scientific reports

Check for updates

OPEN

Quantum secure image encryption using hybrid QTRNG and QPRNG

T. S. Gururaja & Padmapriya Pravinkumar✉

Secure image transmission has become critical in the emerging quantum computing landscape. This research introduces a hybrid key quantum encryption approach that combines Quantum True Random Number Generation (QTRNG) and Quantum Pseudo Random Number Generation (QPRNG) to enhance security and efficiency. The proposed Quantum Hybrid Random Number Generator (QHRNG) integrates entanglement-based true randomness with structured pseudo-randomness using Hadamard, controlled rotation and Clifford gates to generate high-entropy encryption keys. The generated Quantum keys are applied to encrypt grayscale images using the Novel Enhanced Quantum Image Representation (NEQR). The encryption process is strengthened by quantum bit-level scrambling and selective Quantum Fourier Transform. The proposed method is implemented on Qiskit's Aer and Basic simulators and validated on IBM's torino quantum hardware. Compared to standalone QTRNG and QPRNG methods, QHRNG consistently provides higher randomness and stronger security in noisy and noiseless environments. This work offers a robust and future-ready framework for quantum secure image encryption.

Keywords Secure image encryption, QTRNG, QPRNG, QHRNG, Hybrid key

In recent years, thanks to the rapid growth in Quantum technology. Quantum image encryption offers a revolutionary method for safeguarding multimedia information by harnessing quantum mechanical principles such as superposition, entanglement, and the no-cloning theorem. These quantum properties create a unique and powerful framework for protecting sensitive image data against emerging security risks[1]. The development of quantum cryptography is one of the threats to multimedia communication. The increased computational capacity of quantum computing may render multimedia protocols and applications insecure. Shihua Zhou[2] introduced a DNA-controlled NOT operations, where natural DNA sequences and chaotic maps generate encryption coordinates that scramble quantum images at the bit-plane level. The method demonstrates strong performance regarding key space security, making it suitable for real-time encryption and cloud transmission. This encryption may still depend on classical pseudorandom sequences, which are less secure than proposed quantum randomness.

Jingbo Zhao et al.[3] proposed an alternate quantum walk (AQW) and the controlled Rubik's Cube operation to secure colour image data. This approach integrates quantum computational processes with permutation techniques inspired by the Rubik's Cube structure. The security of this method relies heavily on the randomness generated by quantum walks and controlled operations. These sources might not provide the same level of unpredictability as QTRNG, leading to potential vulnerabilities. If the parameters governing the quantum walk or Rubik's Cube operations are compromised, the encryption can be reverse engineered, reducing its robustness against cryptanalysis. Quantum Fourier Transform (QFT) and quantum bit-level operations in our proposed method offer a stronger defence against cryptographic attacks than the parameterized operations of the Rubik's Cube.

Xingbin Liu et al.[4] proposed an intra-inter-bit level permutation method to encrypt images. They used the logistic chaotic map to create a pseudo-random key sequence XORed with NEQR quantum operation. The logistic map is inherently pseudorandom, which means its randomness relies on deterministic processes. The encryption algorithm is vulnerable to attacks if the initial conditions are known. Small approximations can lead to predictable outputs, compromising the encryption's robustness. Quantum Fourier Transform, and bit-level scrambling offer superior confusion and diffusion compared to logistic map-based permutations. Xinsheng Li et al.[5] proposed a quantum chaotic system, Sparse Bayesian Learning (SBL) and a 3D Arnold cat map for joint compression and encryption. This algorithm was computationally intensive for high-resolution images, and its complexity decreases its scalability for real-time multimedia applications. Our proposed hybrid approach can adapt to various multimedia formats, providing a versatile solution.

Arslan Shafique et al.[6] combine QKD and chaotic sequences using the 6D Chen system Ikeda map for key generation and scrambling. This approach is robust against various cyberattacks (brute-force, clipping, noise)

SASTRA Deemed to be University, Thanjavur, Tamilnadu, India. ✉email: padmapriya@ece.sastra.edu

but partially depends on classical cryptography. Our method uses purely quantum methods like QTRNG and QPRNG for key generation, ensuring higher randomness and quantum-level security. Quantum-secure encryption resists classical, and quantum attacks due to inherent quantum randomness and strong scrambling. The quantum logistic map enhances randomness, while the public key cryptosystem ensures secure key exchange between users. It has parameter sensitivity that may lead to predictable patterns if attackers do not carefully choose or identify the parameters. Chaotic systems are effective against classical attacks; public key cryptography like RSA and elliptic curve cryptography are vulnerable to quantum attacks. Using quantum logistic maps[7] introduces potential instability due to sensitivity to initial conditions. This unpredictability might cause difficulties in ensuring consistent and reliable performance when applied to real-world applications.

QHF-based systems[8] are more vulnerable to quantum noise and decoherence, leading to potential errors in derived keys or hashes. The proposed encryption method directly secures image data by employing bit-level and pixel-level operations and robust error handling by simulating noisy and noiseless environments, ensuring high resilience against quantum noise. Perovskite materials[9] used in light-emitting diodes are sensitive to environmental factors like temperature, humidity and light exposure. This can impact the stability and reliability of the random numbers generated. The randomness quality of PeLED-based QRNG heavily depends on the efficiency and operational characteristics of the physical hardware, which can degrade over time. The proposed encryption framework utilizes quantum randomness generated through simulated quantum circuits, which are less affected by hardware limitations.

Quantum Random Walk (QRW)[10] based PRNG relies heavily on quantum coherence, making it vulnerable to noisy environments. Computational overhead may increase due to the complexity of quantum walks. The proposed scheme employs simpler quantum circuits such as Hadamard and CRz gates to ensure efficient randomness generation with minimal computational costs. The Continuous Chaotic Map-based scheme[11] randomness is derived from deterministic chaotic equations that are inherently pseudo-random, and it is sensitive to noise parameters that can lead to decryption errors. Baker Map and 2D Logistic Map Scheme[12] rely heavily on the properties of classical chaos as a deterministic mathematical model. Its pseudo-random properties exhibit periodicity and predictability under certain conditions. The encryption performance may degrade under noise due to the sensitivity of chaotic systems.

Quantum logistic maps[13] provide a pseudo-random mechanism, but its randomness is not inherently superior to the proposed quantum key. SHA-3 is resilient against pre-image attacks. Grover's algorithm can reduce the complexity of brute-force attacks on hash functions, potentially weakening its security when faced with quantum adversaries. Wen-Wen Hu et al.[14] utilises a two-stage process: a generalised Arnold transform for scrambling and pixel-level encryption using a quantum key image derived from a chaotic logistic map. The three-level quantum image encryption algorithm integrates block-level permutation, bit-level scrambling, and pixel-level diffusion using Quantum Arnold Transform and logistic map-based chaos. This scheme has high complexity and scalability issues due to inefficient multi-layer encryption operations for large-scale encryption[15].

QRNG[16], leveraging quantum mechanical principles offer a promising solution by producing truly random sequences[32–34]. QRNGs are typically more expensive and complex than traditional pseudorandom number generators, requiring specialized quantum hardware and environmental control to maintain accuracy. The computational indistinguishability from Haar random states[17] might require careful calibration and precision control of the quantum circuits, which can be difficult to achieve in practice due to hardware imperfections. In the IBM quantum lab[18–22], QTRNG uses the Random Quantum Fourier Transform (RQFT)[19] technique and Hadamard gate to produce high-quality, unpredictable random numbers for cryptographic applications. Unlike FRQI, NEQR[20] stores pixel gray-scale values using the basis state of a qubit sequence, allowing for the clearer distinction of gray scales. NEQR achieves a quadratic speedup in image preparation. The testing and validation of true random and pseudorandom number generators for cryptographic applications focusing on the requirements and standards set by NIST's SP 800 – 90[35] and 22 series to ensure secure and unpredictable randomness[23,24].

Xiaopeng Yan et al.[26] introduce a fractional-order wavelet decomposition and Arnold transform to enhance image security through iterative key matrix generation and diffusion. Wentao Hao et al.[27] proposed two-dimensional quantum walks utilizing quantum entanglement for pseudorandom number generation and the NEQR model for enhanced security. The sine logistic map[28] itself is a classical chaotic system. This classical key generation method is a significant weakness because it does not fully leverage the security potential of quantum mechanics. A hybrid quantum image encryption algorithm[29] combines DNA encoding with QTRNG[34] and RQFT to enhance the security of image encryption. Chebyshev maps[30] rely on classical key generation methods, making them vulnerable to potential quantum attacks.

This study is motivated by the critical need to overcome the limitations of existing classical chaotic key generation methods by developing a quantum-secure key generation framework that ensures high entropy and robust encryption for quantum-secure image encryption. In this proposed algorithm, the generated quantum true random sequences are non-deterministic and unbiased compared to the probabilistic nature of quantum mechanics. The scalability of quantum circuits can be easily extended to larger n-qubit systems to produce longer random sequences. Entanglement introduced by controlled rotation gates maximizes the quantum state's entropy and ensures high-quality randomness. True randomness generated at the quantum level resists manipulating and predicting the quantum random number. This QTRNG provides the secure key for quantum and classical cryptographic applications[31]. In Monte-Carlo Simulation, high-quality randomness is crucial for accurate simulations in finance, physics, and engineering. Quantum pseudo-random sequences are highly entangled but retain deterministic control, making them computationally efficient and preserving randomness. The deterministic nature of controlled randomness allows reproducibility under specific quantum operations. Maintaining the security of the quantum system, fast quantum pseudo randomness offers computational efficiency. The scalability of a modular quantum circuit can handle large keys for high-resolution images.

Quantum Hybrid random sequences result in a dual nature of randomness that ensures the resistance of both classical and quantum attacks.

The novelty of this proposed approach:

- The proposed Quantum True-Pseudo Hybrid method is the first approach to integrate true quantum randomness (QTRNG) with pseudo-randomness (QPRNG) in a unified framework.
- A novel use of Clifford gates to transform true random seeds into high-dimensional entangled quantum states.
- Quantum hybrid random number generation emerges as the most reliable generator achieving high p-values across nearly all tests and simulation environments. This hybrid approach mitigates the adverse effects of noise and variability in quantum simulations.
- The proposed quantum image encryption utilizes the quantum nature of keys, which makes them non-clonable due to the no-cloning theorem.
- The proposed quantum random numbers create an entanglement and phase transformations ensure the pixel values are highly randomized makes the encryption robust against cryptanalysis.
- The selective Quantum Fourier Transform introduces global quantum correlations and interference, adding a layer of quantum complexity to the encryption.

The proposed quantum secure key generation framework

Generation of quantum true random number

The proposed quantum circuit optimizes the output randomness and unpredictability through a selected combination of Hadamard with a controlled rotation gate.

$$|\psi\rangle = |H\rangle^{\otimes n} |0\rangle^{\otimes n} \tag{1}$$

where $|\psi\rangle$is a quantum states and $|0\rangle^{\otimes n}$ is zero initialization in the quantum register.$|H\rangle^{\otimes n}$represents here the application of the Hadamard gate into all the qubits. The Hadamard gate creating a superposition state for each qubit, this ensures the probability amplitude of $|0\rangle$ and $|1\rangle$ are equal.

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \tag{2}$$

Equation (2) represents a superposition of all possible of n bit basis states. It ensures that total probability of measuring any one of the 2^n states. The non-symmetric controlled rotation $CRZ\,(\theta)$ gate entangles the qubit and correlation between their states that cannot be replicated classically.

$$CRZ\left(\frac{\pi}{2}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\frac{\pi}{2}} \end{bmatrix} \tag{3}$$

The controlled rotation gate introduces a phase rotation parameter ($\pi/2$) on the target qubits and controls the other qubits. It adds an additional layer of phase randomness. The consecutive qubits (q_i, q_{i+1}) apply controlled RZ gate with angle ($\pi/2$). After applying the control rotation gate, the generalized quantum state is represented as

$$|\psi\rangle = \prod_{i=0}^{n=2} CRZ\,(\theta)_{i,i+1} |H\rangle^{\otimes n} |0\rangle^{\otimes n} \tag{4}$$

where Π is product notation represents a series of controlled rotation gates between adjacent qubit pairs ensures cascading entanglement. Whenever measuring the quantum state $|\psi\rangle$ the measurement outcome probability of $|\langle x||\psi\rangle|^2$ computational basis $\{|x\rangle\}$ were $x \in \{0,1\}^n$. Stimulate the quantum circuit for the given shots repetitions, collecting the frequency $f\,(x)$ for each outcome of x. For each bitstring x with the frequency.

$$f\,(x):\ S_1 = S_1 + x^{f(x)} \tag{5}$$

where $f\,(x)$ times bitstrings are concatenated to get a QTRNG sequence and 'S_1' is an accumulative expression for QTRNG outcome. It is incremented in each shot for each value of x. This quantum circuit creates a superposition and entanglement with an equiprobable state, satisfies the quantum principle, and generates a quantum true random number. This proposed quantum circuit balances simplicity and complexity. It has low depth, making it feasible to implement on quantum hardware. It achieves a high level of entropy through entanglement and phase manipulation. This QTRNG enhances the randomness quality by combining local and global phenomena.

$$|0\rangle^{\otimes n} \xrightarrow{|H\rangle^{\otimes n}} \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle \xrightarrow{CRZ} \textit{Entangled state} \xrightarrow{} \textit{Classical outcome} \tag{6}$$

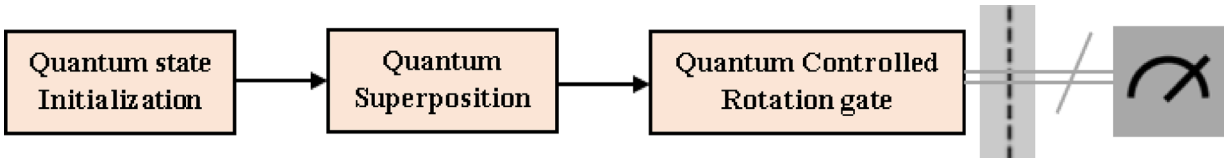


Fig. 1. Conceptual flow diagram of proposed QTRNG.

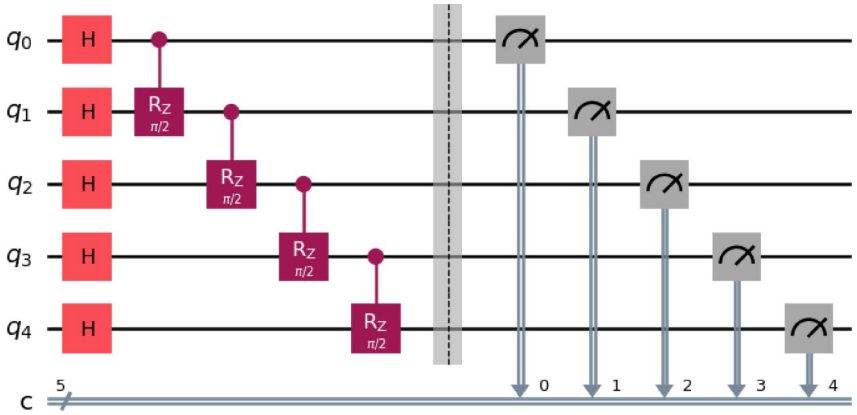


Fig. 2. Software implementation of proposed QTRNG (5 qubit circuit).

Quantum gate	Parameter	Purpose	Explanation
Hadamard gate	Rotation around Y-axis $\left(\frac{\pi}{2}\right)$ and X-axis (π)	Creates equal superposition of basis states	It prepares qubits in a superposed state and enabling quantum parallelism.

Table 1. Proposed QTRNG component with parameter.

Quantum gate	Parameter	Purpose	Explanation
Controlled Rotation Z gate	$\left(\frac{\pi}{2}\right)$	Applies conditional phase rotation (entanglement)	It introduces controlled entanglement and ensures a stable quantum interference

proposed QTRNG Circuit equation,

The quantum circuit illustrates preparing and measuring a quantum state through a sequence of operations. It begins with an initial state of $\{|0\rangle\}$, where all n qubits are initialized to the basis state $\{|H\rangle\}$. Hadamard gates $|H\rangle^{\otimes n}$ are applied to each qubit, transforming the circuit into an equal superposition of all possible 2^n basis states. This results in the quantum state $\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle$ enabling quantum parallelism. A controlled rotation Z gate is applied, introducing entanglement among the qubits by rotating around the Z-axis based on the state of control qubits. This step transforms the separable superposition into an entangled quantum state, a key feature in quantum information processing. Finally, the qubits are measured, collapsing the entangled state into a classical outcome. Finally, a measurement is performed on the entangled state, yielding classical results that reflect the encoded quantum information.

Figure 1 illustrates the quantum process flow from initialization and superposition to entanglement using controlled rotation gate operation followed by measurement for quantum true randomness.

The QTRNG circuit produces intrinsic probabilistic nature of quantum mechanics to generate truly random bits. The approach begins with qubit initialization $|0\rangle$ followed by the application of Hadamard gate to place them into equal superposition of $|0\rangle$ and $|1\rangle$. Upon measurement, this ensures each bit has an equal and unpredictable chance of collapsing into either state. To further strengthen randomness and inter qubit unpredictability in multi qubit system, a controlled rotation CRZ gate is introduced to entangle qubits. It adds quantum correlations that classical systems cannot replicate. The final measurement extracts truly random bits from superposed and entangled state. The overall conceptual flow of QTRNG focuses on minimizing determinism and maximizing entropy by relying entirely on quantum state collapse with no classical or deterministic processing involved.

Figure 2 proposed a quantum circuit for true random number generation, adding an equal number of quantum and classical registers in every qubit. Initialize $|0\rangle$ state and Hadamard gate in all qubits for superposition. Apply

Input: n – Number of qubits, θ_z – Rotation angle, $shots$ – Number of shots for stimulation runs
Output: S_1 − Concatenated bitstrings based on measurement frequency
Require: $backend \leftarrow$ BasicProvider.get_backend('basic_simulator')
$backend \leftarrow$ Aer.get_backend('Aer_simulator')
while $n > 0$ **do**
Initialize $qr \leftarrow$ QuantumRegister, $cr \leftarrow$ ClassicalRegister, $ckt \leftarrow$ QuantumCircuit of n size
for i in range(n) **do**
$ckt.h(qr[i])$
end for
for i in range$(n-1)$ **do**
$ckt.crz(\theta, qr[i], qr[i+1])$
end for
$ckt.barrier()$
$ckt.measure(qr, cr)$
$transpile_{circuit} \leftarrow transpile(ckt, backend)$
$result \leftarrow backend.run(transpiled_{circuit}, shots = shots).result()$
$counts \leftarrow result.get\_counts()$
$S_1 = \oplus_{x \in counts}\ x^{f(x)}$
end while
return S_1

Algorithm 1 . Quantum entangled true random number generation.

a controlled rotation gate for an entanglement. Finally, each qubit of the quantum circuit is measured to produce a quantum true random number.

Table 1 presents the quantum gates and their parameters in the proposed QTRNG to achieve superposition and phase entanglement which are crucial for generating true quantum randomness

.

This proposed QTRNG algorithm combines a quantum superposition gate and a measurement technique to generate a quantum true random number. The quantum circuit is constructed with the quantum register and classical register initialization. The Basic simulator[18] can run 24 qubits alone; this proposed circuit contains 24 quantum and classical registers. The Hadamard gate is applied to each qubit in the quantum register to produce a superposition state. It distributes the probability amplitude evenly across all possible states to create equal superposition. A series of controlled CRZ gates are applied sequentially between adjacent qubits. These controlled rotation gate introduce correlations between the qubits by using the phase shift of $\left(\frac{\pi}{2}\right)$ to the target qubit only if the control qubit is in $|1\rangle$ state. A barrier is applied to the circuit to visually separate the preparation phase from the measurement stage to enhance the quantum circuit's readability. While measuring each qubit of the quantum register, the state collapses to superposition into a definite state with the outcomes recorded in the classical register.

