{local}}_{i,t}$. Any deterministic leakage about $W_{i,t}$ revealed before export must be included in the adversary's side information.

*Contribution for the Next $\mathcal{EC}$.* The function $\mathsf{Contribute}_i$ is used when node $N_i$ acts as an entropy provider. Let $\mathcal{D}_{i,t} = \{N_{j_1}, \ldots, N_{j_d}\}$ represent the set of descendants that receive a contribution, and let $\ell^{\text{out}}_{i\to j_k,t}$ denote the number of bits assigned to recipient $N_{j_k}$. Thus, the total number of extracted bits is

$$L_{i,t} = \sum_{k=1}^{d} \ell^{\text{out}}_{i\to j_k,t}. \tag{15}$$

Let $\chi_{i,t}$ be a public extractor seed independent of $W_{i,t}$. As $W_{i,t}$ may contain adversarially chosen upstream randomness, $\chi_{i,t}$ is sampled from the local QRNG only after $\mathsf{CheckContext}$ has fixed $W_{i,t}$. The seed is drawn from a disjoint sampling window, is fresh for each execution and contributes no entropy to $h^{\text{joint}}_{i,t}$. Only after the contribution is fixed it can be published, preventing it from being chosen as a function of the seed. Hence, nodes without a local QRNG are restricted to local operations, unless supplied with independent seed material, which is outside the scope of this work. We define $Z_{i,t} = \text{Ext}_{L_{i,t}}\left(W_{i,t}; \chi_{i,t}\right)$.

*Remark 3 (Security Concerns).* For a selected extraction error $\varepsilon$, the output length is chosen according to the security bound of the instantiated extractor. For standard two-universal hashing, the leftover-hash lemma bound [19] gives an output length of approximately $L_{i,t} \leq h_{i,t}^{\text{joint}} - 2\log_2 \frac{1}{\varepsilon}$, up to the exact statistical-distance convention used by the selected instantiation. Therefore, the extracted string $Z_{i,t}$ is $\varepsilon$-close to an $L_{i,t}$-bit uniform string under the assumptions associated with the entropy bound defined in (10).

The output is partitioned according to the recipient set, *i.e.*

$$Z_{i,t} = R^{\text{out}}_{i\to j_1,t} \parallel \cdots \parallel R^{\text{out}}_{i\to j_d,t}, \tag{16}$$

where $|R^{\text{out}}_{i\to j_k,t}| = \ell^{\text{out}}_{i\to j_k,t}$.

For each recipient, $N_i$ creates a fresh capsule carrying $R^{\text{out}}_{i\to j_k,t}$. The capsule metadata describe the current contributor and are derived locally from the extractor guarantee and local policy. The pair $(W_{i,t}, \omega_{i,t})$ is single-use and erased after processing. Any new randomness claim requires fresh execution and input, with reuse prevented by epoch sequencing and distribution logic.

### 3.1.2 Proof-of-Concept Architectures

Within the entropy distribution layer, the connection between two nodes represents the transfer of an $\mathcal{EC}$. All nodes can be distributed according to different network topologies. Different scenarios may require different compromises between implementation complexity, source diversity, availability, and resistance to the compromise of individual QRNGs or intermediate nodes. We consider three representative configurations, seen as building blocks for more complex scenarios.

*Hierarchical Architecture.* The first architecture corresponds to a hierarchical distribution model with one high-assurance root and several lower-layer QRNGs, as can be seen in Figure 2. As a reference, we consider one DI node, two semi-DI nodes, and two trusted nodes below each semi-DI node, but the topology can naturally be extended[12]. For each level, the local contribution is obtained according to (3). The remote contribution and context are obtained directly from the received $\mathcal{EC}$. The root node executes Setup using fresh local entropy and no incoming capsule. When it provides randomness to $S_A$ and $S_B$, it invokes Contribute once for the recipient set $\{S_A, S_B\}$. Each child node receives its own capsule, generates fresh local entropy, and creates a new Accumulator object. The same mechanism is then applied when the child nodes provide randomness to the trusted nodes below them.

*Remark 4 (Limitation).* The hierarchical architecture is the simplest topology in terms of entropy propagation and context management. Its main limitation is the existence of a single point of failure for each node. For example, the malfunction or compromise of the DI node removes its contribution to both of the SI branches.

[12] a larger number of children nodes or additional levels

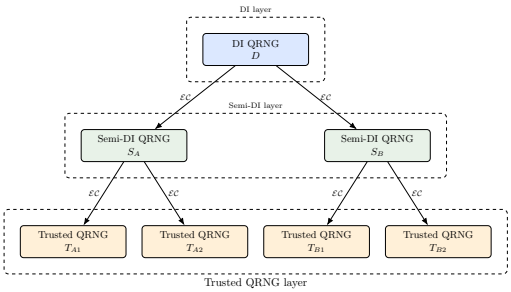



Fig. 2: Hierarchical entropy-distribution architecture

*Multi-root Architecture.* To overcome this natural disadvantage of the hierarchical topology, we introduce the second type of network architecture, which contains local entropy pools that aggregate $\mathcal{EC}$s from nodes with similar assurance level before the distribution to lower-level nodes. We illustrate this topology in Figure 3. The accepted payloads are framed into $W_{S_P,t}$ and their entropy is accounted according to (10). Once the policy is satisfied, Contribute performs one extraction and partitions the resulting string into two outputs. The pool remains unpredictable as long as at least one available source contributes sufficient entropy. After receiving its contribution from the pool, each semi-DI node adds its local entropy, generating the local contribution accordingly.

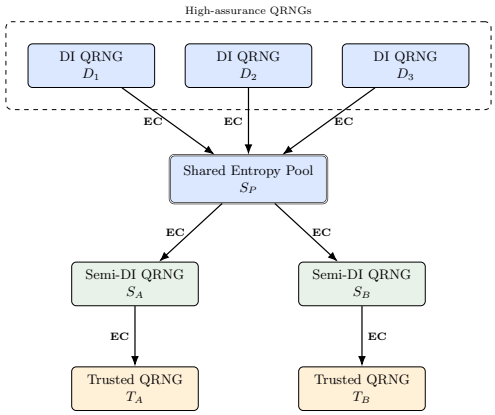



Fig. 3: Multi-root architecture based on a shared entropy pool

*Remark 5 (Diversification).* This architecture provides a two-fold diversification. First, the shared pool aggregates entropy from several upper-level nodes. In addition, the compromise of one upper-level QRNG does not automatically reveal

the complete pool state. Similarly, intercepting the output intended for one recipient does not necessarily reveal the contributions sent to other nodes.

*Remark 6 (Limitation).* A disadvantage is represented by the possibility of pool state $S_P$ being compromised. In this case, all lower-level nodes whose remote contributions are generated from this pool can be affected.

*Federated Architecture.* The third type of architecture considers the interconnection between independently operated entropy distribution networks. Here, each network operates on its own hierarchy, while some selected nodes can establish entropy-distribution links with equivalent nodes from another network. Hence, there is no need for a common global randomness authority or a shared entropy pool. The randomness exchange can take place at different levels, depending on the deployment scenario and the requirements of each network. We provide such an example in Figure 4. The additional contribution increases the diversity of entropy available at each receiving node. In particular, the two networks may use different implementations of QRNG's at equivalent levels in the topology and different network policies.

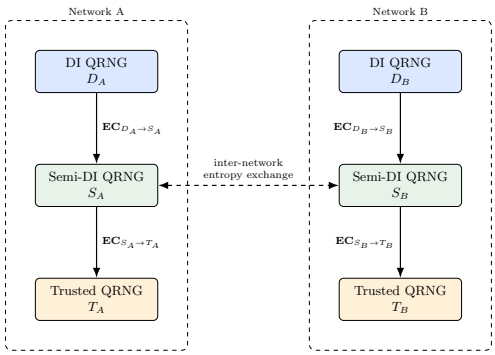



Fig. 4: Federated entropy-distribution architecture

*Remark 7 (Availability).* A failure affecting one network does not affect the entropy generated by the other network. In the case of entropy failures, the peer node from the normal operating network can discard the other contribution.

*Remark 8 (Limitation).* The federated architecture nevertheless introduces a dependency that does not appear in a strict hierarchy.

### 3.2 The Authentication Layer

The nodes in the entropy distribution layer must be authenticated on the network and establish secure communication channels to exchange $\mathcal{EC}$s securely. These

processes are handled within the authentication layer, which may be instantiated using different quantum-resistant cryptographic algorithms. A classical method implies the usage of hybrid certificates and digital signature schemes that use standardized post-quantum algorithms. Another variant, which might provide larger security margins (against quantum adversaries), is to use symmetric-key based authenticated key exchange protocols [3]. Consequently, in Section 4, we present implementation performance metrics of a proof-of-concept randomness distribution network, in which the cryptographic backbone is represented by specific (optimal) instantiations of the protocols in [3].

### 3.3 The Secure-Transport Layer

Once the identities of the communicating nodes have been established, secure-transport layer serves for protecting the information exchanged through the network, encapsulated in the $\mathcal{EC}$s. In particular, the transport mechanism should provide confidentiality and message authenticity, preventing an adversary from modifying, injecting, or replaying valid entropy contributions. This layer can be instantiated using different cryptographic mechanisms.

As our proposed network architecture is modular, we consider the current layer outside the scope of our paper.

## 4 Implementing Symmetric-Based AKE Protocols

For benchmarking purposes, we compare our symmetric-key framework with hybrid post-quantum cryptographic schemes (lattice-based).