The transpiled quantum circuit is optimized it for the specific backend to ensure compatibility with quantum gate operations. After the quantum circuit is constructed, each qubit in the quantum register is measured. It collapses the superposition into a definite state with the classical outcomes recorded in the classical register. The transpiled circuit is optimized, and a specific backend is used to ensure compatibility with available quantum operations. The simulation is run multiple times (20,000 shots) using the Basic and AerSimulator backend to verify the noisy and noiseless measurement results. The collected measurement counts are then transformed into a bitstring by repeating each unique measurement outcome according to its frequency and concatenating them into a single sequence that reflects the quantum randomness generated by this algorithm. This method leverages the principles of quantum mechanics such as superposition and entanglement to produce a sequence of bits that are inherently unpredictable making it suitable for cryptographic and random number generation applications.

Generation of quantum pseudo random number

The Quantum Pseudo Random Number Generators (QPRNGs) leverage quantum circuits to generate binary sequences that exhibit high statistical randomness but are based on structured entanglement and gate operations. The Clifford combination of Hadamard, Phase and Double CNOT gate[19] creates pseudo randomness.

$$|\psi_{\ initial}\rangle = |0\rangle^{\otimes n} = |000\ldots 0\rangle \tag{7}$$

where $|\psi_{\ initial}\rangle$ represents the initial state of n-qubit quantum system, all the qubits are initialized to $|0\rangle$ state. This forms the standard starting point for most quantum algorithms and circuits. The Hadamard gate is applied on each qubit. The Hadamard gate transform $|0\rangle$ state into superposition of $|0\rangle$ and $|1\rangle$.

$$|\psi_{\ 1}\rangle = |H\rangle^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} |x\rangle \tag{8}$$

where $|x\rangle$ denotes the computational basis states indexed by integer x and summation covers all 2^n possible combinations. The phase gate is introduced here to check the initial state of the quantum state. Every quantum initialization begins with $|0\rangle$ state, if $|1\rangle$ state applied to the quantum circuit, adding a phase factor to the respective qubit. When S gate is applied to all qubits, the resultant state is represented as $|\psi_2\rangle = S^{\otimes n}(\psi_1)$.

$$|\psi_2\rangle = S^{\otimes n}\left(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle\right) = S^{\otimes n}\left(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}e^{i\varphi(x)}|x\rangle\right) \tag{9}$$

where $e^{i\varphi(x)}$ is the phase factor depending on x and $\varphi(x)$ is determined by the binary outcome of 1 in binary string x contributes a phase of 'i'. The phase factor is added selectively to $|1\rangle$ state for all the qubits. The DCX gates entangle the qubits pairwise in a chain-like structure. A Double Controlled Not (DCX) gate is a sequence of two CNOT gates between two qubits. The first CNOT gate has control qubit q_i and target qubit q_{i+1}. The second CNOT gate has control qubit q_{i+1} and target qubit q_i. The DCX gates create the entanglement and correlation between qubits. The DCX Operation on two qubits q_i and q_{i+1}.

$$\text{DCX}\left(|q_i, q_{i+1}\rangle\right) = CNOT_{i\to i+1} \circ CNOT_{i+1\to i}\left(|q_i, q_{i+1}\rangle\right) \tag{10}$$

where $|q_i, q_{i+1}\rangle$ represents two-qubit input state, q_i and q_{i+1} are the qubit positions i and $i+1$.

$CNOT_{i\to i+1}$ represents a CNot gate with qubit i as control and $i+1$ as the target.

' $\circ$ ' is a composition operator indicating that the operations are applied in sequence from right to left.

$CNOT_{i+1\to i}$ represents a CNot gate with qubit $i+1$ as control and i as the target.

The DCX gate added to the n qubits, it creates an entangled pair of $q_0 \leftrightarrow q_1, q_1 \leftrightarrow q_2$ upto $q_{n-2} \leftrightarrow q_{n-1}$. This entangled pair modifies the quantum state's amplitude distribution, creating a pseudo-random state. The unitary operator acting of DCX gates acting on n qubits is represented here. U_{DCX}. The final state representation of a quantum pseudo-random circuit is

$$\left|\psi_{final}\right\rangle = U_{DCX}S^{\otimes n}|H\rangle^{\otimes n}|0\rangle^{\otimes n} \tag{11}$$

where U_{DCX} represents the unitary operation corresponding to the DCX gate, it performs a conditional swap of qubit states and introduces controlled interaction with entanglement between adjacent qubits. Every qubit of the quantum circuit is measured in the computational basis $\{|0\rangle, |1\rangle\}$. The quantum state collapsed into 2^n possible basis state $|x\rangle$. The probability of measuring a particular n-bitstring x is represented as

$$P(x) = |\langle x|\,|\psi_{final}\rangle|^2 \cdot (x \in \{0,1\}^n) \tag{12}$$

where Eq. 11 represents the inner product between the computational basis state and the final quantum state and gives the square of the absolute value of amplitude corresponding to outcome x. $\left|\psi_{final}\right\rangle$ contains the pseudo-random amplitude distribution generated by the Clifford gates.

$$\left|\psi_{final}\right\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}c_x|x\rangle \tag{13}$$

Where $\frac{1}{\sqrt{2^n}}$ is a normalization factor for an n-qubit system, it ensures the total probability sums to 1. c_x is the complex amplitudes determined by the Clifford combinations of Hadamard, Phase and DCX gates. The amplitude of c_x contains pseudo-random phase and correlations introduced by the quantum gates. Post-measurement of the quantum state collapses to one of the basis states $|x\rangle$. Here 'x' is a bit string of the n qubit length. Each measurement outcome represents a quantum pseudo-random number in the range $\left(0 - 2^{n-1}\right)$. If the qubit length is 5, the random number range between [0 to 31]. Increasing the number of qubits allows for generating larger pseudo-random numbers and greater randomness due to the higher dimensional quantum state.

$$|0\rangle^{\otimes n} \xrightarrow{|H\rangle^{\otimes n}} \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle \xrightarrow{|S\rangle^{\otimes n}, U_{DCX}} \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}c_x|x\rangle \xrightarrow{\text{measure}} \textit{Classical outcome} \tag{14}$$

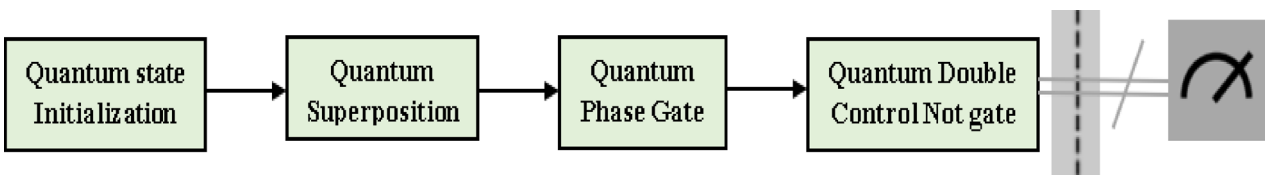


Fig. 3. Conceptual flow diagram of proposed QPRNG.

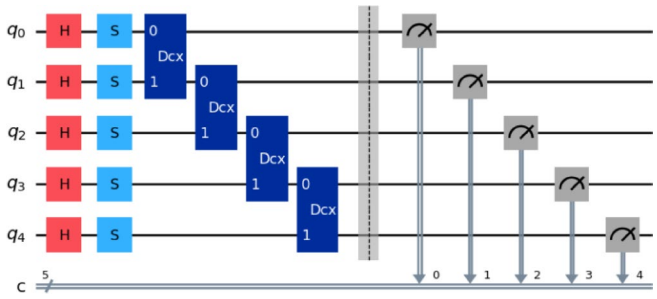


Fig. 4. Software implementation of QPRNG (5 qubit circuit).

Quantum gate	Parameter	Purpose	Explanation
Hadamard gate	Rotation around Y-axis $\left(\frac{\pi}{2}\right)$ and X-axis (π)	Creates equal superposition of basis states	It prepares qubits in a superposed state and enables quantum parallelism.
Phase gate	Phase shift $\left(\frac{\pi}{2}\right)$, Rotation around Z axis	Applies a fixed phase shift	It performs phase modulation by rotating the qubit state $\left(\frac{\pi}{2}\right)$ around Z axis.

Table 2. Proposed QPRNG components with parameter.

Quantum gate	Parameter	Purpose	Explanation
Double Control Not gate	Swap two qubits conditionally	Applies a bidirectional controlled-Not logic	It performs two CNOT gates with swapped control and target and is useful for qubit routing and efficient entanglement.

Equation 14 illustrates the transformation of an initial quantum state into a complex superposed and entangled state. Hadamard creates an equal superposition and phase modulation using S gate with entanglement through U_{DCX} modifies the phase and correlation between qubits. A measurement collapses this quantum state into classical bitstring outcome based on the probability. $|x\rangle^2$ enabling quantum pseudo randomness extraction.

Figure 3 illustrates a QPRNG, where a sequence of quantum operations transforms initialized qubits into deterministic but unpredictable outcomes upon measurement.

The QPRNG begins with quantum state initialization followed by superposition through Hadamard gates to expand the state space. The quantum phase gates are applied to encode phase-based variations that alter the interference patterns. The core quantum entangling DCX gate creates bidirectional entanglement and allows small changes in one qubit to affect others. This deterministic evolution of qubit states simulates randomness by generating high-complexity quantum states. The final measurement yields bitstring that appear random due to the structure and depth of gate operations. The overall conceptual flow of QPRNG focus to create structured unpredictability through gate complexity and entanglement.

Figure 4 proposed a quantum circuit for pseudo random number generation, adding an equal number of quantum and classical register in every qubit. Initialize $|0\rangle$ or $|1\rangle$ state and Hadamard in all qubits for superposition. If the quantum circuit initialised $|1\rangle$ state, it introduces the phase factor. Apply double controlled not gate for a controlled entanglement. Finally, each qubit of the quantum circuit is measured to produce a quantum pseudo random number.

The Table 2 summarizes the quantum gate components used in the proposed QPRNG architecture. The Hadamard gate generates superposition states and quantum parallelism essential for randomness. The Phase gate introduces a fixed phase shift via Z-axis rotation enhances state variability. The DCX gate implements bidirectional conditional logic through dual CNOT operations supports qubit routing and entanglement. These gates ensure coherent state manipulation and efficient pseudo-random number generation in the quantum domain.

The proposed QPRNG algorithm utilizes quantum superposition and deterministic operations to generate pseudo-random numbers. Hadamard gates are applied to all qubits in the quantum register, creating an equal superposition state by evenly distributing probability amplitudes across all possible states. A phase gate introduces deterministic phase changes for the $|1\rangle$ state, while double CNOT gates ensure controlled deterministic operations. Barriers are incorporated to separate the preparation and measurement stages, enhancing visual circuit readability. During measurement, each qubit collapses from superposition into a definite state with outcomes recorded in the classical register. The circuit is transpiled for backend compatibility, optimizing gate operations for efficient execution. Simulations are performed with 20,000 shots using both basic and aer simulator backends to analyze noisy and noiseless outputs. The measurement results are processed into a bitstring by repeating each unique outcome according to its frequency and concatenating them into a single sequence. Unlike true randomness, which is inherently unpredictable, the QPRNG ensures that the same initial conditions (quantum circuit configuration and seed) will always yield the same output sequence.

Input: n – Number of qubits, $shots$ – Number of shots for stimulation runs
Output: S_2 − Concatenated bitstrings based on measurement frequency
Require: $backend \leftarrow$ BasicProvider.get_backend('basic_simulator')
$backend \leftarrow$ Aer.get_backend('Aer_simulator')
while $n > 0$ **do**
Initialize $qr \leftarrow$ QuantumRegister, $cr \leftarrow$ ClassicalRegister, $ckt \leftarrow$ QuantumCircuit of n size
for i in range(n) **do**
$ckt.h(qr[i])$
$ckt.s(qr[i])$
end for
for i in range$(n-1)$ **do**
$ckt.dcx(\ qr[i], qr[i+1])$
end for
$ckt.barrier()$
$ckt.measure(qr, cr)$
$transpile_{circuit} \leftarrow transpile(ckt, backend)$
$result \leftarrow backend.run(transpiled_{circuit}, shots = shots).result()$
$counts \leftarrow result.get\_counts()$
$S_2 = \bigoplus_{x \in counts} x^{f(x)}$
end while
return S_2

Algorithm 2 . Quantum pseudo random number generation.

This determinism is particularly advantageous in cryptographic applications where repeatability is essential for encryption and decryption. Additionally, the algorithm's use of quantum principles like superposition and controlled operations adds complexity and security, making it resistant to traditional brute-force attacks. The deterministic nature of the QPRNG also ensures that the generated sequences can be independently verified a critical feature for ensuring reliability and trust in secure systems. This method uses deterministic quantum operations to produce reproducible pseudo-random bit sequences, making it suitable for cryptographic and random number generation tasks.

Generation of quantum hybrid random number

The traditional QTRNG rely purely on the inherent randomness of quantum measurements, while QPRNG exploit deterministic but complex operations. This proposed work bridges the gap by combining these approaches into hybrid framework. A true random seed provided by QTRNG ensures the initialisation phase's unpredictability. The transformation of initialized state is enhanced by Quantum Clifford gates introducing pseudo randomness. After measuring the quantum circuit, the classical outcomes show a hybrid randomness. This hybrid model ensures unpredictability from QTRNG and Scalability from QPRNG. The true random bit sequence $\mathrm{x} \in \{0,1\}^n$ is generated using proposed QTRNG. This bit sequence is used to initialize the qubits in the computational basis $|x\rangle$

$$|\psi_0\rangle = |x\rangle = \ |x_0,\ x_1, x_{2\ldots\ldots} x_{n-1}\rangle\ x_i \in \{0,\ 1\} \tag{15}$$

where x is a binary string of length. The seed is fed into a QPRNG circuit, which applies Clifford gates to generate a longer sequence with higher entropy. For n-qubit input $|\psi_0\rangle$, applying $|H\rangle^{\otimes n}$ result in

$$|\psi_1\rangle = |H\rangle^{\otimes n} |\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{x.y} |x\rangle \tag{16}$$

where $x.y$ is the bitwise dot product between initialized state x and basis state y. The phase gate introduces a complex coefficient i^k. Here k depends on the binary state of the qubit.

$$|\psi_2\rangle = \ S^{\otimes n} |\psi_1\rangle = \ \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{x.y}\ i^{wt(x)} |x\rangle \tag{17}$$

where $i^{wt(x)}$ is phase rotation introduced by S gate, wt(x) denotes the hamming weight in the bitstring x. The DCX gates is a controlled-X gate with target swap operation between the qubit. It creates entanglement by the flipping the states of selected qubits based on control qubits.

$$DCX\ (|\ q_i, q_j\rangle\ ) = |q_i(q_i \oplus\ q_j)\rangle \tag{18}$$

where $|q_i\ (q_i \oplus\ q_j)\rangle$ represents the first qubit remains unchanged and the second qubit becomes XOR of $q_{i,j}$ reflecting the effect of conditional gate operation. The DCX operation modifies the paired states of qubits. It applied across the qubit pairs, the DCX gates transform the state $|\psi_2\rangle$ into a highly entangled state.

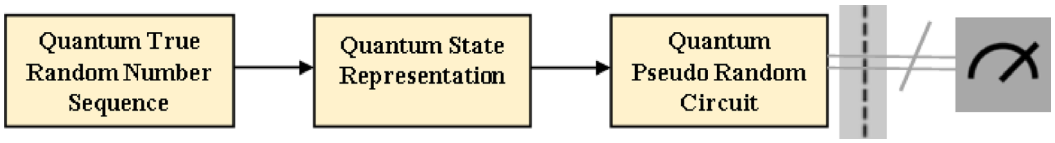


Fig. 5. Conceptual flow diagram of proposed QHRNG.

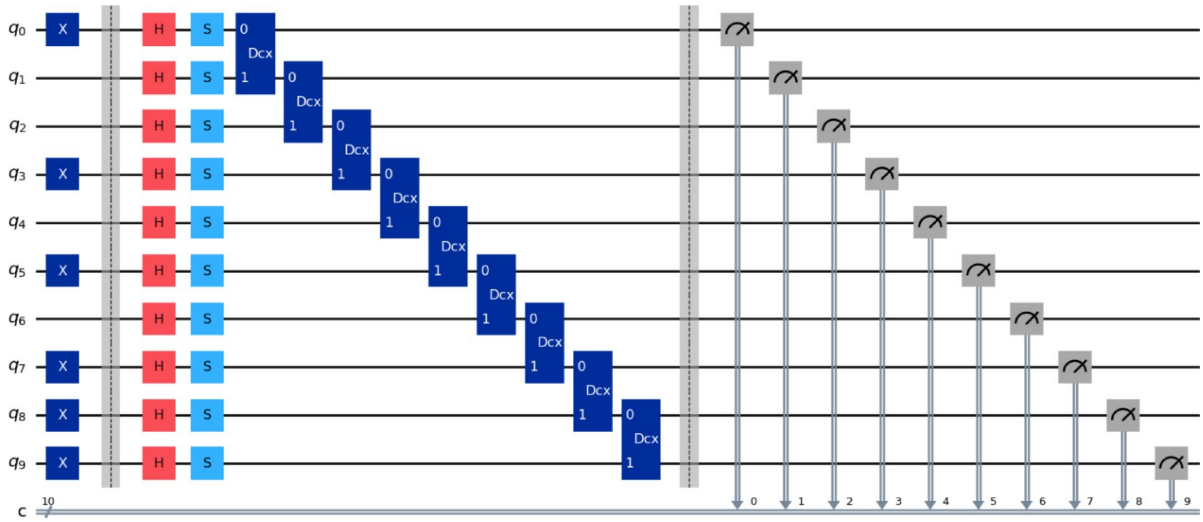


Fig. 6. Software implementation of QHRNG (10 qubit circuit).

$$|\psi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} c_x |x\rangle \tag{19}$$

where, c_x does the entanglement structure influence the complex amplitudes. The quantum state $|\psi_3\rangle$ is measured in the computational basis. The probability outcome of a classical bitstring $x \in \{0, 1\}^n$. $P(x) = |c_x|^2$ where, $|c_x|^2$ is the magnitude squared of the amplitude of $|x\rangle$. This process of true random initialization to pseudo-random final outcome creates a hybrid structure of randomness.

$$|x\rangle^{\otimes n} \xrightarrow{|H\rangle^{\otimes n}|S\rangle^{\otimes n}, U_{DCX}} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} c_x|x\rangle \xrightarrow{} Classical\ outcome \tag{20}$$

This proposed Eq. 20 is an essence of QHRNG blending true randomness with pseudo-randomness to produce the final outcome.

This Fig. 5 shows starts with a true random number from a quantum source turns it into a quantum state, runs through QPRNG circuit that adds more randomness and then measures it to get a hybrid random outcome.

The QHRNG integrates the advantage of QTRNG and QPRNG to achieve superior randomness. The process starts with generating a quantum true random sequence using a subcircuit. This truly random sequence is initialized to quantum register, each qubit flipped using Pauli-X gates depending on the seed value. This enables a non-deterministic initialization configuration. A structured pseudo random quantum circuit including phase modulation (S gate), entanglement (DCX gate) and optimal mixing with superposition (H gate). This layered circuit to propagate and amplify initial randomness while adding deterministic output. QHRNG aims to combine unpredictability with controllable structure and it ensures high randomness.

In Fig. 6 proposed quantum circuit for hybrid random number generation, adding equal number of quantum and classical register in every qubit. Initialize quantum true random sequence in every qubit state. Add Hadamard in all qubits for superposition. If the quantum circuit is initialized $|1\rangle$ state, it introduces phase factor. Apply double controlled not gate for a controlled entanglement. Add a barrier in every step to avoid circuit overlapping and better visualization of quantum circuit Finally, measure each qubit of the quantum circuit to produce a quantum hybrid random number.

The Table 3 presents the quantum gate components employed in the proposed QHRNG architecture. The Pauli-X gate initializes the quantum state by flipping qubits enables controlled state preparation. The Hadamard gate creates superposition states for essential quantum randomness. The Phase gate refines the quantum phase and additional control over randomness distribution. The QTRNG-derived bits introduce hybrid pseudo-

Quantum Gate	Parameter	Purpose	Explanation
Pauli X gate	Flips the quantum state $\|0\rangle \rightarrow \|1\rangle$	State Initialization	It prepares selected qubits for quantum state encoding.
Hadamard gate	Rotation around Y-axis $\left(\frac{\pi}{2}\right)$ and X-axis (π)	Superposition creation	It converts bases into superposition quantum state.
Phase gate	Phase shift $\left(\frac{\pi}{2}\right)$, Rotation around Z axis	Phase modulation	It refining the phase for randomness control

Table 3. Proposed QHRNG components with parameter.