As the protocols introduced in [3] are closely aligned to the scope of our work, we recall the general case[13] in Appendix E and consider an instantiation with HMAC-SHA-256/SHA-256. In our implementation, we fix $u = 1$ and consider $(r = 8, \ell = 1024)$ as parameters for protocol $QS_1$, while $(r = 10, \ell = 819)$ correspond to protocol $QS_2$. The hybrid post-quantum scheme ($PQ$) we selected for a fair comparison combines ML-DSA-87/Ed448 (for authentication) and ML-KEM-1024/X448 (for key establishment).

The symmetric-key based approach concentrates provisioning capabilities at the CA, while all other nodes require only lightweight operations, retaining a small amount of secret-key material. In this way, much faster handshakes are achieved (see Table 1), as opposed to hybrid-PQC approaches, in which every node must supports costly operations and stores larger keys and certificates.

*Peer-to-Peer.* Costs are divided into CA setup, user enrollment and connection establishment. We report median CA setup time ($C$), mean of enrollment medians ($E$), mean of per-party handshake medians ($H$), serialized handshake size ($W$), as well as raw user and CA storage ($M_U, M_C$). The means are computed from independent medians, and storage excludes implementation overhead.

[13] $r \geq 1$ and $u \geq 1$

Table 1: Authentication layer metrics

| **Peer-to-peer** | | | | | | |
|---|---|---|---|---|---|---|
| **Scheme** | $C$ **(ms)** | $E$ **(ms)** | $H$ **(ms)** | $W$ **(bytes)** | $M_U$ **(KiB)** | $M_C$ **(KiB)** |
| $QS_1$ | 7.52 | 6.37 | 0.0280 | 302 | 256 | 256 |
| $QS_2$ | 7.85 | 6.78 | 0.0358 | 334 | 256 | 256 |
| $PQ$ | 1.35 | 4.07 | 7.294 | 28016 | 17.3 | 7.42 |

| **Hierarchical deployment** | | | | | | |
|---|---|---|---|---|---|---|
| **Scheme** | $C_\Sigma$ **(ms)** | $E_\Sigma$ **(ms)** | $M_\Sigma$ **(MiB)** | $M_D$ **(MiB)** | $H_T$ **(ms)** | $H_M$ **(ms)** |
| $QS_1$ | 7.50 | 46.3 | 2.00 | 0.500 | 0.064 | 0.0644 |
| $QS_2$ | 7.95 | 47.4 | 2.00 | 0.500 | 0.0743 | 0.0743 |
| $PQ$ | 1.36 | 24.6 | 0.109 | 0.00727 | 14.3 | 14.1 |

| **Estimated connection-establishment latency** | | | | | |
|---|---|---|---|---|---|
| **Scheme** | $R$ | $L_0$ **(ms)** | $L_1$ **(ms)** | $L_{10}$ **(ms)** | $L_{50}$ **(ms)** |
| $QS_1$ | 1 | 0.0585 | 1.06 | 10.1 | 50.1 |
| $QS_2$ | 1 | 0.0741 | 1.07 | 10.1 | 50.1 |
| $PQ$ | 2 | 14.7 | 16.7 | 34.7 | 115 |

*Hierarchical Deployment.* We consider a seven-node hierarchy, such as the one in Figure 2. The symmetric scheme uses a common rack set, while the hybrid scheme has the root directly certify the other nodes. We report summed CA-setup and enrollment medians ($C_\Sigma, E_\Sigma$), total and root storage ($M_\Sigma, M_D$), and mean handshake medians over the six tree links ($H_T$) and all 21 node pairs ($H_M$).

*Connection Latency with Propagation Delay.* We refer to connection latency as $L_d = C_H + Rd$, where $C_H$ is the median handshake computation time, $R$ the protocol RTT[14] budget, and $d$ is the assumed network RTT[15]. These analytical estimates isolate computation at $d = 0$ and illustrate propagation effects, excluding congestion, bandwidth-dependent transmission, and computation overlap.

## 5 Conclusions

We introduced a modular framework for distributing certified randomness across QRNG networks with heterogeneous trust levels, our proposed architecture allowing nodes to strengthen their current randomness with external contributions. Entropy capsules carry randomness together with provenance, freshness, health, and entropy-certification data, while local policies determine which contributions are accepted and how much entropy is credited. We reported faster handshakes between nodes in our proof-of-concept architectures.

*Future Work.* We envision at least three research directions: ① providing a rigorous security proof for our stateless accumulator proposal, aligned with the Entropy Accumulation Theorem and related theoretical aspects [12], ② deep diving into analyzing the secure transport layer, comparing QKD-based architectures against PQC solutions and ③ implementing a scalable reference simulator that models QKD links, layered entropy pools, and the end-to-end transport of entropy capsules, enabling comprehensive performance evaluation.

[14] Round-Trip Time
[15] $d \in \{0, 1, 10, 50\}$

## 6 Acknowledgements

This work was supported by RExQTCS: Romanian Excellence Quantum Technologies enhancing Cybersecurity, a grant of the Ministry of Education and Research from Romania - National Authority for Research and UEFISCDI, project number PN-IV-P6-6.1-CoEx-2024-0214, within PNCDI IV.

## References

1. Aumasson, J.P., Meier, W., Phan, R., Henzen, L.: The Hash Function BLAKE. Springer (2014)
2. Bernstein, D.J.: ChaCha, a variant of Salsa20. In: Workshop Record of SASC 2008: The State of the Art of Stream Ciphers. vol. 8, pp. 3–5 (2008), https://cr.yp.to/chacha/chacha-20080120.pdf
3. Beunardeau, M., El Orche, F.E., Maimuţ, D., Naccache, D., Rønne, P.B., Ryan, P.Y.A.: Authenticated Key Distribution: When the Coupon Collector is Your Enemy. In: SecITC 2019. Lecture Notes in Computer Science, vol. 12006, pp. 1–20. Springer (2020)
4. Bian, Y., Yang, J., Jiang, H., Huang, W., Su, Q., Yu, S., Zhang, L., Zhang, Y., Xu, B.: 20 Gbps real-time source-independent quantum random number generator based on a silicon photonic chip. Optics Letters **50**, 1216–1219 (2025)
5. Bierhorst, P., Knill, E., Glancy, S., et al.: Experimentally generated randomness certified by the impossibility of superluminal signals. Nature **556**, 223–226 (2018)
6. Bruynsteen, C., Gehring, T., Lupo, C., Bauwelinck, J., Yin, X.: 100-Gbit/s Integrated Quantum Random Number Generator Based on Vacuum Fluctuations. PRX Quantum **4**, 010330 (2023)
7. Bundesamt für Sicherheit in der Informationstechnik (BSI): Implementation attacks against qkd systems. Tech. rep., Federal Office for Information Security (December 2023)
8. Cheng, J., Liang, S., Qin, J., Li, J., Yan, Z., Jia, X., Xie, C., Peng, K.: Semi-device-independent quantum random number generator with a broadband squeezed state of light. npj Quantum Information **10**, 20 (2024)
9. Choi, K., Manoj, A., Bonneau, J.: SoK: Distributed Randomness Beacons. In: 2023 IEEE Symposium on Security and Privacy (SP). pp. 75–92 (2023)
10. Cizauskas, M., Tebyanian, H., Fox, A.M., Bayer, M., Aßmann, M., Greilich, A.: 33 Gbit/s source-device-independent quantum random number generator based on heterodyne detection with real-time FPGA-integrated extraction. Quantum Science and Technology **11**, 025022 (2026)
11. Du, Y., Hua, X., Zhao, Z., et al.: Source-independent quantum random number generators with integrated silicon photonics. Communications Physics **8**, 9 (2025)
12. Dupuis, F., Fawzi, O., Renner, R.: Entropy accumulation. Communications in Mathematical Physics **379**(3), 867–913 (2020)
13. Ferguson, N., Schneier, B.: Practical Cryptography. John Wiley & Sons (2003)
14. Ferguson, N., Schneier, B., Kohno, T.: Chapter 9: Generating Randomness. In: Cryptography Engineering: Design Principles and Practical Applications. Wiley (2010)
15. French Cybersecurity Agency (ANSSI), Federal Office for Information Security (BSI), Netherlands National Communications Security Agency (NLNCSA), Swedish National Communications Security Authority: Position Paper on Quantum Key Distribution. Tech. rep., ANSSI, BSI, NLNCSA, NCSA (February 2024)