Input: n – Number of qubits, $shots$ – Number of shots for stimulation runs
$initial_{state}$ – $S_1 \leftarrow QTRNG\ Bitstring$
Output: S_3 – Concatenated bitstrings based on measurement frequency
Require: $backend \leftarrow$ BasicProvider.get_backend('basic_simulator')
$backend \leftarrow$ Aer.get_backend('Aer_simulator')
while $n > 0$ **do**
Initialize $qr \leftarrow$ QuantumRegister, $cr \leftarrow$ ClassicalRegister, $ckt \leftarrow$ QuantumCircuit of n size
for i in range(n) **do**
if $initial_{state[i]} ==$ "1" **do**
$ckt.x(qr[i])$
end if
end for
for i in range(n) **do**
$ckt.h(qr[i])$
$ckt.s(qr[i])$
end for
for i in range$(n-1)$ **do**
$ckt.dcx(\ qr[i], qr[i+1])$
end for
$ckt.barrier()$
$ckt.measure(qr, cr)$
$transpile_{circuit} \leftarrow transpile(ckt, backend)$
$result \leftarrow backend.run(transpiled_{circuit}, shots = shots).result()$
$counts \leftarrow result.get\_counts()$
$S_3 = \bigoplus_{x \in counts}\ x^{f(x)}$
end while
return S_3

Quantum Gate	Parameter	Purpose	Explanation
Double Control Not gate	QTRNG-derived bits	Pseudo random logic using true randomness	It applies double CNOT logic controlled by QTRNG output and generate hybrid random sequences.
Measurement gate	Output	Generation of hybrid random bits	Measures qubit superposition to extract unpredictable random bits.

Algorithm 3 . Hybrid quantum random number generation.

random logic by combining classical and quantum randomness. Finally, the measurement gate extracts hybrid random bits by collapsing quantum states ensures unpredictability in output.

The proposed QHRNG integrates quantum superposition, entanglement and phase modulation. The process begins by generating a random binary seed using a QTRNG module. Upon measurement, this produces a non-deterministic binary sequence which serves as the initialization seed for the hybrid model. This initialization is based on principles such as Born's rule governs the probabilistic nature of measurement outcomes, the no-cloning theorem prevents replication of quantum states and entropy maximization through entanglement. Each bit of the QTRNG seed is encoded into a quantum state using Pauli-X gates. To enhance the complexity of state space, S gates were applied to all initialized qubits. This introduces global interference patterns and affects measurement outcomes in non-trivial ways. The pseudo randomness layer is then implemented using DCX gates applied between adjacent qubit pairs to generate structured entanglement across the quantum

register enables both complexity and reproducibility. The measurement of final quantum state collapses the entangled superposition into high-entropy bitstrings. These output sequences benefit from true randomness during initialization, structured pseudo-random transformations and measurement indeterminacy. This hybrid circuit structure ensures randomness is not only seeded from a fundamentally non-deterministic source but also expanded through deterministic yet complex gate operations. QHRNG ensures high entropy of quantum measurements and the gate-level complexity of Clifford-based circuits to enhance the quality and security of the generated random sequences. The hybrid circuit was tested extensively using IBM Qiskit backends including noiseless simulators (BasicSimulator) and noisy simulators (AerSimulator) as well as IBM's physical hardware with 20,000-shot executions to ensure statistical reliability. The output sequences were evaluated using NIST SP 800 – 22 and 800-90B statistical test suites confirming their strong randomness characteristics. The design methodology ensures QHRNG benefits from the unpredictability of quantum measurement and the structure of circuit level entanglement making it suitable for secure cryptographic applications.

Proposed quantum image encryption and decryption scheme

Novel enhanced quantum image representation

Zhang et al.,[20] proposed the Novel Enhanced Quantum Image Representation used to convert classical to quantum images. The gray scale image representation $2^n \times 2^n$, the pixel values typically ranging from 0 to 255 and commonly expressed as 2^q (q = 8). The binary sequence $\psi^{0}_{YX}, \psi^{1}_{YX}, \psi^{2}_{YX,} \psi^{3}_{YX,} \cdots \psi^{q-2}_{YX,} \psi^{q-1}_{YX}$ encodes the gray-scale values of corresponding pixel location "YX". A Fig. 7 shows a sample 4 × 4 image representation.

$$f(YX) = \psi^{0}_{YX}, \psi^{1}_{YX}, \psi^{2}_{YX,} \psi^{3}_{YX,} \ldots\ldots \psi^{q-2}_{YX,} \psi^{q-1}_{YX,} \psi^{k}_{YX} \in [0,\ 2^{q-1}] \tag{21}$$

where ψ represents a quantum operator with "YX" indicating the pixel location. The image is encoded using NEQR circuit expressed as,

$$|I_{gray}\rangle = \frac{1}{2^n} \sum_{y=0}^{2^{n-1}} \sum_{x=0}^{2^{n-1}} |g(Y,X)\rangle |YX\rangle \tag{22}$$

where, $|I_{gray}\rangle$ denotes the quantum version of gray scale image, $g(Y,\ X)$ denotes the gray-scale value of image at pixel (Y, X) which is stored in the basis qubit state $|f(Y,X)\rangle$. The NEQR model requires $q + 2n$ qubits to construct the quantum circuit for an image size $2^n \times 2^n$ with grayscale range 2^q. The pixel values for each row are mapped to quantum states as follows:

$$\begin{aligned}
|Pixel_{00}\rangle &= (|73\rangle \otimes |0000\rangle) + |255\rangle \otimes |0001\rangle + |24\rangle \otimes |0010\rangle + |121\rangle \otimes |0011\rangle \\
|Pixel_{01}\rangle &= (|32\rangle \otimes |0100\rangle) + |112\rangle \otimes |0101\rangle + |79\rangle \otimes |0110\rangle + |16\rangle \otimes |0111\rangle \\
|Pixel_{10}\rangle &= (|132\rangle \otimes |1000\rangle) + |83\rangle \otimes |1001\rangle + |13\rangle \otimes |1010\rangle + |251\rangle \otimes |1011\rangle \\
|Pixel_{11}\rangle &= (|252\rangle \otimes |1100\rangle) + |39\rangle \otimes |1101\rangle + |172\rangle \otimes |1110\rangle + |112\rangle \otimes |1111\rangle
\end{aligned} \tag{23}$$

The pixel values are encoded into the quantum circuit, the decimal values of the gray-scale value are converted into binary strings and encoded into the quantum registers.

$$\begin{aligned}
|Pixel_{00}\rangle &= (|01001001\rangle \otimes |0000\rangle) + |11111111\rangle \otimes |0001\rangle + |00011000\rangle \otimes |0010\rangle + |01111001\rangle \otimes |0011\rangle) \\
|Pixel_{01}\rangle &= (|00100000\rangle \otimes |0100\rangle) + |01110000\rangle \otimes |0101\rangle + |01001111\rangle \otimes |0110\rangle + |00010000\rangle \otimes |0111\rangle \\
|Pixel_{10}\rangle &= (|10000100\rangle \otimes |1000\rangle) + |01010011\rangle \otimes |1001\rangle + |00001101\rangle \otimes |1010\rangle + |11111011\rangle \otimes |1011\rangle \\
|Pixel_{11}\rangle &= (|11111100\rangle \otimes |1100\rangle) + |00100111\rangle \otimes |1101\rangle + |10101100\rangle \otimes |1110\rangle + |01110000\rangle \otimes |1111\rangle
\end{aligned} \tag{24}$$

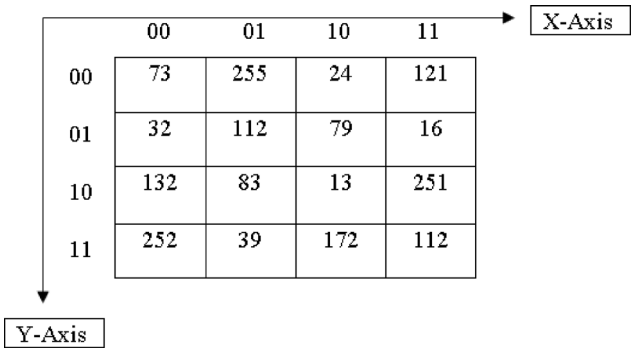


Fig. 7. A 4 × 4 Gy-scale image grid.

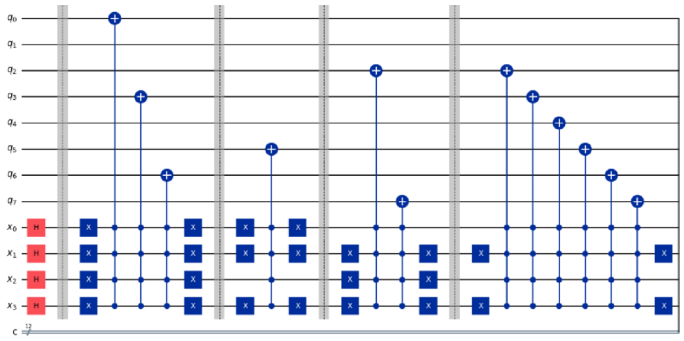


Fig. 8. Software implementation of NEQR Circuit.

Parameter	Quantum component	Purpose	Explanation
x_2	CNOT	Bitwise scrambling of pixel values	It performs quantum XOR by flipping each bit b_k using the control qubit x_2.
x_3	Controlled-SWAP	Qubit position scrambling	It conditionally swaps intensity qubit positions when $\|x_3\rangle = 1$, increasing quantum state diffusion.

Table 4. Gate iteration parameter for quantum bit-level scrambling.

In Eqs. 23&24, ' $\otimes$ ' denotes the tensor product of two quantum states. The entire image representation in quantum form is expressed as

$$|I_{gray}\rangle = |Pixel_{00}\rangle + |Pixel_{01}\rangle + |Pixel_{10}\rangle + |Pixel_{11}\rangle \tag{25}$$

$$\begin{aligned} |I_{gray}\ \frac{1}{2} &= [(|010010010000\rangle + |111111110001\rangle + |000110000010\rangle \ + |011110010011\rangle \\ &+ |001000000100\rangle + |011100000101\rangle \ + |010011110110\rangle \ + |000100000111\rangle \\ &+ |100001001000\rangle + |010100111001\rangle + |000011011010\rangle + ||111110111011\rangle \\ &+ |111111001100\rangle + |001001111101\rangle \ + |101011001110\rangle \ + |011100001111\rangle] \end{aligned} \tag{26}$$

In Fig. 8 shows the Qiskit simulation of image representation of corresponding pixel to the quantum states is represented.

Proposed quantum encryption and decryption

Quantum image encryption

Step 1 A grayscale image I of dimension of $M \times N$ is divided into smaller non-overlapping blocks I_b of size 4×4. as shown in Fig. 6. The three different quantum keys are generated. The initial step includes loading each subblock image converts into a grayscale format and then transforming into a numpy array. Each pixel value in this array is then converted to its binary form preparing it for XOR encryption.

$$I_b\,(i,j) = \sum {}^{7}_{k=0} b_k.2^k\ b_k \in \{0,1\} \tag{27}$$

where, $I_b\,(i,j)$ is represents the grayscale pixel intensity values in each block $I_b\,(i,j) \in \{0,1,\ldots\ 255\}$and b_k are the binary bits. Each pixel intensity $I_b\,(i,j)$ is converted into its binary representation.

Step 2 The image processed into proposed three quantum generated keys ($S_1,\ S_2,\ S_3$). Quantum encryption is realized by mapping XORed image into a quantum circuit. The binary representation of each pixel $I_b\,(i,j)$ is XORed with the corresponding bit in the quantum key.

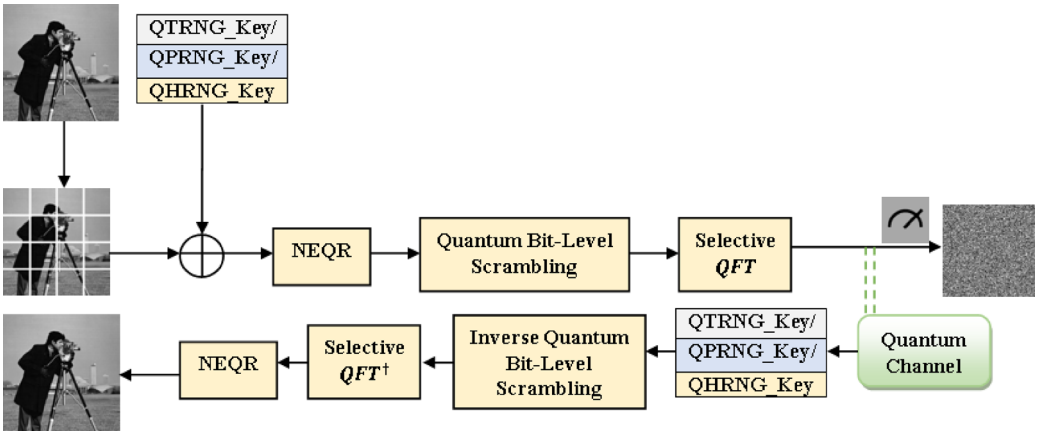
$$Q\,(i,j) = I_b\,(i,j)\ \oplus\ S_k \tag{28}$$

where, ' $\oplus$ ' represents bit-wise XOR operation between image and quantum key. $Q\,(i,j)$ is the quantum randomized pixel value and S_k is the binary string quantum keys. The proposed algorithm individually executed with these quantum key as $x_1.\,(S_1 - QTRNG,\ S_2, -QPRNG\ S_3 - QHRNG)$.

Step 3 The NEQR encodes grayscale images into quantum states. The two separate set quantum registers are used, one for the pixel values labelled intensity and the other for the pixel positions labelled index. The equal number of quantum and classical registers are used. In every positional qubit, quantum Hadamard gate takes an advantage of all positions of input image. The multi control NOT gate is utilized to encode the pixel value of the image into quantum states. When the control qubit is set to to $|1\rangle$ state, it causes the target qubits to flip from the $|0\rangle$ to $|1\rangle$ state. Each pixel is encoded as.

Quantum component	Parameter	Purpose	Explanation	Role in Quantum Encryption
Hadamard gate	Rotation around Y axis $\left(\frac{\pi}{2}\right)$ and X axis (π)	Creates superposition of basis states	It prepares qubits in equal superposition for quantum parallelism	It converts pixel intensities into frequency components by enabling QFT.

Table 5. Selective QFT parameter.



Quantum component	Parameter	Purpose	Explanation	Role in Quantum Encryption
Controlled Rotation Z gate	$\left(\frac{\pi}{2}\right)$	Applies controlled phase shift for entanglement	It introduces conditional phase rotations to maintain quantum interference	It encodes pixel position-dependent phase shifts that contribute to encryption. To decrypt apply a negative phase shift across the qubit

Fig. 9. Proposed quantum encryption and decryption.

$$|I_{pixel}\rangle = |P_x\rangle \otimes |P_y\rangle \otimes |Q_g\rangle \tag{29}$$

where, ' $\otimes$ ' is a tensor product. $|P_x\rangle$ and $|P_y\rangle$ are the Quantum states representing the row and column positions of the pixel and $|Q_g\rangle$ is Quantum state representing gray scale intensity of the pixel. The Quantum state encoding for key randomized pixel $Q(i,j)$, the row i and column j are represented as $|P_x\rangle = |i_b\rangle$, $|P_y\rangle = |j_b\rangle$ were the binary strings (i_b, j_b) representing the image position. The pixel value $|Q_g\rangle$ is represented by $|b_7 b_6 \ldots b_0\rangle$.

$$|I_{pixel}\rangle = |i_b\rangle \otimes |j_b\rangle \otimes |b_7 b_6 \ldots b_0\rangle \tag{30}$$

Equation 30 represents the tensor product of row and column indices with binary encoded intensity value forming the complete quantum state of a pixel.

Step 4 The quantum gate iteration parameters (x_2, x_3) for quantum bit-level scrambling during encryption. For each bit b_k of the pixel value $Q(i_b, j_b)$, a quantum XOR operation is applied as $b'_k = b_k ? x_2$ where, b'_k is the scrambled bit. This operation is implemented using a controlled-Not gate. In Table 4, the parameters for quantum bit-level scrambling.

$$CNOT : |b_k\rangle \otimes |x_{2,3}\rangle \rightarrow |b_k \oplus x_2\rangle \otimes |x_3\rangle \tag{31}$$

Equation 31 represents the action of control gate. The gate iteration parameters act as the control qubit and image pixel b_k as the target qubit. When $|x_3\rangle = 1$, an X gate is applied to $|b_k\rangle$ controlled by the x_2 qubit, effectively flipping the bit. Additionally, a swap gate operation is used to further scrambles the quantum state.

Step 5 The Hadamard and controlled Rotation-Z $\left(\frac{\pi}{2}\right)$ Based selective QFT introduces quantum interference, adding an additional layer of randomness and encryption by transforming the pixel qubits on a Fourier basis. Measure all the qubits in the NEQR-encoded quantum state to retrieve the scrambled and diffused image data.

$$|I_{pixel}\rangle = |i_b\rangle \otimes |j_b\rangle \otimes |b'_7 b'_6 \ldots b'_0\rangle \tag{32}$$

Input: $folder_{path}$: Path to subblock classical images ⟵ $subblock_p$
$block_{size}$: Block dimensions (4×4)
$x1_{key}$: Quantum Secure Key (QTRNG/QPRNG/QHRNG)
$x2_{key}, x3_{key}$: Quantum Gate iteration parameter
Require: $backend \leftarrow$ BasicProvider.get_backend('basic_simulator')
$backend \leftarrow$ Aer.get_backend('Aer_simulator')
Output: Encrypted subblock images save in an output folder
while $p \leq n$ **do**
$g_{channel} \longleftarrow$ grayscale channel of $subblock_p$, $binary \longleftarrow$ convert $g_{channel}$ to $binary$
$randomized_{message} \longleftarrow$ binary data XOR with $x1_{key}$
Initialize Registers ⟵ $Index, Intensity, cr$. $qc \longleftarrow$ QuantumCircuit(Index, Intensity, cr)
$qc.id(Intensity)$
$qc.h(Index)$
for $i = 1$ to rows (image) **do**
for $j = 1$ to columns (image) **do**
$pos_x, pos_y \longleftarrow$ binary locations of (i, j)
for b in binary of (pos_x, pos_y) **do**
$qc.x$ (Qubits corresponding to $b = 0$)
end for
$pixel_{value} \longleftarrow$ binary of $XORed$ pixel
for $b = 1$ to bit-length of $pixel_{value}$ **do**
$qc.mcx(Index, Intensity[b])$
end for
$qc.x$ (Qubits corresponding to $b = 0$)
end for
end for
for $k = 1$ *to* $x2_{key}$ **do**
$qc.cx$ (Neighboring $Intensity$ qubits)
end for
for $k = 1$ *to* $x3_{key}$ **do**
$qc.swap$ ($Intensity$ qubits)
end for
$selective_{qft} \longleftarrow [Hadamard, CRZ, Phase]$
for each $Intensity$ qubits **do**
Apply gates from $selective_{qft}$
end for
$qc.measure(all\ qubits, cr)$
$transpile_{circuit} \leftarrow transpile(ckt, backend)$
$result \leftarrow backend.run(transpiled_{circuit}, shots = shots).result()$
$counts \leftarrow result.get\_counts()$
Parse $counts$ into $pixel_{locations} and\ values$
Build $pixel_{grid}$ from parsed values
Save $pixel_{grid}$ as encrypted subblock image.
Apply a Hadamard gate for all qubits # if receiver
Measure all qubits and store the results in classical register # if intruder
end while

Algorithm 4. Quantum encryption.

Equation 32 represents the encrypted quantum pixel state formed by the tensor product of row and column indices and the encrypted intensity value $|b'_7 b'_6 \ldots b'_0\rangle$. Here b' is scrambled pixel bits.

The Table 5 summarizes the quantum components and parameters used in the Selective QFT for quantum encryption. The Hadamard gate enables superposition to facilitate quantum parallelism and the Controlled Rotation-Z gate applies conditional phase shifts to preserve quantum interference. These gates play a key role in quantum-encrypted data by manipulating phase components.