16. Genzini, M., Vigliar, C., Zahidy, M., Tebyanian, H., Gajda, A., Petermann, K., Zimmermann, L., Bacco, D., Da Ros, F.: On-Chip Semi-Device-Independent Quantum Random Number Generator Exploiting Contextuality. PRX Quantum **7**, 033022 (2026)
17. Han, R., Yu, J.: Fair delivery of decentralised randomness beacon. In: Financial Cryptography and Data Security - 27th International Conference, FC 2023, Bol, Brač, Croatia, May 1–5, 2023, Revised Selected Papers, Part I. Lecture Notes in Computer Science, vol. 13913, pp. 218–234. Springer (2023)
18. Herrero-Collantes, M., Garcia-Escartin, J.C.: Quantum random number generators. Reviews of Modern Physics **89**(1) (Feb 2017), http://dx.doi.org/10.1103/RevModPhys.89.015004
19. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC). pp. 12–24 (1989)
20. ITU-T: G.8261/Y.1361: Timing and synchronization aspects in packet networks. Recommendation G.8261/Y.1361, International Telecommunication Union (2024)
21. ITU-T: G.781: Synchronization layer functions for frequency synchronization based on the physical layer. Recommendation G.781, International Telecommunication Union (2026)
22. Jaeger, L.: The Second Quantum Revolution: From Entanglement to Quantum Computing and Other Super-Technologies. Springer (2018)
23. Kavuri, G.A., Palfree, J., Reddy, D.V., et al.: Traceable random numbers from a non-local quantum advantage. Nature **642**, 916–921 (2025)
24. Killmann, W., Schindler, W.: AIS 31: Functionality Classes and Evaluation Methodology for True (Physical) Random Number Generators. Bundesamt für Sicherheit in der Informationstechnik (BSI) (2011), mathematical-Technical Reference
25. Kincaid, P.S., De Marinis, L., Testa, F., Andriolli, N., Contestabile, G.: Source-device-independent monolithically integrated QRNG in a black box with a generation rate in excess of 30 Gbit/s. Optica Quantum **4**, 218–223 (2026)
26. Kulikov, A., Storz, S., Schär, J.D., et al.: Experimental Randomness Amplification. Nature **653**, 1033–1038 (2026)
27. Laurent-Puig, N., Doosti, M., Innocenzi, A., Diamanti, E.: Unconditional Authentication in Quantum Key Distribution via Hybrid Entangled Physical Unclonable Functions (2026), https://arxiv.org/abs/2605.04650
28. Li, L., Zhang, B., Yin, H., et al.: Harnessing measurement-induced quantum uncertainty for ultrafast on-chip source-device-independent randomness certification. Photonics Research **14**, 3326–3337 (2026)
29. Liu, W.Z., Li, M.H., Ragy, S., et al.: Device-independent randomness expansion against quantum side information. Nature Physics **17**, 448–451 (2021)
30. Liu, Y., Zhao, Q., Li, M.H., et al.: Device-independent quantum random-number generation. Nature **562**, 548–551 (2018)
31. Mannalath, V., Mishra, S., Pathak, A.: A comprehensive Review of Quantum Random Number Generators: Concepts, Classification and the Origin of Randomness. Quantum Information Processing **22**(12), 439 (12 2023), https://doi.org/10.1007/s11128-023-04175-y
32. Marangon, D.G., Smith, P.R., Walk, N., et al.: A Fast and Robust Quantum Random Number Generator with a Self-Contained Integrated P;hotonic Randomness Core. Nature Electronics **7**, 396–404 (2024)
33. Mizutani, A., Sasaki, T., Takeoka, M., Kato, G.: Protocol and Security Proof of Decoy-state BB84 Quantum Key Distribution. Technical report, Quantum Key

Distribution Technology Promotion Committee, Quantum Forum (May 2025), https://qforum.org/assets/documents/20250514-PSPD(1.0).pdf
34. National Institute of Standards and Technology: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. Tech. Rep. NIST Special Publication (SP) 800-22 Rev. 1a, National Institute of Standards and Technology, Gaithersburg, MD (2010), https://nist.gov
35. National Institute of Standards and Technology: FIPS PUB 180-4: Secure Hash Standard. NIST (aug 2015)
36. National Institute of Standards and Technology: Recommendation for Random Number Generation Using Deterministic Random Bit Generators. Tech. Rep. NIST Special Publication (SP) 800-90A Rev. 1, Gaithersburg, MD (June 2015), https://csrc.nist.gov/pubs/sp/800/90/a/r1/final
37. National Institute of Standards and Technology: Module-Lattice-Based Digital Signature Standard. Technical Report NIST Federal Information Processing Standards Publication (FIPS) 204, U.S. Department of Commerce (August 2024)
38. National Institute of Standards and Technology: Module-Lattice-Based Key-Encapsulation Mechanism Standard. Technical Report NIST Federal Information Processing Standards Publication (FIPS) 203, U.S. Department of Commerce (August 2024)
39. National Institute of Standards and Technology: Recommendation for Random Bit Generator (RBG) Constructions. NIST Special Publication (SP) 800-90C (September 2025)
40. National Security Agency (NSA): Quantum cryptographic solutions are limited compared to post-quantum algorithms. NSA Cybersecurity Post-Quantum Cryptography Resource Hub (June 2026), https://www.nsa.gov/Cybersecurity/Post-Quantum-Cryptography-Resource-Hub/Article-Display/Article/4616320/quantum-cryptographic-solutions-are-limited-compared-to-post-quantum-algorithms/, accessed: October 2026
41. NATO Digital Policy Committee (DPC): Position Paper on Quantum Key Distribution. Note by the Chairman AC/322-D(2026)0037, North Atlantic Treaty Organization, Brussels, Belgium (August 2026), https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/NATO_Position-QKD.pdf?__blob=publicationFile&v=3, publicly Disclosed
42. Pironio, S., Acín, A., Massar, S., et al.: Random Numbers Certified by Bell's Theorem. Nature **464**, 1021–1024 (2010)
43. Qiao, G.R., Bai, B., Weng, Z.X., et al.: Real-Time Vacuum-State Quantum Random-Number Generator on a Chip. Physical Review Applied **24**, 044031 (2025)
44. Quantum Key Distribution Technology Promotion Committee: Supporting Document – Evaluation Method for Protection Profile for Prepare and Measure Quantum Key Distribution Modules. Technical Report QF-TD-QKD-2025-002_SDv1.0, Quantum Forum (May 2025), https://qforum.org/assets/documents/20250513-SD(1.0).pdf
45. Schindler, P., Judmayer, A., Stifter, N., Weippl, E.: HydRand: Efficient Continuous Distributed Randomness. In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 73–89 (2020). https://doi.org/10.1109/SP40000.2020.00003
46. Shalm, L.K., Zhang, Y., Bienfang, J.C., et al.: Device-independent randomness expansion with entangled photons. Nature Physics **17**, 452–456 (2021)
47. Stanford Emerging Technology Review Working Group: Chapter 07: Quantum Technologies. In: Lin, H.S., Giles, M. (eds.) The Stanford Emerging Technology Review 2026 (2026), https://ro.scribd.com/document/978742347/SETR-2026-Q

uantum-Technologies, co-chaired by Condoleezza Rice, Jennifer Widom, and Amy Zegart
48. Yu, S., Xu, B., Luo, B., et al.: An Ultra-Fast Quantum Random Number Generation Scheme Based on Laser Phase Noise. Communications Physics (2026)
49. Zhang, G., Primaatmaja, I.W., Chen, Y., et al.: Self-Testing Quantum Randomness Expansion on an Integrated Photonic Chip. PRX Quantum **7**, 020353 (2026)
50. Zhang, J., Li, Y., Zhao, M., et al.: One-sided device-independent random number generation through fiber channels. Light: Science & Applications **14**, 25 (2025)
51. Zhang, L., Liu, T., Ou, Z., Kan, H., Zhang, J.: AsyRand: asynchronous distributed randomness beacon with reconfiguration. Cryptology ePrint Archive, Report 2025/406 (2025), https://eprint.iacr.org/2025/406
52. Zhang, Y., Shalm, L.K., Bienfang, J.C., et al.: Experimental Low-Latency Device-Independent Quantum Randomness. Physical Review Letters **124**, 010505 (2020)

## A Literature Review for Device-Dependent QRNGs

A DD QRNG assumes that both the entropy source and measurement apparatus are correctly modeled and calibrated. The output's min-entropy should be obtained on the basis of the mathematical model of the physical process and the measurement device. Modern high-quality DD QRNGs can provide strong conditional min-entropy bounds that explicitly include classical and quantum side information: calibration procedure and uncertainty, environmental limits and online health tests, and the calculated min-entropy that is obtained based on the modeling. The validity of the bound depends on a device model rather than being inferred from a DI witness. Trusted QRNGs can provide meaningful entropy bounds[16]. Therefore, a security-oriented implementation should distinguish between the noise originating from the quantum process and the one coming from the classical processing system. We present a selection of trusted QRNGs implementations in Table 2.

## B Literature Review for Semi-Device-Independent QRNGs

Semi-DI QRNGs offer a compromise between DI protocols[17] and trusted-device protocols[18]. In source-DI QRNGs, the optical source can be untrusted, while the measurement device is trusted and described by a calibrated POVM[19]. In the complementary approach, the measurement device is untrusted, while trusted or constrained quantum states are used to estimate randomness from the observed input-output behavior. This approach can be useful when the detectors are more difficult to characterize than the state preparation process. The latest practical semi-DI results are summarized in Table 3.

[16] only as good as the model and calibration envelope
[17] require less trust in the devices
[18] strongly rely on mathematical models
[19] randomness obtained from the uncertainty of the measurement outcomes

Table 2: Representative DD QRNGs

| Work | Entropy source | Entropy treatment | Reported rate | Architectural significance |
|---|---|---|---|---|
| [6] | Vacuum fluctuations | DD framework, classical and quantum secure & ADC nonlinearity | 100 Gbps | Integrated optoelectronic co-design & high-rate vacuum QRNG |
| [32] | Integrated photonic core | Model-based secure-rate adaptation & long-term monitoring | 2 Gbps secure real time | Eight-board reproducibility & 38-day uncontrolled-environment deployment |
| [43] | Vacuum fluctuations | Device/model calibration with on-device processing | 5.2 Mbps real time | Fully functional compact package & constant output over industrial temperature range |
| [48] | Laser phase noise | Wiener-process phase-noise model; conditional min-entropy analysis | 156 Gbps offline & 10 Gbps real time | 20 GHz entropy bandwidth & explicit distinction between optimized offline and real-time pipelines |

## C Literature Review for Device-Independent QRNGs

In the case of device-independent QRNG, the internal quantum devices are treated as black boxes. The security guarantee of the output random sequences lies in the non-local properties of quantum mechanics. This type of generation mechanism requires the existence of an entanglement source, so that each of the two parties receives one of the two entangled pairs. After local processing, each of the parties will return one classical value. Based on the quantum states and the classical outputs, it can be checked whether the experiment resulted from a quantum mechanical setup by means of checking the Clauser–Horne–Shimony–Holt (CHSH) inequality. Although a rigorous DI QRNG requires more than observing the validation of CHSH in the quantum scenario. Important conditions include detection/fair-sampling loophole, locality/no-signaling loophole (one party cannot causally communicate the choice/outcome to the other during a trial), and measurement independence. We present Table 4 some DI QRNG constructions that were reported in the literature.