From the measurement results, parse the pixel position and intensity to construct the encrypted image. The Hadamard gates are applied during the final stage of encryption[29] to transform the quantum state into a superposition, securing the encoded image before transmission to the receiver. The Fig. 9 proposed method uses block-based approach enables parallel processing, efficient and scalable encryption.

Parameter	Quantum component	Operation	Purpose	Explanation
x_2	CNOT	Reversal of controlled-NOT	Undo bitwise masking on neighboring qubits	CNOT gates are applied in reverse order for restore intensity qubit values
x_3	SWAP	Reversal of SWAP operations	Undo scrambling of qubit positions	SWAP gates are reversed for restoring original qubit ordering

Table 6. Key iteration parameters of inverse quantum bit-level scrambling.

Input: Apply Hadamard gate in each qubit # Reverse initial superposition
$folder_{path}$: Path to subblock encrypted images ⟵ $subblock_p$
$x1_{key}$: Quantum Secure Key (QTRNG/QPRNG/QHRNG)
$x2_{key}, x3_{key}$: Quantum Gate iteration parameter
Require: $backend$ ← BasicProvider.get_backend('basic_simulator')
$backend$ ← Aer.get_backend('Aer_simulator')
Output: Decrypted subblock images save in an output folder
while $p \leq n$ **do**
$g_{channel}$ ⟵ grayscale channel of $subblock_p$, $binary$ ⟵ convert $g_{channel}$ to $binary$
$decrypted_{message}$ ⟵ binary data XOR with $x1_{key}$
Initialize Registers ⟵ $Index, Intensity, cr$. qc ⟵ QuantumCircuit(Index, Intensity, cr)
$qc.id(Intensity)$
$qc.h(Index)$
for $i = 1$ to rows (image) **do**
for $j = 1$ to columns (image) **do**
pos_x, pos_y ⟵ binary locations of (i, j)
for b in binary of (pos_x, pos_y) **do**
$qc.x$ (Qubits corresponding to $b = 0$)
end for
$encrypted_{pixel_{value}}$ ⟵ binary of $encrypted_{pixel}$
for $b = 1$ to bit-length of $encrypted_{pixel_{value}}$ **do**
$qc.mcx(Index, Intensity[b])$
end for
$qc.x$ (Qubits corresponding to $b = 0$)
end for
end for
for $k = x2_{key}$ to 1 **do**
$qc.cx$ (Neighboring $Intensity$ qubits)
end for
for $k = x3_{key}$ to 1 **do**
$qc.swap$ ($Intensity$ qubits)
end for
$selective_{qft_{dec}}$ ⟵ $[Phase, CRZ, Hadamard]$
for each $Intensity$ qubits **do**
Apply gates from $selective_{qft_{dec}}$
end for
$qc.measure(all\ qubits, cr)$
$transpile_{circuit}$ ← $transpile(ckt, backend)$
$result$ ← $backend.run(transpiled_{circuit}, shots = shots).result()$
$counts$ ← $result.get\_counts()$
Parse $counts$ into $pixel_{locations} and\ values$
Build $pixel_{grid}$ from parsed values
Save $pixel_{grid}$ as decrypted image.
end while

Algorithm 5. Quantum decryption.

Quantum image decryption

Step1 The encrypted quantum image is transmitted to the receiver encoded as qubits in a superposition of quantum states. To decrypt the image, the receiver applies Hadamard gates to each qubit, effectively reversing the initial superposition and retrieving the original quantum information encoded in the image.

Step 2 The encrypted grayscale image is divided into 4×4 subblocks similar to the encryption process. Each subblock is converted into binary representation of the intensity. The encrypted binary data is XORed with the same quantum key S_1 used during encryption.

Step 3 A quantum circuit is initialized to reverse the quantum transformations and reversing the scrambling and selective QFT[21]. Identity gate is applied to maintain the quantum state without modifying it initially and Hadamard Gate creates uniform superposition states to reverse initial transformations applied to pixel locations. The binary positions ($|P_x\rangle$ and $|P_y\rangle$) of each pixel are used to identify its location in the quantum circuit. If a bit in the binary position is zero, X gate is applied to the corresponding qubit to reverse its flipping. Multi-control X gates are applied to reverse the scrambling at the bit level.

Step 4 The key iteration parameters [Table 6] (x_2, x_3) determine the number of controlled operations performed during decryption. For $x_{key} = x_2$ to 1, neighbouring intensity qubits are reversed by the CNOT gate. For $x_{key} = x_3$ to 1, the SWAP operations between intensity qubits are reversed using the SWAP gate.

Step 5 The inverse selective QFT reverses the superposition state and rotation of qubits for each intensity qubit in the quantum circuit. Inverse operations decode the quantum state back to the binary pixel values. The sender and receiver only know the secure quantum key, selective QFT and Inverse selective QFT circuit combination. Since the qubits are placed into a superposition after QFT, measuring the results of the quantum channel will yield a random value, which will be decided at the time of measurement, making any attempts to verify results in an entirely random output. To recover the original diffused image, receiver must disentangle the qubits using the same gate sequence and obtain all the superposition values. Any intruder measures the qubits without being aware of the correct Inverse selective QFT combination will yield different results. The reverse process of the encryption algorithm is used for decrypting through same secure quantum key. The resulting binary values are transformed back into the same pixel values of original input image, when no intruder in the quantum channel tries to disturb the communication.

Experimental results and discussion

The experiments were performed on a Dell workstation running Window 11 Pro with 16GB RAM, a 64-bit architecture and Intel(R) Xeon Silver 4210R CPU @ 2.40 GHz. This system serves as the host for executing quantum simulations through IBM Qiskit Version 1.1, which emulates quantum behaviour through classical computational resources. The encryption tests were conducted with images from the USC-SIPI[25] image database. The quantum key encryption methods using four test images (Barbara, Cameraman, Pepper, Mandril) across three types of quantum random number generators, namely QTRNG, QPRNG and QHRNG were implemented using Qiskit's simulator backends which are designed to emulate quantum circuit behavior on classical hardware. The Qiskit Basic Simulator is a noiseless simulator and Qiskit Aer Simulator is a noisy simulator in quantum environments. The subcategories "A" and "B" represent the noisy Aer simulator and noiseless basic simulator, respectively. This allows us to evaluate the quantum randomness quality, encryption performance and robustness of proposed algorithm. Performance metrics such as security statistics (NIST SP800-22, SP800-90B, Restart Experiment, Autocorrelation - Randomness analysis, NPCR, UACI, Avalanche effect – differential attacks, Correlation and histogram analysis -statistical analysis, Noise analysis, Key space analysis, Key sensitivity analysis, Chosen/Known plain text attack and computational complexity of algorithm).

Upon analysing the three key generation methods (QTRNG, QPRNG, QHRNG), it has been evident that the QHRNG outperforms the others. Hence, it has been executed on IBM_TORINO backend through IBM Qiskit Version 2.2, a real superconducting quantum processor provided through the IBM Quantum Platform. As illustrated in the attached job summary [Fig. 11a, b and c] the quantum job was successfully processed under the ibm_q/open/main instance. This real-device execution demonstrates the practical feasibility of implementing the proposed QTRNG, QPRNG and QHRNG on quantum physical hardware. This hardware-based realization provides credible proof that our proposed secure quantum key generation model can be deployed on both classical systems and emerging quantum backends offering a balanced hybrid solution.

Quantum hardware simulation of proposed QTRNG circuit

The QTRNG circuit designed to generate high quality quantum random numbers using quantum superposition and controlled phase correlations across a multi-qubit system. This circuit operates on 24 qubits initialized in the $|0\rangle$ state. A layer of Hadamard gates is applied to each qubit to transform the register to uniform superposition ensures that each qubit independently collapses with equal probability. To introduce controlled phase randomness, a sequence of controlled-RZ gates with $\pi/2$ rotation angle is applied adjacent qubit pairs produces entanglement-mediated phase interference patterns enrich the statistical complexity of the output distribution. All qubit measured in the computational basis with the physical topology and native gate set of IBM Torino superconducting quantum processor, the circuit if fully transpiled using Qiskit's transpiler. This step maps logical qubits to hardware qubits resolves connectivity constraints and decomposes the circuit into native operations supported by the device. The QTRNG circuit was executed with 50,000 measurement shots provides a randomness extraction. The measurement results returned by the backend are aggregated by unique bitstrings according to their frequency counts.

```
[ ]: backend = service.backend('ibm_torino')
n_qubits = 24
n_shots = 50000
qr = QuantumRegister(n_qubits, 'q')
cr = ClassicalRegister(n_qubits, 'c')
qc = QuantumCircuit(qr, cr)
for qubit in qr:
    qc.h(qubit)
for i in range(len(qr) - 1):
    qc.crz(pi / 2, qr[i], qr[i + 1])
qc.barrier()
qc.measure(qr, cr)
fig = qc.draw('mpl')
fig.savefig('qtrng_circuit.tiff', format='tiff', dpi=600)
transpiled_circuit = transpile(qc, backend=backend)
sampler = SamplerV2(mode=backend)
job = sampler.run([transpiled_circuit], shots=n_shots)
result = job.result()
counts = result[0].data.meas.get_counts()
bit_strings = ''.join([key * value for key, value in counts.items()])
print("Bitstrings from torino backend:", bit_strings)
```

```
[3]: service = QiskitRuntimeService(
    channel='ibm_quantum_platform',
    instance='crn:v1:bluemix:public:quantum-computing:us-east:a/8f966d007f5443929d259b5b3a16be5b:4f5dee8b-4f3a-408b-a558
)
job = service.job('d4hnc4ccdebc73f29870')
job_result = job.result()
pub_result = job_result[0]
counts = pub_result.data.c.get_counts()
QTRNG_Outcome = ''.join([bits * count for bits, count in counts.items()])
print("\nIBM_Torino_Hardware_Result")
print(QTRNG_Outcome)
```

```
IBM_Torino_Hardware_Result
11010101010001111101101001101010010011011001000110111111100110110000011001111100101001010110100011111111100011100110000111
```

(a)

Fig. 10. (**a**) Hardware Implementation of QTRNG (**b**) Hardware Implementation of QPRNG (**c**) Hardware implementation of QHRNG.

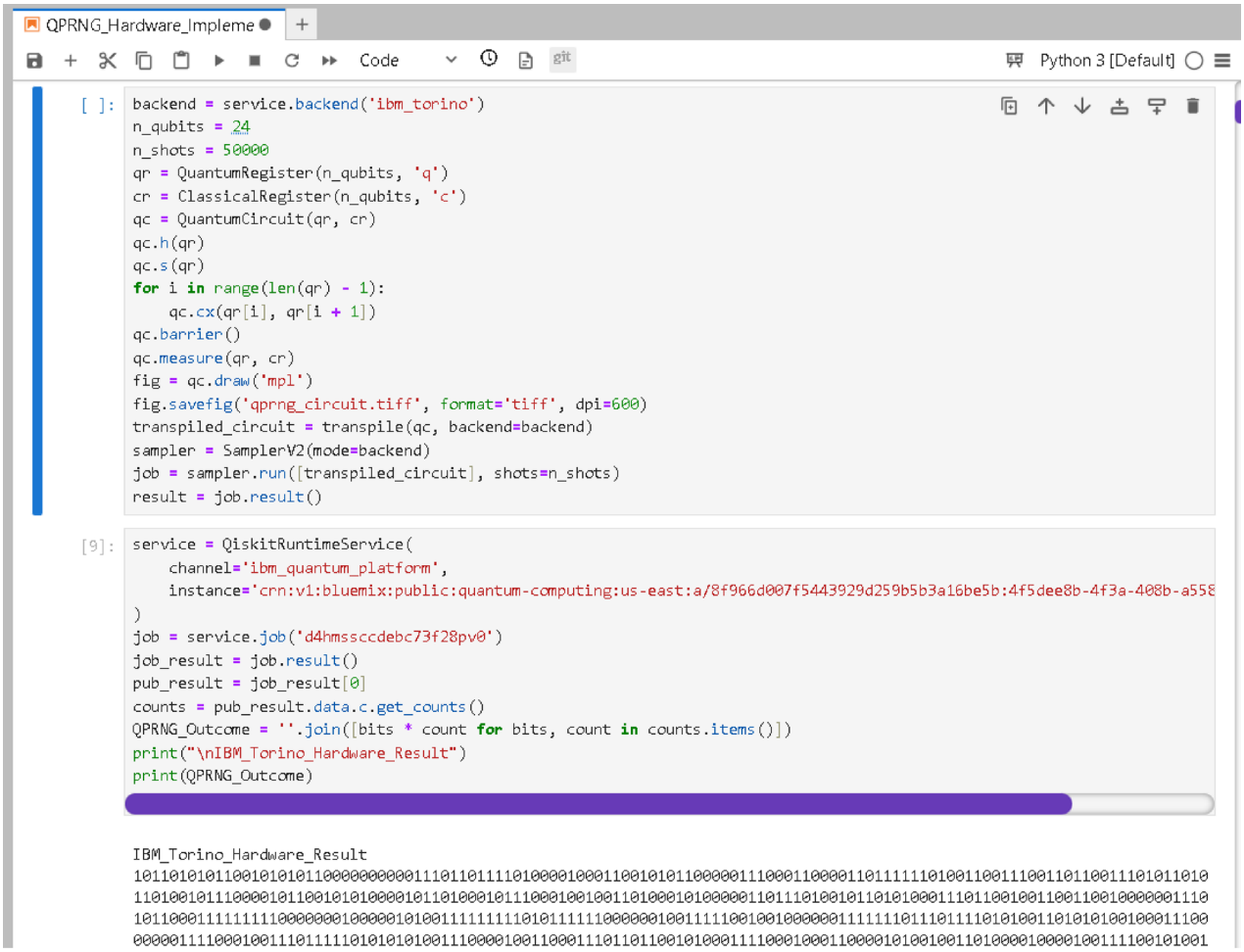
Quantum hardware simulation of proposed QPRNG circuit

The QPRNG circuit constructed using 24 qubits to generate pseudo-random quantum bitstreams by relying on gate operations combined with quantum measurement uncertainty. Unlike QTRNG circuit, QPRNG incorporates a structured sequence of operation to introduce deterministic quantum transformations whose final outcome remain probabilistic due to inherent quantum measurement. All 24 qubits are initialized into $|0\rangle$ state, a layer of Hadamard gates is applied across the entire register to place every qubit equal superposition. A S gate introduces deterministic phase evolution that modifies the quantum amplitudes preserving randomness upon measurement. A DCX gates acts as control for next qubit chain allows correlations to propagate linearly through 24-qubit register. This creates deterministic, structured entanglement patterns with unavoidable hardware randomness makes the circuit suitable for PRNG. This circuit is transpiled for execution on the IBM torino quantum processor. The QPRNG circuit was executed using 50,000 shots produces output bitstring with their respective counts

.

Quantum hardware simulation of proposed QHRNG circuit

The circuit utilized 24 qubits initialized according to a predefined QTRNG binary string using Pauli-X gates to set specific qubit states. Quantum superposition was introduced using Hadamard gates, and additional phase-modulated randomness was introduced through S gates with $\pi/2$ phase shift. A chain of double CNOT gates was applied between adjacent qubit pairs to enhance entanglement and correlation among qubits. The circuit was transpiled to align with the IBM Qiskit backend physical constraints and executed with 50,000 shots to ensure statistically significant randomness. The measurement results were collected and the resulting bitstrings were aggregated by repeating each outcome according to its frequency count producing a long binary stream suitable for statistical randomness testing. This design exploits both quantum indeterminacy and circuit-level entanglement to generate quantum hybrid random numbers that can be applied in quantum cryptography and secure communication protocols.



(b)

Fig. 10. (continued)

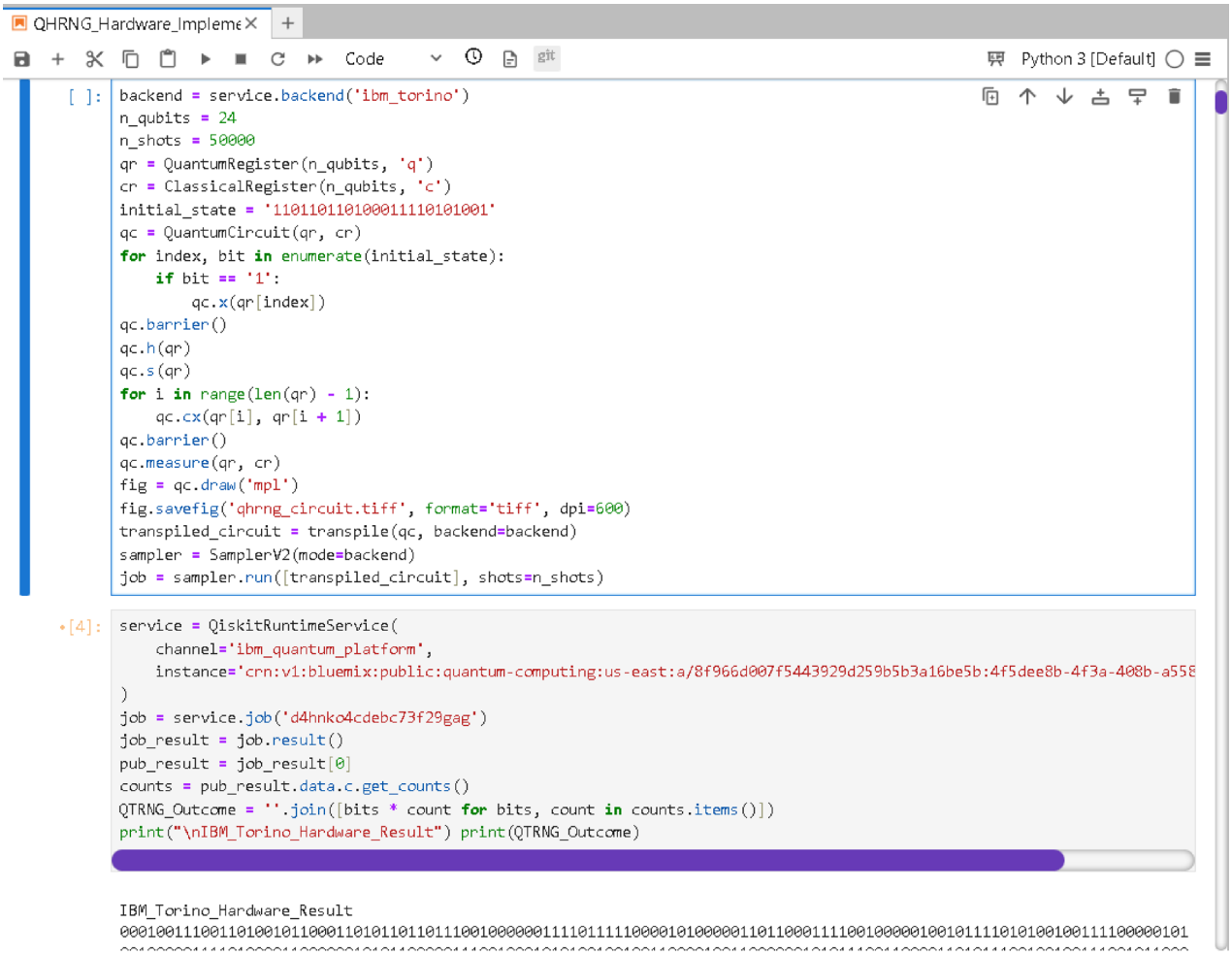
To realize a high-quality QTRNG, QPRNG and QHRNG was implemented and executed on the IBM Quantum hardware backend *ibm_torino* [Fig. 10a, b and c].

Figure 11a, b and c illustrates the execution details of a quantum job submitted through the IBM Quantum Platform, showcasing the integration of cloud-based quantum computing with user-defined quantum programs. The job executed on the ibm_torino superconducting quantum processor uses the Qiskit Runtime environment. It enables near-real-time quantum program execution by keeping quantum processing close to the hardware. The program utilized sampler for probabilistic sampling from quantum circuits.