## D Description of the Entropy Capsule

1. Version ($v$) indicates the format of $\mathcal{EC}$ (enables backwards compatibility);
2. Source ID ($ID$) identifies the sender of $\mathcal{EC}$;
3. Source class ($Class$) describes the trust model associated with the producing node (DD, DI or semi-DI); this information does not imply that the receiving node inherits the same classification (*e.g.* a semi-DI node receiving a contribution from a DI node remains a semi-DI QRNG node).
4. Epoch ($e$) and sequence number ($q$) describe the position of the capsule within the stream of the received $\mathcal{EC}$s; the epoch can identify a given operating period

Table 3: Representative semi-DI QRNGs

| Work | Trust model | Entropy/certification | Reported result | Significance |
|---|---|---|---|---|
| [8] | Source-DI CV | EUP; $H_{\min}^{\varepsilon} = 7.21 \pm 0.04$ b/sample, conservatively $3.32 \pm 0.04$ after effective-bit-depth correction | 580.7 Mbps private randomness | Explicit treatment of noisy LO, saturation and source distrust |
| [11] | Source-DI DV | Complementary/source monitoring model; 0.21% error | 9.49 Mbps | Integrated discrete-variable source-DI implementation |
| [4] | Source-DI CV | Source-independent security + measurement model | 20 Gbps real time; 28 Gbps extractor throughput | High-rate real-time chip-based pipeline |
| [50] | 1SDI CV | $H_{\min} = 0.07057$ bits/sample at 2 km | 7.06 Mbps at 2 km | Network-native asymmetric trust model |
| [10] | Source-DI CV | POVM/measurement-uncertainty entropy bound; quantum side information | 33.92 Gbps sustained real time | Commodity-style high-bandwidth electronics + real-time extraction |
| [25] | Source-DI CV | Trusted measurement, untrusted source | 35 Gbps | First monolithic source-DI black-box package with electrical-only I/O |
| [28] | Source-DI | Finite-size corrected $H_{\min}^{\varepsilon} = 3.645$ b/sample | 583.2 Gbps reported secure aggregate rate | Demonstrates scale-out by parallel integrated channels |
| [49] | MDI prepare-and-measure | Quantum-side-information security; behaviour-based receiver certification | 15.33 Mb/run; expansion $5.11 \times 10^{-4}$ at 10 MHz | Self-testing detector on foundry-compatible silicon photonics |
| [16] | Contextuality-based semi-DI | KCBS violation; $H_{\min} = 0.077 \pm 0.002$ bits/round | $21.7 \pm 0.5$ bps asymptotic | On-chip certification from contextuality without entanglement |

or reseeding interval, while the sequence number provides an ordering of the capsules generated during that period; these values allow the receiving node to detect old or duplicated contributions and provide a simple mechanism against replay at the randomness plane;

5. Generation time ($\tau$) records when the contribution was created; together with `freshness_limit` ($\Delta$), it allows the receiving node to determine whether the contribution is sufficiently recent to be included in the current entropy state;
6. Random payload ($R$) contains the actual random contribution transported between two layers; this value should normally correspond to conditioned or extracted randomness rather than directly to raw measurements of the physical entropy source; raw samples are specific to the stochastic model and implementation of the QRNG and are not required by the lower layer;
7. Payload length ($l$) indicates the size of the transported contribution and provides the necessary context for interpreting the entropy claim; for a payload $R$ of $n$ bits, the capsule may state a lower bound $H_{\min}(R|E) \geq h$, where $h$ is represented by `claimed_min_entropy` ($h^R$);
8. Claimed min entropy ($h^R$) should be interpreted as the entropy guaranteed by the producing node under its corresponding trust model (lower layer may apply its own policy before accepting this value);

Table 4: Selected milestones in rigorous DI quantum randomness

| Work | Model | Security | Time/rate | Advance |
|---|---|---|---|---|
| [42] | Bell-theorem randomness certification | Tens of certified bits | Long acquisition | First experimental Bell-certified randomness |
| [5] | Loophole-free photonic Bell & no-superluminal-signalling model | 1024 bits & within $10^{-12}$ of uniform | Order of minutes (experimental) | Strong operational certification from loophole-free correlations |
| [30] | Loophole-free & secure against quantum and classical adversaries | $6.2469 \times 10^7$ b & total failure $< 10^{-5}$ | 96 h ($\sim$ 181 bps) | Large quantum-proof DI output |
| [52] | Quantum-proof loophole-free protocol | 512-bit blocks & error lower than $2^{-64}$ | Lower than 5 min average/block | Low-latency DI randomness |
| [29] | Entropy accumulation & quantum side information | Net gain $2.57 \times 10^8$ b & soundness $3.09 \times 10^{-12}$ | 19.2 h & 13,527 bps | DI randomness expansion |
| [46] | Spot checking & classical side information | 1.181B output vs 0.953B input & 24% expansion | 91 h & 3606 bps | Independent expansion experiment |
| [23] | DI nonlocality & distributed hash-chain traceability | 512-bit beacon pulse; error$\times$success $\leq 2^{-64}$ | 99.7% success over first 40 days | Auditable, traceable public certified randomness |
| [26] | Loophole-free Bell test with superconducting circuits | DI amplification of weak randomness | Task-oriented rather than rate record | First experimental DI randomness amplification |

9. Health state ($\mathcal{H}$) indicates whether the QRNG considered itself to be operating inside its accepted conditions when the capsule was generated; a capsule generated while the source is outside the valid operating state can therefore be discarded without interpreting the random payload.