Table 7 summarizes the execution details of a quantum job run on the *ibm_torino* backend using the sampler program showing job configuration, status, runtime duration and timestamps with a total completion time of 19 s. The reported value of 15 seconds corresponds specifically to the "Actual Qiskit Runtime Usage", as provided directly by the IBM Qiskit Runtime Service for each job. All three jobs are using the Sampler primitive and executed one Primitive Unified Block (PUB) run on the same backend (ibm_torino) identical shots configuration. The Qiskit Runtime Service internally allocates a fixed quantum runtime which results in the same reported usage time. The Total Completion Time (19 seconds) includes queuing, scheduling and overhead- so that identical. This behavior is expected and reproducible for short sampler jobs and is confirmed by the job status screenshots provided in the Fig. 11 (a), (b) and (c). The reported runtime corresponds to IBM Qiskit Runtime's actual usage metric which is quantized and identical for short sampler jobs executed under the same configuration.

Comparative analysis of hardware QRNG models

The statistical outcomes of NIST SP800-22 randomness evaluation indicates the improvement achieved through the proposed IBM Torino based quantum random number generation methods. The baseline QTRNG initially passes only 5 of the 15 tests, it already demonstrates comparable or slightly better performance than the hybrid Hadamard QRNG reported by *Salehi et al.*[32] shows that even a straightforward hardware-driven approach can achieve a foundational level of quantum randomness. When randomness conditioning and smarter sampling

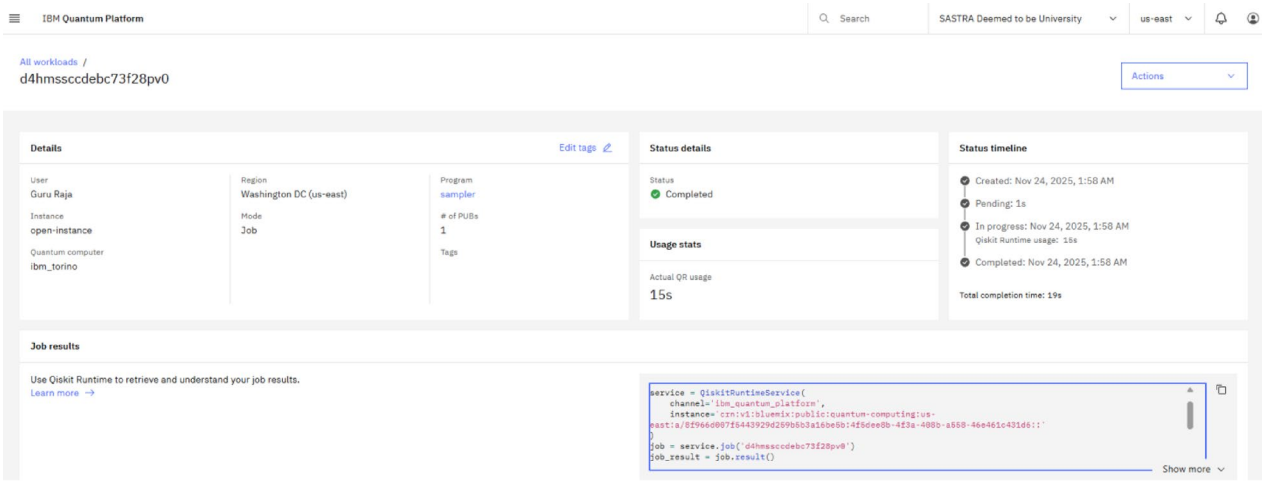


(c)

Fig. 10. (continued)

Input: $n \leftarrow$ number of qubits (24); $shots \leftarrow$ number of shots for quantum execution;
$initial\ state \leftarrow QTRNG$ generated bitstring
Output: $S_1 \leftarrow$ Concatenated random bitstring based on measurement outcomes
Require: backend ← service.backend(**"ibm_torino"**)
Initialize qr ← QuantumRegister(n), cr ← ClassicalRegister(n), ckt ← QuantumCircuit(qr,cr)
for i in range(n) **do**
  for i in range(n) **do**
    ckt.h(qr[i])
  end for
  for i in range(n-1) **do**
    ckt.crz(θ, qr[i], qr[i+1])
  end for
  ckt.barrier()
ckt.measure(qr, cr)
transpiled_circuit ← transpile(ckt, backend)
sampler ← SamplerV2(mode=backend)
job ← sampler.run([transpiled_circuit], shots=shots)
result ← job.result()
counts ← result[0].data.c.get_counts()
S1 ← ''.join([key * value for key, value in counts.items()])

Algorithm 6. Quantum true random number generation on IBM quantum hardware.



(a)

Fig. 11. (**a**) IBM Quantum Platform Job Status_QTRNG (**b**) IBM Quantum Platform Job Status_QPRNG (**c**) IBM Quantum Platform Job Status_QHRNG.

strategies are applied in QPRNG, the performance increases to 6 successful tests. This confirms that incorporating post-processing particularly methods designed to suppress system noise, gate error and bit-bias effects directly improves the statistical reliability of the quantum output. The proposed QHRNG model which successfully passes 10 NIST tests and stands out as the better performer compared to QTRNG/QPRNG/[32]/[33] in Table 8. Unlike the earlier approaches, QHRNG combines physical randomness harvesting with optimized entropy extraction and adaptive error tolerance enables it to better mitigate decoherence effects, qubit instability and device-level quantum noise. Even when compared *to Li.et al.'s* source-independent superconducting QRNG[33] which performs well with 8 passed tests and offers strong cryptographic assurance. QHRNG demonstrates superior entropy uniformity and more consistent p-value stability across multiple test categories including the Binary Matrix Rank, Linear Complexity and Random Excursion Variant tests.

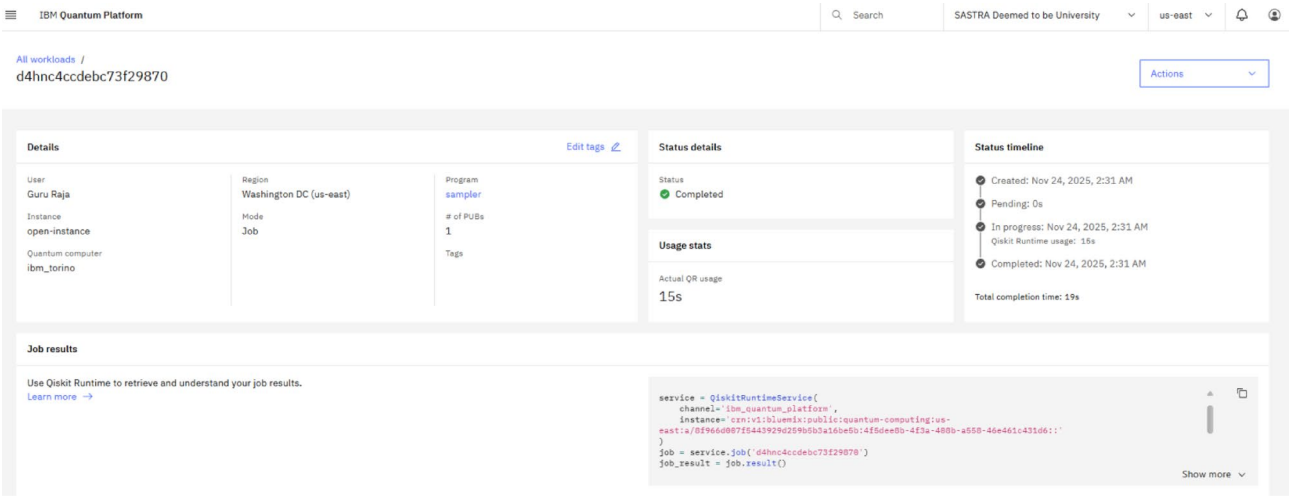
These results highlight a key trend: as the techniques evolve from raw quantum sampling (QTRNG), to noise-aware post-processing (QPRNG) and finally to hybrid conditioning and entropy amplification (QHRNG), the reliability and statistical quality of quantum randomness increase substantially. Overall, the results confirm that the proposed QTRNG, QPRNG and especially QHRNG approaches outperform existing implementations in the literature demonstrates measurable progress toward practical, secure and standards-compliant quantum random number generation.

The Table 8 reports NIST SP800-22 test outcomes derived from raw bitstreams obtained directly from real quantum hardware (ibm_torino). These results do not apply simulator-level noise suppression or entropy amplification. As a result, certain tests fail due to hardware noise, Qubit decoherence, Measurement bias and finite-shot statistical fluctuations. QHRNG passes 10 out of 15 NIST tests, outperforming: QTRNG/QPRNG/ Prior works[33] and [34]. Key cryptographic indicators such as Binary Matrix Rank, Linear Complexity, Random Excursion Variant show consistent passing behavior. Failures occur primarily in tests known to be highly sensitive to bias and sequence length (e.g., Monobit, Approximate Entropy) especially under raw quantum hardware extraction without any post-processing. The comparative performance improvement, robustness across multiple independent NIST categories and consistency of QHRNG under both hardware and simulator evaluations. The entropy estimation results [Table 9] clearly shows improvement across the proposed QRNG approaches. QTRNG shows a strong baseline performance with most entropy estimators scoring above 0.80 indicates that even the raw quantum output provides solid unpredictability. The QPRNG entropy values increase especially for Markov, LZ78Y and Tuple LRS estimators shows that correcting system noise and bias leads to more stable and uniform randomness. The highest performance is achieved with the proposed QHRNG model, where most entropy measurements reach or exceed 0.97 demonstrating strong randomness extraction and reduced predictability. When compared with existing work, the improvement becomes more evident. *Kumar's* NISQ-based approach in[34] shows fluctuating entropy levels and occasional instability reflects the challenges of using early-stage quantum hardware without advanced conditioning. The vacuum-based QRNG in[35] provides reasonable entropy performance but remains below the levels achieved by the proposed methods. Overall, as summarized in Table 9, the combination of hardware improvements and enhanced post-processing enables QTRNG, QPRNG and particularly QHRNG to outperform prior QRNG designs in terms of randomness consistency and entropy strength.

Preliminary test for proposed quantum key generation framework

Restart experiment

Restart experiment is the most preliminary test to check for repetition of same output sequence[22]. The proposed quantum random number circuit is kept executed to ensure there is no correlation between the output sequence



(b)

Fig. 11. (continued)

with the previous output. If the output sequence is getting repeated then the quantum circuit cannot be claimed as a random number generator. A Quantum Random Number Generator should produce a random output sequence that exhibits no correlation with preceding outputs.

In Fig. 12a, b and c are the results obtained from the quantum circuit are different for each execution of the proposed three secure quantum keys. For a binary sequence to be called random, the probability of occurrence of 1's and 0's needs to be equiprobable. In all the sequences obtained in the restart experiment, it can be seen that the 1s and 0s of the sequences are nearly equiprobable.

Autocorrelation

The statistical measurement was performed to check the correlation between the initial sequences and its shifted counterparts. This checks the number of bit differences between the two sequences.

$$ACT - \frac{|I - NI|}{N} \tag{33}$$

where,

"I" represents the number of identical bits between the initial sequence and the shifted counterpart.

"NI" represents the number of non-identical bits between the initial sequence and shifted counterpart.

"N" is the total number of bits in the sequence.

For a real QRNG, the non-correlated bits between the original and delayed sequence should be as minimal as possible and the highest correlation value 1 occurs only at zero-bit shift.

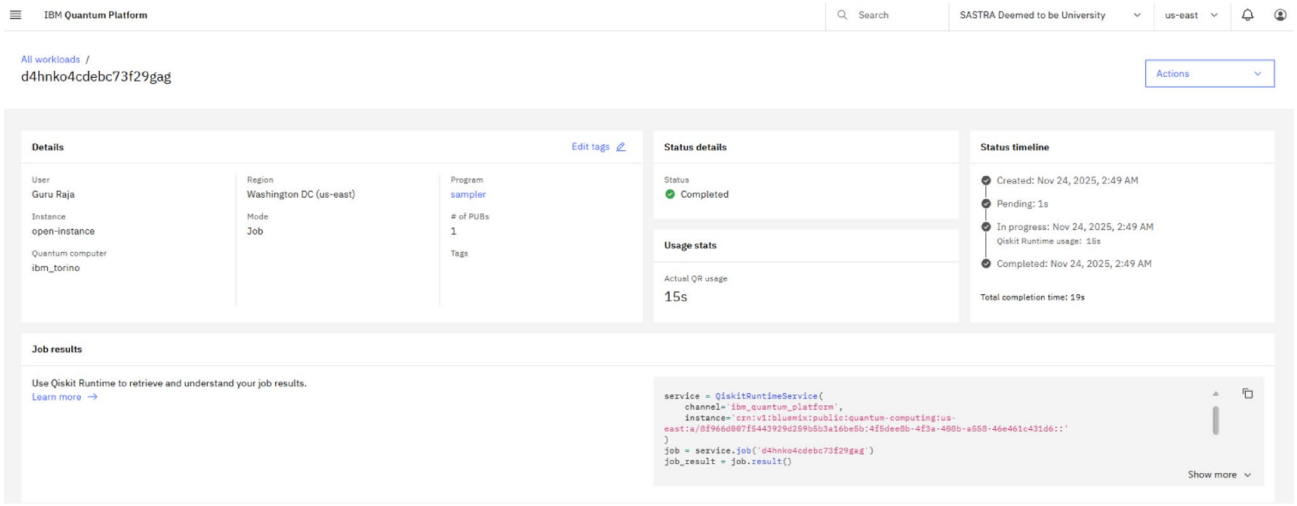
Figure 13a, b, c shows the autocorrelation analysis for 1000-bit sequence, the correlation between shifted bit sequence shows a very less value of less than 0.2 and the correlation is high (= 1) when there is a zero-bit shift. This passes the preliminary test for all the quantum key sequences.

Statistical tests for quantum random number generators

NIST 800 – 22 statistical test

NIST SP 800 – 22 test suit[23] validates true-random, pseudo-random and hybrid-random number generators in many cryptographic applications. The test suit consists of 15 tests that measure the randomness of the generated binary sequences using the proposed quantum key, as shown in Fig. 14. The proposed quantum circuit was executed on the simulator backend *'basic_simulator'* and '*aer-simulator*'. The random number sequence must have uniformity, scalability, and consistency, which is what the NIST test will test. To run this test, suit the binary sequence generated should be at least 4,00,000 in length. The Fig.14 reports simulator-based validation demonstrating the theoretical statistical strength of the proposed methods.

This Fig. 14 presents a comprehensive evaluation of QRNG using the statistical test suite. It displays the p-values obtained from various randomness tests conducted across different secure key. A p-value above the threshold of 0.01 indicates that the random sequence successfully passes the statistical test, demonstrating no significant deviation from randomness. The results indicate that QTRNG_B performs well in the Frequency Test with a p-value of 0.7552 and the Discrete Fourier Transform Spectral Test with 0.8013. QTRNG_A demonstrates strong randomness in the Linear Complexity Test with a p-value of 0.8847 and the Binary Matrix Rank Test with 0.9781. However, it underperforms in the Frequency Test with a p-value of 0.2309 and in the Cumulative Sums Reverse Test with 0.1541. QPRNG_A shows excellent statistical behaviour in the Run Test with a p-value of 0.8773, Serial Test a with 0.9587 and Serial Test b with 0.8113. QPRNG_B also achieves high p-values in Maurer's



(c)

Fig. 11. (continued)

Input: $n \leftarrow$ number of qubits (24); $shots \leftarrow$ number of shots for quantum execution;
$initial\ state \leftarrow QTRNG$ generated bitstring
Output: $S_2 \leftarrow$ Concatenated random bitstring based on measurement outcomes
Require: backend ← service.backend**("ibm_torino")**
Initialize qr ← QuantumRegister(n), cr ← ClassicalRegister(n), ckt ← QuantumCircuit(qr,cr)
for i in range(n) **do**
ckt.h(qr[i])
ckt.s(qr[i])
end for
for i in range(n − 1) **do**
ckt.dcx(qr[i], qr[i + 1])
end for
ckt.barrier()
ckt.measure(qr, cr)
transpiled_circuit ← transpile(ckt, backend)
sampler ← SamplerV2(mode=backend)
job ← sampler.run([transpiled_circuit], shots=shots)
result ← job.result()
counts ← result[0].data.c.get_counts()
S2 ← ‘’.join([key * value for key, value in counts.items()])

Algorithm 7. Quantum pseudo random number generation on IBM quantum hardware.

Universal Statistical Test with 0.9689 and Serial Test b with 0.8783. Both PRNG exhibit lower performance in the Frequency Test with QPRNG_A value of 0.0828 and QPRNG_B value of 0.0198, indicating a slightly biased distribution. The QHRNG_A delivers balanced results with high p-values in the Frequency Test within a Block at 0.8963, Run Test at 0.9304 and Non-Overlapping Template Matching Test at 0.8770. It shows some lower values in the Longest Run of Ones Test at 0.1432 and Serial Test b at 0.2518. These are still above the minimum acceptance threshold. This statistical analysis validates that the quantum hybrid random number generators deliver high entropy and strong randomness across a wide range of tests making them highly reliable for secure cryptographic applications.

NIST 800-90B statistical test

Entropy sources are essential in producing randomness in random sequences. NIST SP 800-90b test suit[24] estimates the different Min-entropy associated with the produced random sequence. Entropy is a term used to describe the sequence's uncertainty or the created binary sequence's unpredictable nature. High levels of uncertainty and unpredictability are indicated by an entropy value close to 1. The NIST SP 800-90B test suite provides two categories of entropy estimators: Independent and Identically Distributed (IID) data and non-IID data, accommodating more complex dependencies and structures within the input sequence. In the IBM Qiskit framework, the proposed quantum circuits are executed using the '*basic_simulator*' and '*aer_simulator*' backend

Input: $\boldsymbol{n} \leftarrow$ number of qubits (24); $shots \leftarrow$ number of shots for quantum execution;
$initial\ state \leftarrow QTRNG$ generated bitstring
Output: $S_3 \leftarrow$ Concatenated random bitstring based on measurement outcomes
Require: backend ← service.backend**("ibm_torino")**
Initialize qr ← QuantumRegister(n), cr ← ClassicalRegister(n), ckt ← QuantumCircuit(qr,cr)
for i in range(n) **do**
  if initial_state[i] == '1' **then**
    ckt.x(qr[i])
  end if
end for
ckt.barrier()
for i in range(n) **do**
ckt.h(qr[i])
ckt.s(qr[i])
end for
for i in range(n − 1) **do**
  ckt.dcx(qr[i], qr[i + 1])
end for
ckt.barrier()
ckt.measure(qr, cr)
transpiled_circuit ← transpile(ckt, backend)
sampler ← SamplerV2(mode=backend)
job ← sampler.run([transpiled_circuit], shots=shots)
result ← job.result()
counts ← result[0].data.c.get_counts()
S_3 ← ''.join([key * value for key, value in counts.items()])

Algorithm 8. Hybrid quantum random number generation on IBM quantum hardware.

Field	QTRNG	QPRNG	QHRNG
Mode	Job	Job	Job
QPU Name	ibm_torino	ibm_torino	ibm_torino
Instance	ibm-q/open/main	ibm-q/open/main	ibm-q/open/main
Job_ID	d4hnc4ccdebc73f29870	d4hmssccdebc73f28pv0	d4hn04cdebc73f29gag
Program	Sampler	Sampler	Sampler
# of PUBs	1	1	1
Status	Completed	Completed	Completed
Actual Usage	15 s	15 s	15 s
Created	Nov 24, 2025, 2:31 AM	Nov 24, 2025, 1:58 AM	Nov 24, 2025, 2:49 AM
Pending Time	1 s	1 s	1 s
In Progress	Nov 24, 2025, 2:31 AM	Nov 24, 2025, 1:58 AM	Nov 24, 2025, 2:49 AM
Qiskit Runtime Usage	15 s	15 s	15 s
Completed	Nov 24, 2025, 2:31 AM	Nov 24, 2025, 1:58 AM	Nov 24, 2025, 2:49 AM
Total Completion Time	19 s	19 s	19 s

Table 7. Qiskit runtime job Summary.

with 20,000 shots to generate the corresponding random number sequence. The retrieved binary sequence is transformed into an ASCII file with the binary file extension. The Ubuntu platform uses the test suit, where the ASCII files are executed for individual non-IID tests. The source ASCII file must be at least 1GB in size. For non-IID test, the entropy estimators measure the Min-Entropy per 1-bit sample. The min-entropy is measured as,

$$Min - Entropy = \min_{1 \le i \le k} \left(\log_2 \left(p_i \right) \right) \tag{34}$$

where p_i is the probabilities of each possible outcome i.