## E Authenticated Key Exchange Protocols

Each user receives $u$ distinct keys per rack. The function $I$ is characterized by three indices: ① $i$ denoting the concerned user, ② $\rho$ denoting the rack and ③ $\mu$ an index running from 1 to $u$.

More precisely, $k^{\rho}_{I(i,\mu,\rho)}$ indicates that the $\mu$-th key from the rack $\rho$ is given to the user $i$. Note that $k_{I(i)}$ defined in the previous section just corresponds to $k^1_{I(i,1,1)}$.

*Key Generation:* CA generates $r$ racks of $\ell$ distinct keys: $R_\rho = \{k^\rho_1, \ldots, k^\rho_\ell\}$, where $\rho \in [1, r]$.

*User Enrolment.* CA gives to user $\mathrm{ID}_i$:

– $u \times r$ secret keys:

$$\begin{array}{cccc} k^1_{I(i,1,1)} & k^2_{I(i,1,2)} & \cdots & k^r_{I(i,1,r)} \\ k^1_{I(i,2,1)} & k^2_{I(i,2,2)} & \cdots & k^r_{I(i,2,r)} \\ \vdots & \vdots & & \vdots \\ k^1_{I(i,u,1)} & k^2_{I(i,u,2)} & & k^r_{I(i,u,r)} \end{array}$$

where $\forall \rho \in [1,r], \forall \mu \in [1,u], I(i,\mu,\rho) \xleftarrow{\$} [1,\ell]$

– A table $T_i$ of $\ell \times r$ derived keys:

$$T_i = \begin{bmatrix} t^1_{i,1} & t^2_{i,1} & \cdots & t^r_{i,1} \\ t^1_{i,2} & t^2_{i,2} & \cdots & t^r_{i,2} \\ \vdots & \vdots & & \vdots \\ t^1_{i,\ell} & t^2_{i,\ell} & \cdots & t^r_{i,\ell} \end{bmatrix}$$

where $\forall \rho \in [1,r], \forall j \in [1,\ell], t^\rho_{i,j} = f(k^\rho_j, \mathrm{ID}_i)$

*Remark 9.* Note that the user can derive the table values for his own keys and in principle does not need to store these. In this way memory can be saved at the cost of computational efficiency during key derivation.

*Key Exchange:* Assume now that users $\mathrm{ID}_i$ and $\mathrm{ID}_j$ want to establish a secure communication channel. To generate their common secret key, they do the following:

1. Exchange their indices $I(i,\mu,\rho)$ and $I(j,\mu,\rho)$ for $\mu \in [1,u], \rho \in [1,r]$;
2. User $\mathrm{ID}_i$:
   – generates $u \times r$ derived keys:
   $$t^\rho_{j,I(i,\mu,\rho)} = f(k^\rho_{I(i,\mu,\rho)}, \mathrm{ID}_j), \quad \forall \mu \in [1,u], \quad \forall \rho \in [1,r]$$
   – reads $u \times r$ derived keys from his table $T_i$:
   $$t^\rho_{i,I(j,\mu,\rho)} = f(k^\rho_{I(j,\mu,\rho)}, \mathrm{ID}_i), \quad \forall \mu \in [1,u], \quad \forall \rho \in [1,r]$$
3. User $\mathrm{ID}_j$:
   – generates $u \times r$ derived keys:
   $$t^\rho_{i,I(j,\mu,\rho)} = f(k^\rho_{I(j,\mu,\rho)}, \mathrm{ID}_i) \quad \forall \mu \in [1,u], \quad \forall \rho \in [1,r]$$
   – reads $u \times r$ derived keys from his table $T_j$:
   $$t^\rho_{j,I(i,\mu,\rho)} = f(k^\rho_{I(i,\mu,\rho)}, \mathrm{ID}_j) \quad \forall \mu \in [1,u], \quad \forall \rho \in [1,r]$$
4. Both users $\mathrm{ID}_i$ and $\mathrm{ID}_j$ generate a common session keys by using $h$ to combine the $2u \times r$ derived keys:
   $$\mathsf{sk} = h\big(t^\rho_{i,I(j,1,1)}, \ldots, t^\rho_{i,I(j,u,r)}, t^\rho_{j,I(i,1,1)}, \ldots, t^\rho_{j,I(i,u,r)}\big).$$