$log_2 \left(p_i \right)$ is the logarithm base 2 of the probability p_i.

In information theory, log base 2 is used to measure in bits. For most of the Entropy estimators exhibit good performance, as evidenced by results greater than 0.8 per 1 bit. The value of the T-Tuple test estimate shows the less entropy of 0.086/1bit. Overall, for all test instances, the average min-entropy for both suggested methodology is more than 0.7 per 1 bit as indicated in Fig. 15. Min-Entropy results for all other entropy estimation tests.

S. No.	NIST SP800-22 Test	QTRNG	QPRNG	QHRNG	[32]	[33]
1	Frequency (Monobit) Test	0.0000 (F)	0.0000 (F)	0.0000 (F)	0.0000 (F)	**0.4340 (P)**
2	Block Frequency Test	0.0000 (F)	0.0000 (F)	**0.0710 (P)**	0.0000 (F)	**0.7541 (P)**
3	Runs Test	0.0000 (F)	0.0000 (F)	0.0000 (F)	0.0000 (F)	**0.1202 (P)**
4	Longest Run of Ones in a Block Test	0.0000 (F)	0.0000 (F)	**0.3433 (P)**	0.0000 (F)	**0.6536 (P)**
5	Binary Matrix Rank Test	**0.6659 (P)**	**0.0315 (P)**	**0.9106 (P)**	0.0000 (F)	**0.6537 (P)**
6	Discrete Fourier Transform Test	0.0000 (F)	**0.1024 (P)**	**0.9196 (P)**	0.0000 (F)	**0.9150 (P)**
7	Overlapping Template Matching Test	0.0000 (F)	0.0000 (F)	**0.4916 (P)**	**0.9997 (P)**	-
8	Non-Overlapping Template Matching Test	0.0000 (F)	0.0000 (F)	0.0000 (F)	**0.8576 (P)**	-
9	Maurer's Universal Test	0.0000 (F)	**0.0591 (P)**	**0.4806 (P)**	0.0000 (F)	-
10	Linear Complexity Test	**0.3463 (P)**	**0.6329 (P)**	**0.6217 (P)**	**0.4210 (P)**	-
11	Serial Test	**0.0133 (P)**	0.0172 (F)	**0.2197 (P)**	**1.0000 (P)**	-
12	Approximate Entropy Test	0.0000 (F)	0.0000 (F)	0.0000 (F)	**1.0000 (P)**	**0.1624 (P)**
13	Cumulative Sums Test	0.0000 (F)	0.0000 (F)	0.0000 (F)	0.0000 (F)	**0.7827 (P)**
14	Random Excursion Test	**0.6075 (P)**	**0.6609 (P)**	**0.5509 (P)**	0.0000 (F)	-
15	Random Excursion Variant Test	**0.7080 (P)**	**0.4530 (P)**	**0.5477 (P)**	0.0000 (F)	-
No. of. Passed NIST Test Results		**5**	**6**	**10**	**5**	**8**

Table 8. Performance comparison of IBM Torino quantum hardware based QRNG NIST SP800-22 randomness evaluation: Significant values are in bold.

S. No	Type of Entropy Estimators	QTRNG	QPRNG	QHRNG	[34]	[35]
1.	Collision	0.8104	0.9171	0.7953	1.0000	0.67
2.	Compression	0.6856	0.8028	0.7299	0.5612	0.73
3.	Conditioned	0.6754	0.7259	0.9859	-	0.73
4.	Markov	0.8695	0.9722	0.7285	0.8502	0.58
5.	Lag prediction	0.7999	0.7747	0.9735	0.8191	0.80
6.	Lz78y	0.8757	0.9659	0.7284	0.8292	0.80
7.	Most common	0.8758	0.9658	0.9733	0.8292	0.79
8.	Multi markov	0.8757	0.9661	0.9737	0.0575	0.80
9.	Multi most common	0.8774	0.9759	0.9868	0.8292	0.80
10.	Ttuple Lrs	0.9851	0.9907	0.9081	0.0001	0.75

Table 9. Performance comparison of IBM Torino quantum hardware based QRNG NIST SP800-90B entropy Estimation.

Figure 15 represents the performance of various QRNG methods using the Qiskit BasicSimulator and Qiskit AerSimulator based on the NIST SP 800-90B entropy estimation tests. The entropy estimators listed on the y-axis include statistical tests like Most Common Value, Collision, Markov, Compression, T-Tuple, Multi-most Common in Window, Lag Prediction, LZ78Y Prediction, and Multi MMC Prediction. Each value in Fig. 15 represents the minimal entropy estimate of the respective quantum random sequences. The higher entropy values (closer to 1) indicate better randomness and unpredictability. From Fig. 15, QPRNG_B and QTRNG_B achieve perfect scores (1.000) in the Collision and Multi MMC Prediction tests, indicating strong performance in avoiding repeated values and patterns. Markov and LZ78Y estimate that memory and compressibility also remain high across methods, suggesting minimal redundancy in the generated bits. The compression test shows the lowest entropy values (0.5183 in QPRNG_A), implying that some structures remain in the bitstream. The T-Tuple test detects recurring n-bit patterns that consistently score 0.0860 for all methods, highlighting a conservative lower-bound estimation. Overall, the results indicate that the quantum hybrid methods produce high-quality randomness while the quantum pseudo methods show slightly reduced entropy in certain tests. Figure 15 shows that empirical evidence supports the effectiveness of the proposed QRNG in making secure and statistically random sequences.

Performance metrics

Differential attack analysis

Number of pixel change rate (NPCR)

NPCR evaluates an encryption algorithm's sensitivity to slight variations in the original image by measuring the impact of single-pixel alterations. It calculates the proportion of pixels in the encrypted image that are affected by this change in the original image. The robustness of image encryption methods against differential attacks is frequently evaluated using this metric.

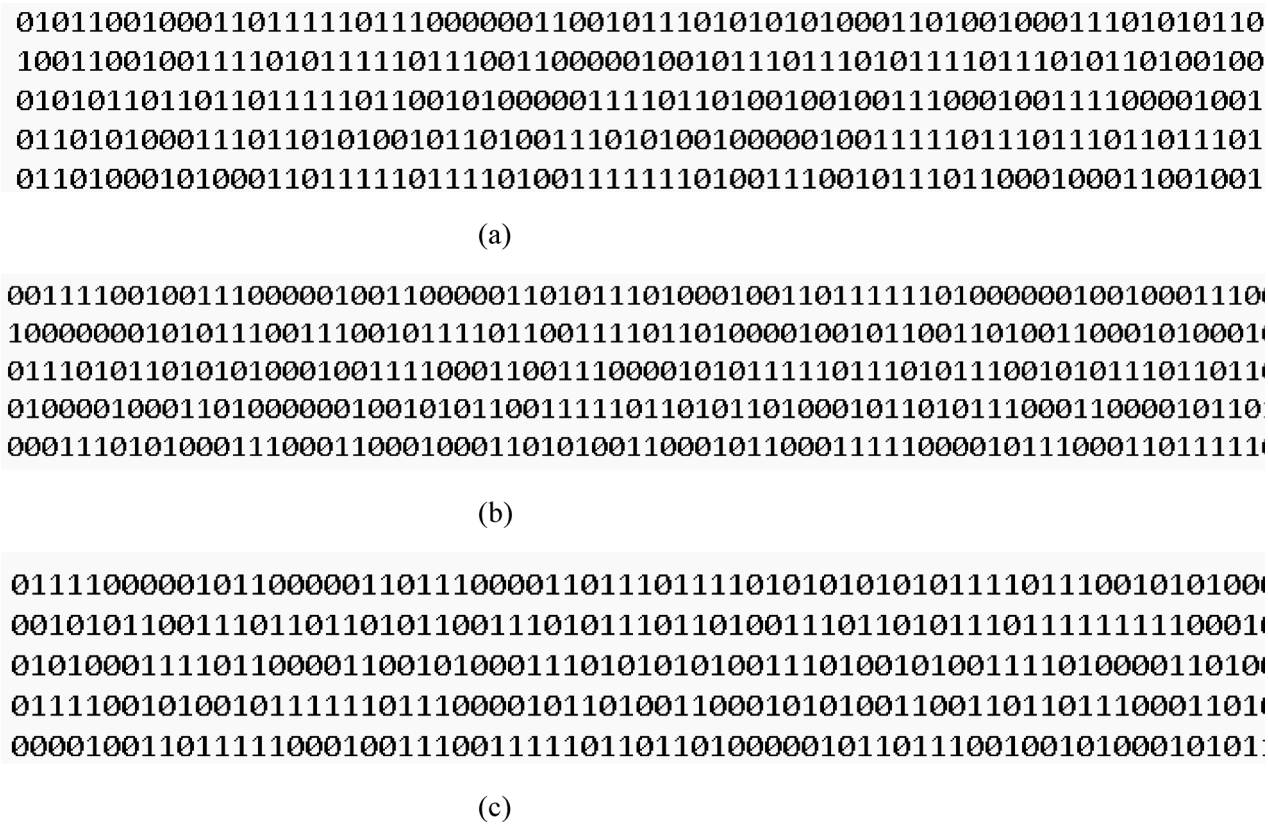


Fig. 12. Restart Experiment analysis for quantum key: (**a**) QTRNG Sequence (**b**) QPRNG Sequence (**c**) QHRNG Sequence.

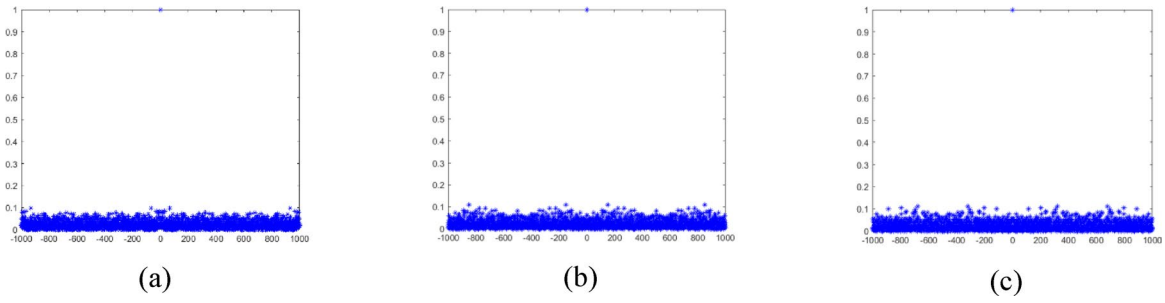


Fig. 13. Autocorrelation analysis (**a**) QTRNG (**b**) QPRNG (**c**) QHRNG.

$$NPCR = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} F(i,j)}{M \times N} \times 100$$
$$F(i,j) = \left\{ \begin{array}{l} 0, \ if \ E_1(i,j) = E_2(i,j) \\ 1, \ if \ E_1(i,j) \neq E_2(i,j) \end{array} \right\} \quad (35)$$

where,

m and n represents the number of rows and columns of the image.

$F(i,j)$ represents binary comparison function.

$E_1(i,j)$ and $E_2(i,j)$ represents the pixel value located at position (i,j) within two separately encrypted images derived from original image and slightly modified image.

The NPCR values for all proposed encrypted images are consistently close to the ideal value of 99.56%, indicating that all quantum-based keys demonstrate high sensitivity to variations in the original image and that even minimal changes lead to substantial alterations in the encrypted output. This characteristic enhances the algorithm's robustness against differential attacks. The Barbara image range between 99.58% (QTRNG_B) and 99.62% (QPRNG_B) as in Fig. 16. These results show slight variations depending on the random key

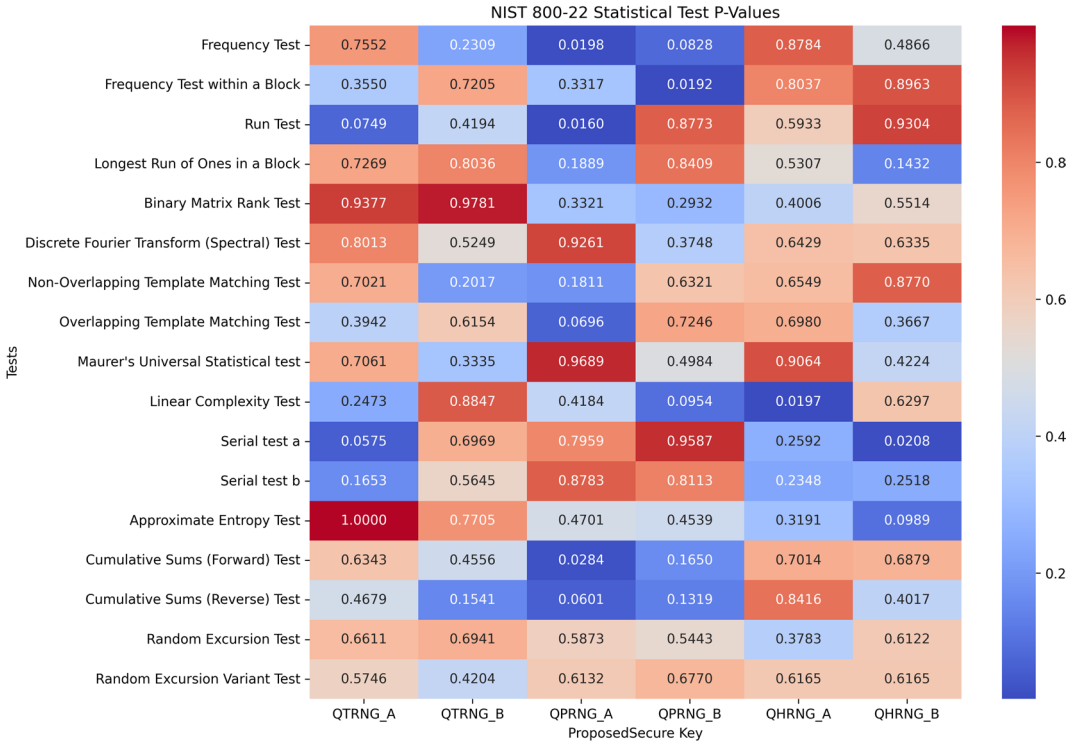


Fig. 14. NIST SP800-22 Statistical test for proposed simulator QRNG.

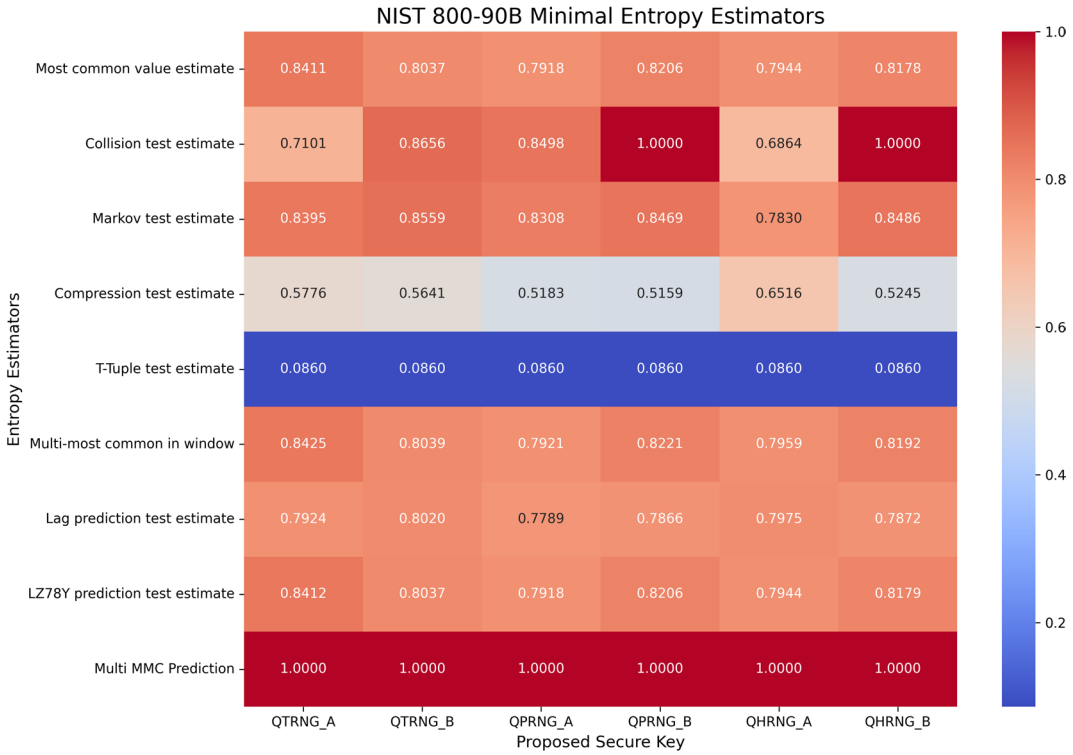


Fig. 15. NIST SP800-90B min-entropy estimator for proposed simulator QRNG

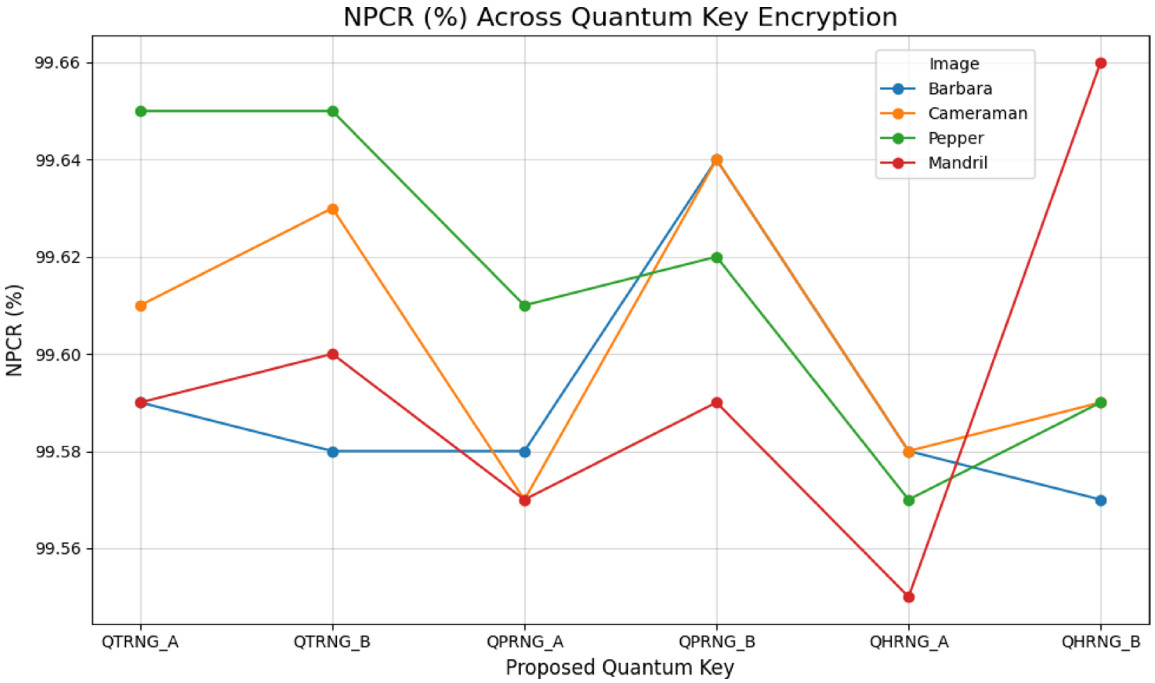


Fig. 16. NPCR analysis for Image dataset across Quantum key encryption.

Test Images	Proposed	Ref.[2]	Ref.[26]	Ref.[27]	Ref.[28]
Barbara	99.5697	99.52	99.6281	-	99.5438
Cameraman	99.5895	99.53	99.6124	99.6123	99.5804
Pepper	99.5895	99.57	99.6233	99.6075	99.5300
Mandril	99.6612	99.55	-	99.6079	-

Table 10. NPCR (%) comparative analysis of proposed quantum key encryption.

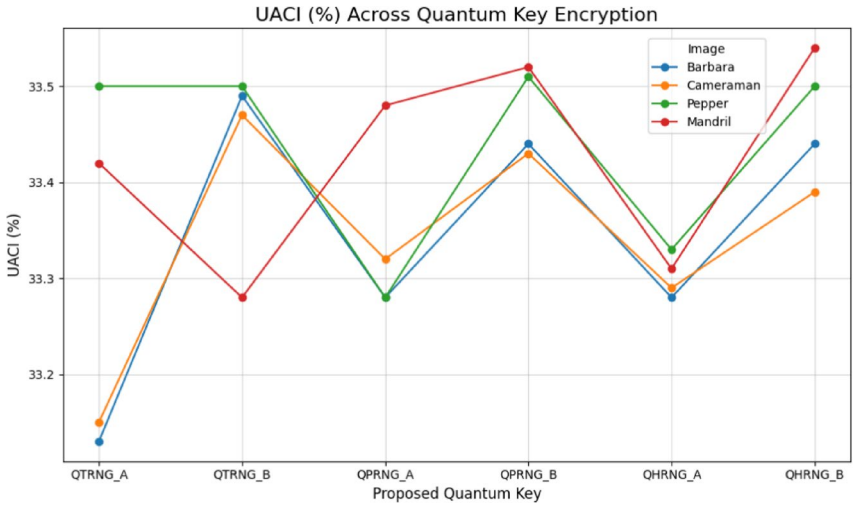


Fig. 17. UACI analysis for Image dataset across Quantum key encryption.

and simulator type; all values indicate a high degree of pixel sensitivity. The noiseless simulator (B) generally outperforms the noisy Aer simulator (A), producing slightly higher NPCR values for Barbara. The Cameraman image range from 99.58% (QHRNG_A) to 99.62% (QPRNG_B). The QPRNG-based encryption scheme demonstrates slightly better performance compared to QTRNG and QHRNG. This indicates that pseudo-random

Test Images	Proposed	Ref. [2]	Ref. [26]	Ref. [27]	Ref. [28]
Barbara	33.4363	33.41	33.5254	-	33.3799
Cameraman	33.3946	33.47	33.5018	33.5685	33.5685
Pepper	33.5026	33.40	33.5032	33.5291	33.5291
Mandril	33.5394	33.51	-	33.3799	-

Table 11. UACI (%) comparative analysis of proposed quantum key encryption.

Test Images	Key_1	Key_2	Key_3
Barbara	50.0120	49.9717	49.9727
Cameraman	50.0347	50.1014	49.9675
Pepper	50.0301	50.1218	49.9855
Mandril	49.9950	49.8199	50.1548

Table 12. Avalanche effect with different quantum Keys.

keys can provide robust encryption while maintaining computational efficiency. The Pepper image range from 99.60% (QPRNG_A) to 99.66% (QHRNG_B). The mandril image ranging from 99.56% (QTRNG_A) to 99.66% (QHRNG_B) as in Table 8. The highest NPCR value observed for this image comes from the hybrid random key generator (QHRNG_B), which combines the strengths of true and pseudo-randomness. This highlights the effectiveness of QHRNG enhances encryption robustness for images with more complex textures.

Table 10. shows that the proposed quantum encryption method achieves the highest NPCR values 99.66% across test images indicating superior resistance to differential attacks. It consistently outperforms methods and demonstrating enhanced sensitivity to pixel changes and stronger encryption reliability.

Unified average changing intensity (UACI)

UACI metric quantifies the average difference in pixel intensities between the original and encrypted image.

$$UACI = \frac{1}{X \times Y} \sum_{i=1}^{X} \sum_{j=1}^{Y} \frac{|E_1(X,Y) - E_2(X,Y)|}{255} \times 100 \tag{36}$$

where, $|E_1(X,Y) - E_2(X,Y)|$ represents the absolute difference between the original and encrypted images.

The factor 255 normalizes the pixel intensity difference for 8-bit grayscale images (pixel values ranging from 0 to 255).

The UACI values across all proposed quantum keys and images range between 33.2% and 33.5%, close to the theoretical ideal value. This indicates a high degree of intensity variation introduced by this proposed encryption. The Barbara image shows UACI values ranging from 33.2% (QTRNG_A) to 33.4% (QPRNG_B) as shown in Fig. 17. The noiseless basic simulator (B) performs slightly better than the noisy Aer simulator (A) in generating more significant intensity variations. The Cameraman image varies from 33.2% (QHRNG_A) to 33.4% (QTRNG_B). This indicates that all quantum random key types provide similar encryption strength for simpler images like Cameraman, where grayscale variations are limited. The Pepper image demonstrates higher UACI values, with the range being 33.3% (QPRNG_A) to 33.5% (QHRNG_B). The QHRNG scheme achieves the best results for this image due to its balanced mix of true and pseudo-randomness, enhancing encryption for moderate texture complexity. The Mandril image, known for its intricate textures, displays a wide variation ranging from 33.2% (QTRNG_A) to 33.5% (QHRNG_B) as shown in Table 11. The hybrid random key generator (QHRNG_B) achieves the highest UACI value, indicating its superior ability to disrupt pixel intensity correlations in highly detailed images.

Table 11 indicates a stronger sensitivity to slight changes in the image and highlights the proposed method's enhanced diffusion capability and resistance against differential attacks. The high UACI values greater than 33.39% reflect effective encryption performance essential for robust image security.

Avalanche effect (AE)

The avalanche effect ensures that even a minor modification in the input image or encryption key results in substantial changes in the encrypted quantum state. This enhances security by making it extremely difficult for an attacker to predict the relationship between the original and encrypted data, thereby resisting differential attacks. Table 12 demonstrates that the proposed quantum encryption scheme responds well to small key changes with values consistently near 50% across different images.

$$AE = \left(\frac{C_1 \oplus C_2}{n}\right) \times 100\% \tag{37}$$

where, C_1 is Ciphertext of the original image and.

C_2 is Ciphertext after a one-bit change in the image.

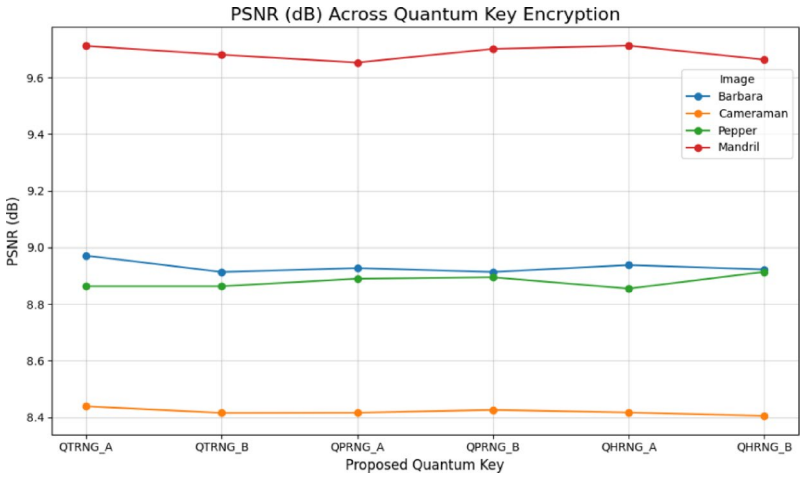


Fig. 18. PSNR analysis for Image dataset across Quantum key encryption.

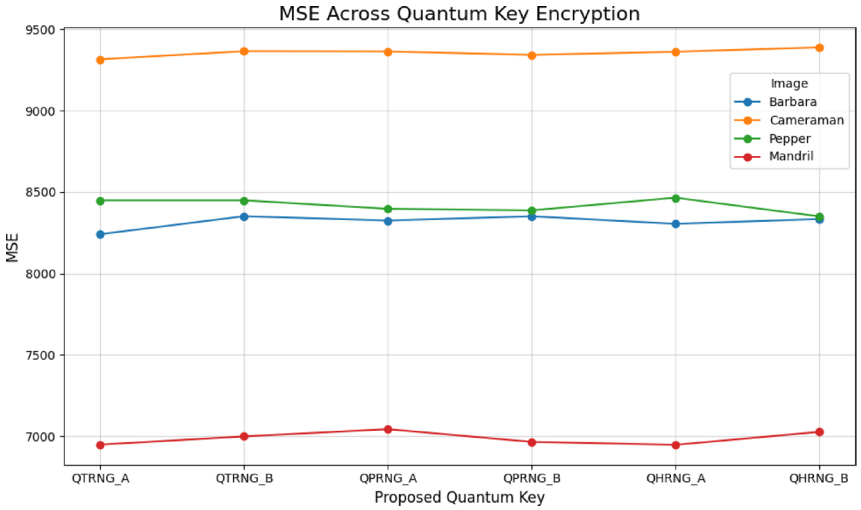


Fig. 19. MSE analysis for Image dataset across Quantum key encryption.

$\oplus$ is Bitwise XOR operation.
n is Total number of bits in the ciphertext.

Robustness to noise and lossy compression

Peak Signal-to-Noise ratio (PSNR)

PSNR is a quantitative measure used to assess the distortion between the original and encrypted images. It is expressed in decibels (dB), where a lower PSNR indicates higher encryption strength due to greater dissimilarity.

$$PSNR = 10.\, log_{10}\left(\frac{I_{MAX}^2}{MSE}\right) \tag{38}$$

where,

I_{MAX}^2 represents maximum pixel value (typically 255).

MSE is the Mean squared error between the original and encrypted image.

Mandril achieves the highest PSNR values above 9.6 dB regardless of the quantum random number generator. This suggests that the encryption system maintains a high-quality original image reconstruction due to its highly detailed and textured nature. It may inherently resist excessive intensity distortions, preserving the image quality post-encryption. Cameraman and Barbara show moderate PSNR values (8.8–9.0 dB), (8.4–8.6) with slightly lower values observed for QTRNG_B and QHRNG_B as in Fig. 18. The simple structure and large uniform regions of the Cameraman's image make it more susceptible to significant intensity variations during encryption, resulting in slightly lower PSNR. The noiseless simulations (A) slightly improve the PSNR, indicating that reducing noise helps preserve image quality better for simpler images. Pepper maintains stable

PSNR values (8.8–9.0 dB) across all quantum random number generators. Its balanced complexity allows the encryption to distribute intensity variations evenly, resulting in consistent quality across all methods. Both true randomness (QTRNG) and hybrid randomness (QHRNG) perform equally well for this image, showcasing their versatility in handling moderately complex textures. Despite the lower PSNR, the consistency across all methods (QTRNG, QPRNG, and QHRNG) and simulators suggest that the encryption system is robust in handling grayscale variations.

Mean squared error (MSE)
MSE quantifies the average squared difference between the corresponding pixel intensities of the original image and the encrypted image. A higher MSE value indicates greater dissimilarity and better encryption strength.

$$MSE = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} [O(x,y) - E(x,y)]^2 \tag{39}$$

where,
$O(x,y)$ is a pixel value in the original image.
$E(x,y)$ is a pixel value in the encrypted image.
M, N is a number of rows and columns in the image.

The MSE values across proposed quantum keys range from 7000 (Mandril) to 9500 (Cameraman). This spread reflects the encryption system's ability to introduce varying randomness and intensity differences depending on the image texture and the quantum random number generation scheme as in Fig. 19. MSE for Barbara remains around 8500 showing consistent encryption results across all quantum random number generators. The minor variability between noisy and noiseless simulators suggests that Barbara's smooth grayscale variations are robustly disrupted, making the encryption equally effective for all methods. Cameraman exhibits the highest MSE values peaking at 9500, particularly for QTRNG_B and QHRNG_B. This indicates Cameraman's simple structure and well-defined regions which allow quantum keys to induce large-intensity changes effectively. Pepper image lies around 8300–8500, showcasing moderate differences introduced by the encryption. The close values for all methods suggest that Pepper's balanced complexity (edges and textures) benefits equally from both true randomness (QTRNG) and hybrid approaches (QHRNG). The Mandril image consistently shows the lowest MSE values (7000–7500) regardless of the quantum random key generator or simulator. This reflects the inherent challenge in encrypting highly detailed and textured images where intensity variations are harder to disrupt uniformly.

Entropy analysis

The information entropy describes the degree of randomness in an encryption system by using the probability of gray value. It is used to evaluate the strength of an encryption scheme. For an image, the information entropy (IE) can be defined as

$$IE = -\sum_{g=0}^{255} P(g) \log_2 P(g) \tag{40}$$

where P (g) represents the probability of gray level g. If the encrypted image is truly random and then the ideal value of entropy should be 8.

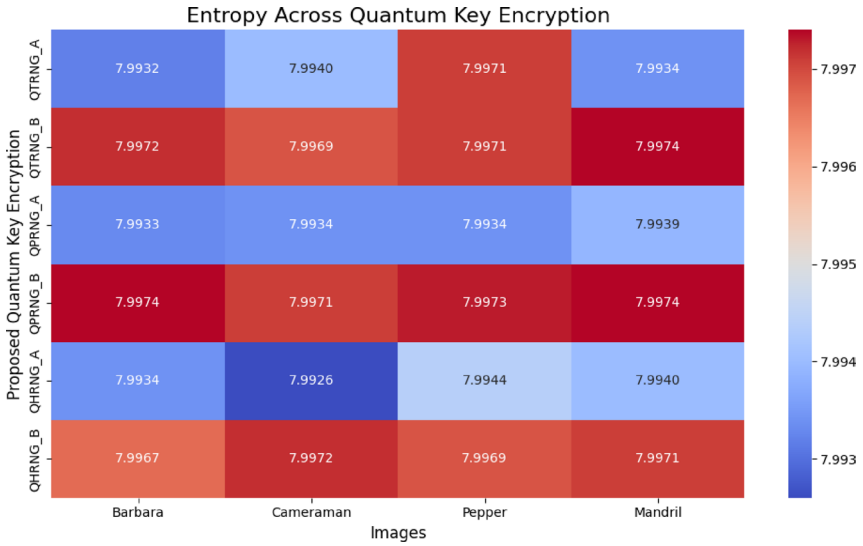


Fig. 20. Information Entropy for Image dataset across Quantum key encryption.

Test Images	Proposed	Ref. [2]	Ref. [26]	Ref. [27]	Ref. [28]
Barbara	7.9967	7.9900	7.9997	-	7.9971
Cameraman	7.9972	7.9006	7.9993	7.9993	7.9970
Pepper	7.9969	7.9894	7.9993	7.9993	7.9973
Mandril	7.9971	7.5422	-	7.9993	-

Table 13. Information entropy comparative analysis of proposed quantum key encryption.

The entropy values across all images and quantum random number generators (QTRNG, QPRNG, and QHRNG) remain consistently high, ranging from 7.9934 to 7.9974, as shown in Fig. 20. High entropy values close to the theoretical maximum (8 for an 8-bit grayscale image) indicate that the encryption system produces highly randomized encrypted images, ensuring robust security. The differences between entropy values are minimal, suggesting that all quantum key types perform effectively regarding randomness and information distribution. Barbara's image shows entropy values ranging from 7.9932 to 7.9974. The smooth grayscale variations in Barbara make it slightly more sensitive to the encryption process as seen in the small fluctuations in entropy. QPRNG_B achieves the highest entropy (7.9974) indicating the most uniform randomness for this image under pseudo-random encryption. The cameraman's entropy values range from 7.9926 to 7.9971. The simple structure and large uniform regions result in slightly lower entropy values than more complex images like Mandril. QHRNG_A achieves the lowest entropy (7.9926), while QPRNG_B and QHRNG_B yield higher values (7.9972), showcasing better randomness in noiseless environments. Pepper maintains entropy values between 7.9934 and 7.9973. Its balanced complexity results in consistent entropy values across all quantum random number generators. QPRNG_B and QHRNG_B again achieve the highest entropy (7.9973), reflecting their effectiveness in enhancing randomness for moderately complex images. Mandril achieves entropy values ranging from 7.9939 to 7.9974, the highest among all images, as in Table 13. Mandril's highly detailed and textured nature allows the encryption system to distribute information more uniformly, resulting in consistently high entropy values. Both QPRNG_B and QHRNG_B reach the maximum entropy (7.9974), highlighting their strong performance for complex images.

The comparative analysis of information entropy in Table 13 shows that the proposed quantum key encryption method achieves near-ideal entropy values close to 8 for all test images.

Statistical analysis

Correlation coefficient (CC)

The correlation coefficient (CC) quantifies the degree of linear relationship between adjacent pixel intensities in a grayscale image. It is calculated for three directions: horizontal, vertical and diagonal. A low correlation coefficient between adjacent pixels indicates effective encryption.

$$r = \frac{\sum_{i=0}^{255}\left(P_i - \bar{P}\right)\left(Q_i - \bar{Q}\right)}{\sum_{i=0}^{255}\left(P_i - \bar{P}\right)^2 \bullet \sum_{i=0}^{255}\left(Q_i - \bar{Q}\right)^2} \tag{41}$$

where, P_i and $Q_i$represents the grayscale intensity values of two adjacent pixels.

$\bar{P}$ and $\bar{P}$ represents the mean intensity values of the pixel.

The correlation values range from − 0.006 to 0.006, showing small variations across images and quantum keys as shown in Fig. 21a, b, c. Ideal encryption systems aim to minimize correlations in encrypted images, disrupting the original pixel dependencies effectively. Negative correlation values indicate a strong disruption of horizontal dependencies which is desirable for robust encryption. The differences between noiseless and noisy simulators highlight the impact of added randomness on correlation values with noisy simulators generally introducing more variability.

The Fig. 22 presents a correlation analysis of adjacent pixel values in the Barbara image before and after encryption using a HRNG method. The correlation is assessed in three directions: horizontal, vertical and diagonal. The original image correlation demonstrates high linear correlation among neighboring pixels, a characteristic feature of natural images. Specifically, the strong clustering along the diagonal line indicates the intensity of a pixel is highly dependent on its adjacent pixel in the specified direction. These encrypted images exhibit a completely random and uniform distribution of pixel values across all three directional analyses. The absence of any discernible pattern or clustering indicates the encryption process has effectively destroyed the statistical correlation present in the original image. This result confirms that the encryption algorithm introduces significant diffusion and a desirable property in cryptographic systems by ensuring the pixel values in the encrypted image are statistically independent. The uniformity in the encrypted scatter plots validates the encryption's strength in resisting statistical and correlation-based attacks. The utilization of quantum entropy sources in the key generation process contributes to the encryption's randomness and unpredictability, enhancing overall image security.

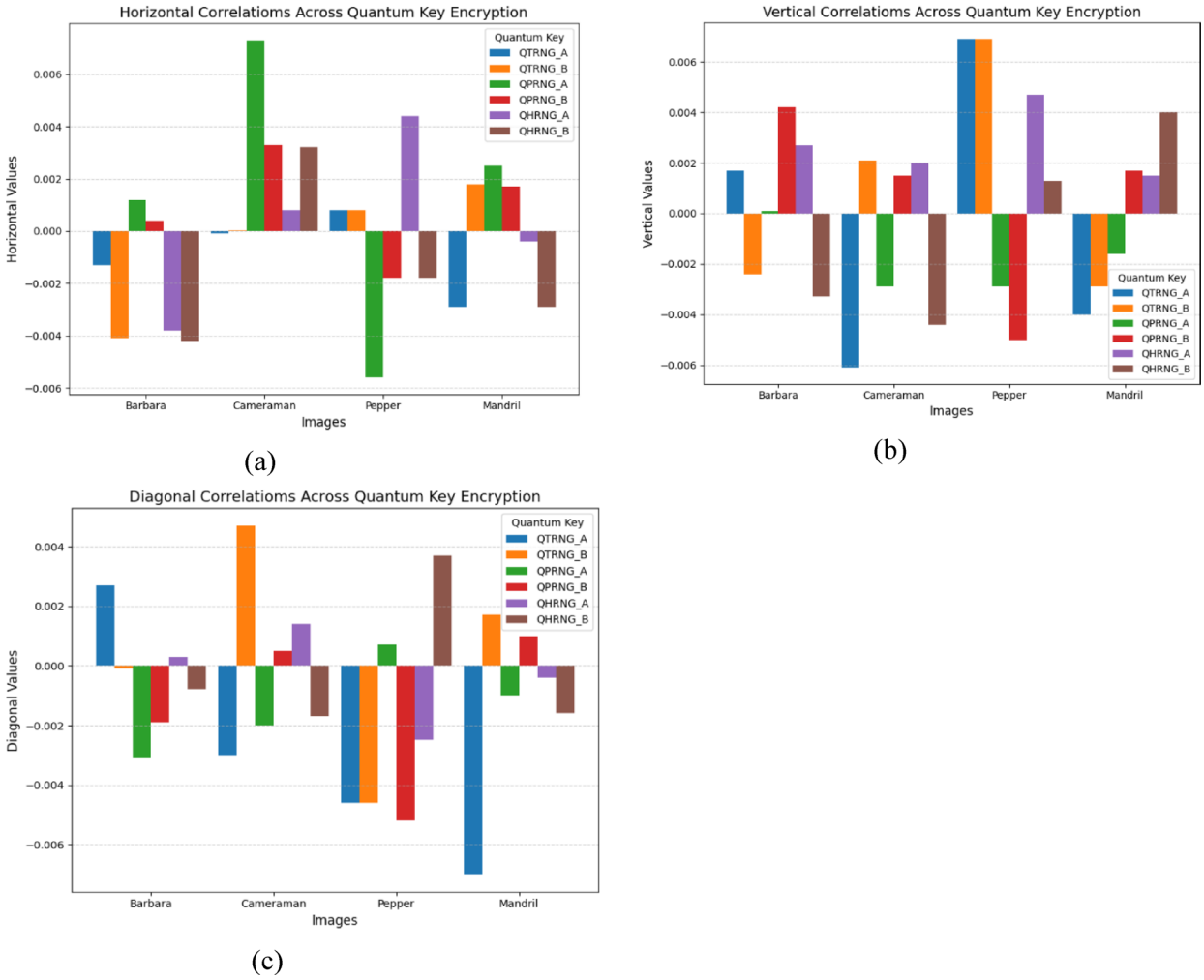


Fig. 21. Correlation Coefficient (CC) (**a**) horizontal CC (**b**) vertical CC (**c**) diagonal CC across quantum key encryption.

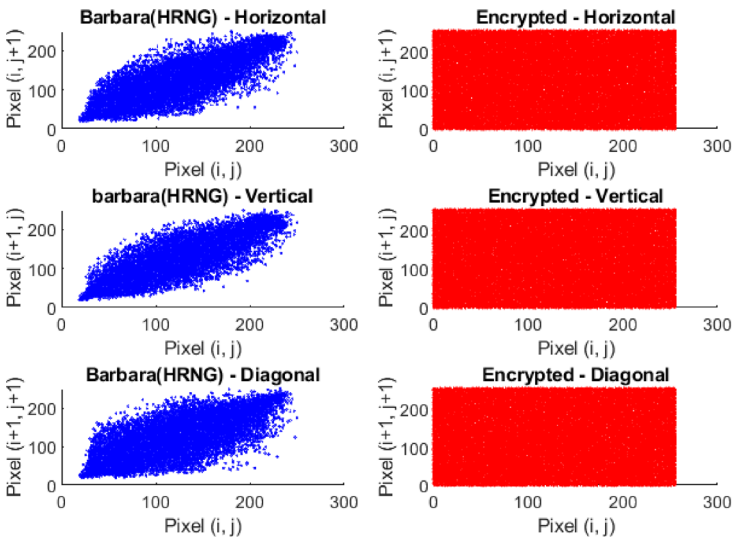


Fig. 22. Adjacent pixel correlation analysis.

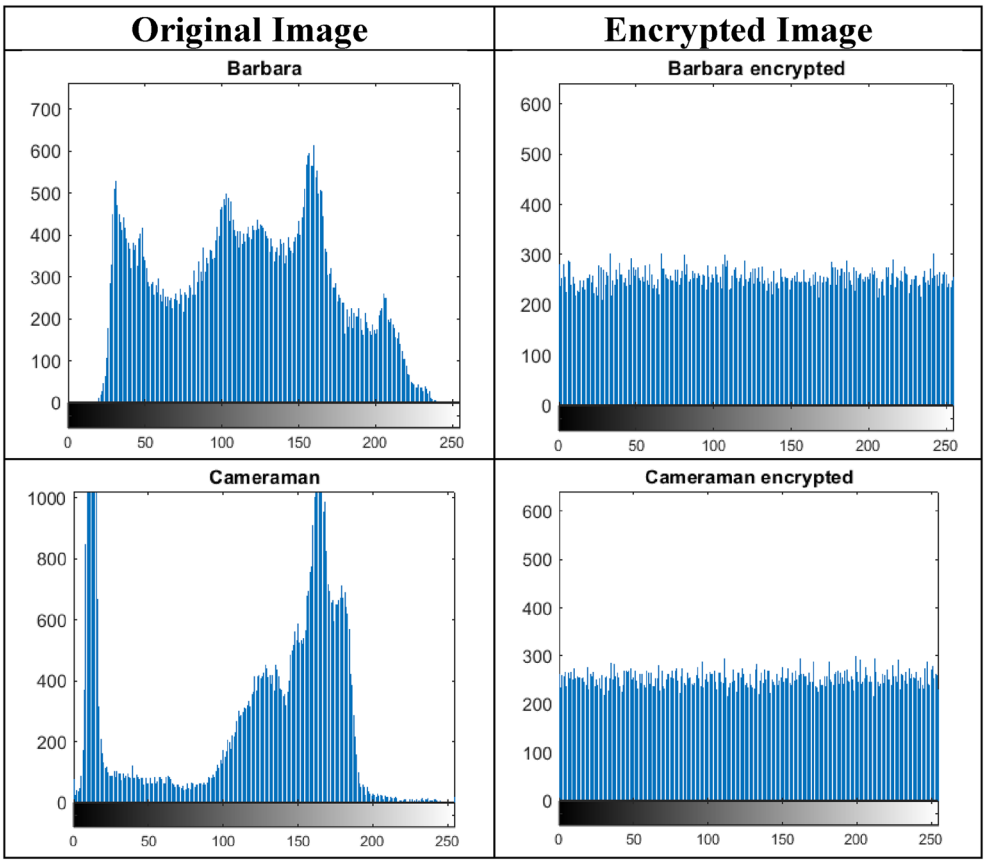


Table 14. Histogram of original and measured encrypted image.

Test Images	Chi-square statistic	Critical Value
Barbara	0.4201	293.2478
Cameraman	1.5454	293.2478
Pepper	0.4921	293.2478
Mandril	0.8488	293.2478

Table 15. Pixel distribution analysis using Chi-square test.

Histogram analysis

A histogram is the frequency distribution of pixel intensity values in an image. It shows how many pixels in the image have a particular intensity level. For grayscale images, intensity values range from 0 (black) to 255 (white). A uniform histogram in an encrypted image suggests high randomness and reduced predictability, making it secure against statistical attacks.

$$H(i) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \delta\ (I(x,y) - i) \tag{42}$$

where, $I(x,y)$represents the intensity of a pixel at position (x,y) in an image size $M \times N$.

$\delta\ (n)$ is a Kronecker delta function.

'i' represents an intensity value of the pixel.

Table 14 represents histogram analysis comparing the intensity distributions of original and encrypted images for the standard test images "Barbara" and "Cameraman". The original image histogram shows highly non-uniform distributions with prominent peaks and valleys, indicating strong correlation and redundancy among pixel intensity values. These patterns reflect the structural and textural content of the images are exploitable through statistical attacks. In encrypted images, the histograms exhibit an almost flat and uniform distribution across the entire grayscale range (0–255). This uniformity suggests that the pixel values have been thoroughly randomized through the quantum hybrid encryption process, it removes identifiable structures and pattern. The intensity values are nearly equiprobable, it is a key characteristic of a well-encrypted image. This makes it computationally infeasible for attackers to deduce any information about the original image using histogram-

Test Images	Deviation from the expected distribution
Barbara	0.0521
Cameraman	0.0496
Pepper	0.0517
Mandril	0.0520

Table 16. Deviation analysis for encryption quality test.

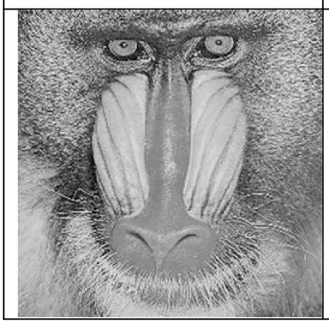
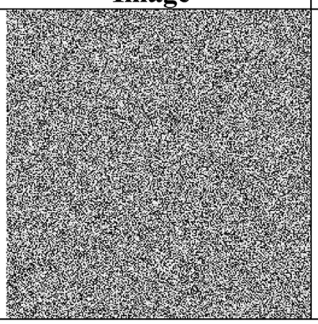
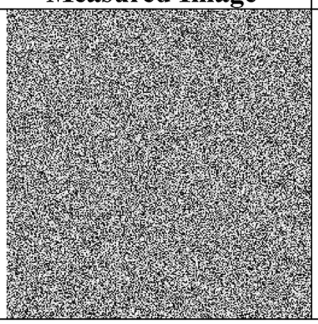
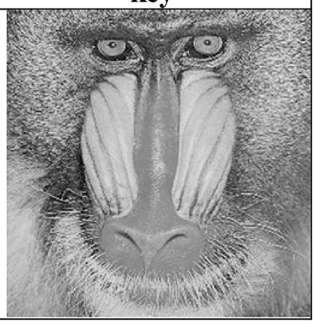
Original Image	Measured Encrypted Image	One bit change key Measured Image	Decrypt using same key
			

Table 17. Key sensitivity analysis.

Metrices	Proposed	[27]
NPCR	99.61%	99.60%
UACI	33.39%	29.44%
PSNR	9.68 dB	9.51dB
MSE	7002.95 dB	8262.66 dB
Avalanche effect	49.91	-

Table 18. Quantitative results of key sensitivity.

based or frequency-based cryptanalytic methods. The transformation from skewed, structured histograms to flat, noise-like distributions after encryption confirms the robustness of the applied quantum-enhanced encryption approach. This ensures strong confidentiality and resistance to statistical attacks.

Encryption quality analysis

Chi-square test

The Chi-Square test is a statistical method used to measure how uniformly the pixel values are distributed in an image. It is used to evaluate the randomness of the pixel intensity values in the encrypted image. A good encryption algorithm will produce an encrypted image where pixel values are randomly distributed and uniformly cover the entire intensity range, resulting in a low chi-square value. A lower Chi-square value indicates better randomness and stronger security, while a high value may reveal non-random patterns and vulnerabilities to statistical attacks.

$$\chi^2 = \sum_{i=0}^{255} \frac{(O_i - E_i)^2}{E_i} \tag{43}$$

where, χ^2 represents Chi-square statistic.

O_i is Observed frequency of the intensity value i in the encrypted image.

E_i is Expected frequency of intensity value i.

The Table 15 presents the result of pixel distribution analysis using Chi-square statistics for all tested images are significantly lower than the critical value, indicating that the encrypted images have a uniformly distributed histogram. This confirms the effectiveness of the proposed encryption method in resisting statistical attacks.

Deviation from ideality

Deviation from ideality in image encryption refers to the extent to which an encrypted image's statistical characteristics differ from a perfectly random image. The encrypted image should appear completely random with no visible structure or patterns, making it resistant to statistical and differential attacks.

The Table 16 represents deviation from expected distribution values for all images are very low and close to zero indicating that the encrypted histograms closely resemble an ideal uniform distribution. This further supports the strength and reliability of the proposed encryption scheme against statistical analysis.

Proposed QHRNG key sensitivity

Key sensitivity refers to single bit change in the quantum key affects the measurement output of encrypted image. A highly key-sensitive algorithm ensures that even minimal key alterations produce distinct encrypted output. This property is essential for preventing brute-force and differential attacks, it ensures that unauthorized decryption attempts with incorrect keys produce drastically altered results.

The Table 17 demonstrates encryption scheme's key sensitivity using the Mandril image. A one-bit change in the key produces an unintelligible output, while the correct key enables perfect decryption. This confirms strong obfuscation and high sensitivity to key variations ensures secure decryption only with the exact key.

The proposed quantum image encryption algorithm demonstrates high key sensitivity, it is a critical indicator of its cryptographic strength. This characteristic ensures that the encrypted image cannot be decrypted unless the exact original key is used. Any minor deviation in the key leads to a drastic transformation in the decrypted image, rendering it unrecognizable. This is evident from the visual and quantitative results in Table 18, Baboon test image. When decrypted with incorrect key appear entirely randomized preventing any meaningful information retrieval. The quantitative results further validate the robustness of the encryption scheme. NPCR reaches 99.61% signifies that nearly every pixel in the measured image changes when the key is slightly modified. UACI reaches 33.39% shows a substantial intensity deviation between the original and measured encrypted images. These high values confirm the algorithm's resistance to differential attacks. The key sensitivity of measured encryption scheme yields a PSNR of 9.68 dB indicates low similarity between the original and measured image. The MSE value of 7002.95 dB quantifies the large deviation introduced during encryption. The avalanche effect reflects the proportion of output bits that change when a single input bit is flipped, which is recorded at 49.91% and closely approaches the excellent diffusion characteristics. In quantum channels, the importance of key sensitivity becomes even more pronounced. Since quantum states are highly fragile, any unauthorized access attempt leads to the collapse of superposition results in garbage outputs. Only the sender and intended receiver possess synchronized encryption, decryption mechanisms, and the correct keys to recover the original image. This ensures that any attempt to tamper with the image data is immediately evident, prompting a secure retransmission.

Key space analysis

Key space refers to the total number of unique keys that can be used in the encryption process. A sufficiently large key space is essential to protect against brute-force attacks, where an attacker tries all possible keys until the correct one is found. The key space should be at least 2^{128} to ensure computational infeasibility of such attacks using classical computing power. The proposed algorithm uses various QRNG keys throughout the encryption process. The QTRNG, QPRNG and QHRNG keys are individually analysed in the encryption algorithm using Qiskit BasicSimulator and AerSimulator backend. The quantum keys are determined by number of qubits and the number of shots in measurement iterations. The measurement of quantum state space of superposition contributes two possible outcomes resulting in 2^n possible combinations for n qubits in a single shot. The proposed QTRNG produces n bit truly random seed, which initializes the QPRNG. This seed varies for each execution, the total key space grows exponentially with the number of qubits and the number of shots performed.

$$\text{Key space} = (2^{qubits})^{shots} = (2^{24})^{20,000} = 2^{4,80,000} \tag{44}$$

This immense key space ensures the strength of the hybrid structure and proposed method inherits the true randomness and entropy from the QTRNG while leveraging the deterministic structure of QPRNG to expand and manipulate the output without compromising unpredictability.

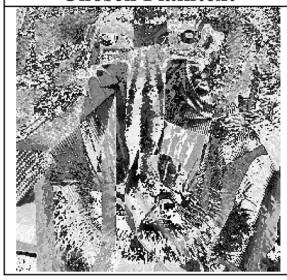
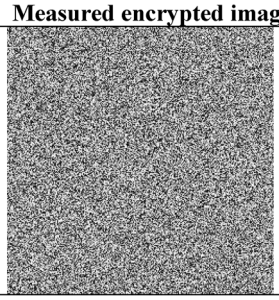
Chosen Plaintext	Measured encrypted image
	

Table 19. Chosen Plain text attack analysis.

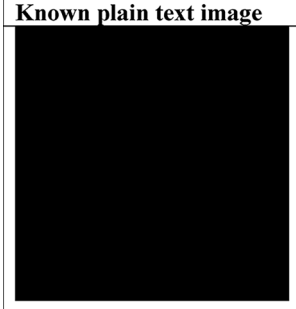
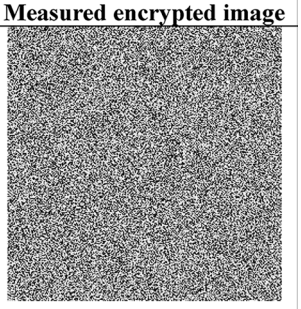
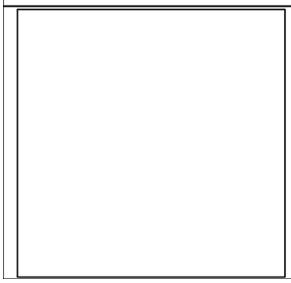
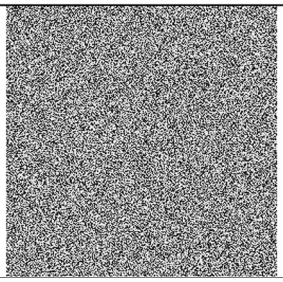
Known plain text image	Measured encrypted image
	
	

Table 20. Known plain text analysis.

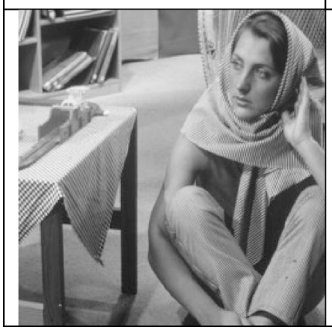
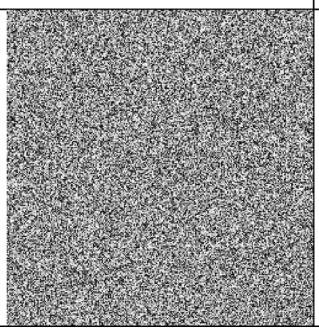
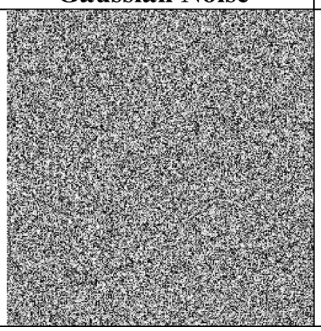
Original Image	Encrypted Image	Encrypted image with Gaussian Noise	Decrypted Image with Gaussian Noise
			

Table 21. Gaussian noise analysis.

Grover's search complexity

Grover's search algorithm can be used to search for a specific key in an unsorted database. It reduces the complexity of a brute-force search from 2^N to $2^{N/2}$. where 'N" is the number of bits in the key. For key space of $2^{4,80,000}$, the time complexity using Grover's search.

$$Key\ space\ = (2^{qubits})^{shots} = (2^{24})^{20,000}\ = 2^{4,80,000} \tag{45}$$

This is exponentially faster than a classical brute-force search and secure against any quantum attacks. The search space is still large and practically unfeasible to search through recent quantum computing capabilities.

Chosen plain text attack (CPA attack)

In a Chosen Plaintext Attack (CPA) scenario, an adversary exploits knowledge of specific plaintexts and their corresponding ciphertexts to deduce information about the encryption key or algorithm. To evaluate the resistance of the proposed quantum image encryption scheme against such an attack, a composite test image was prepared by applying a pixel-wise XOR operation between two standard benchmark images, Mandrill and Barbara. The XOR image introduces a blend of complex texture and structural features, making it a suitable candidate for adversarial analysis in CPA settings. In the proposed quantum image encryption scheme, robustness against chosen plaintext attacks is achieved by several quantum techniques, including quantum key, quantum bit-level scrambling and selective QFT components to enhance the confusion and diffusion properties of the system. It ensures that even a minor change in the plaintext produces a drastically different ciphertext. The visual results are shown in Table 19. provides strong evidence of CPA resistance. The input Mandrill and Barbara image

is subjected to a chosen plaintext modification and corresponding encrypted output. The ciphertext appears as completely unrecognizable, indicating no visible correlation with the input. This proves the algorithm's ability to obscure image patterns effectively, even under chosen conditions. The NPCR achieved 99.63% indicating nearly all pixels in the encrypted image change due to a slight plaintext modification. UACI is about 33.36%, reflecting strong intensity variation, a desirable characteristic for resisting differential and chosen plaintext attacks. The PSNR is 7.19dB and the MSE is 12425.83, both highlighting the high distortion between original and encrypted images.

Known plain text attack (KPA attack)

To evaluate the robustness of the proposed quantum image encryption algorithm against known plaintext attacks, two standard test inputs were used: a full black image (all pixel values set to 0) and a full white image (all pixel values set to 255). These inputs represent the simplest possible forms of plaintext with minimum and maximum intensity values, respectively. After applying the quantum encryption algorithm to these images, the resulting ciphertexts are shown in Table 20. appear as completely random patterns with no visible structure or discernible correlation to the original inputs.

Resistivity to attacks

Gaussian noise

Gaussian noise is a statistical noise with a probability density function equal to the normal gaussian distribution. It affects pixel values by adding random variations. In Table 21, the original grayscale image is first encrypted using the proposed quantum encryption scheme. The encrypted image appears completely random, indicating strong pixel diffusion. Gaussian noise is then introduced into the encrypted image to simulate potential distortions during transmission. Upon decrypting the noisy encrypted image, although some distortion is visible, the essential structure and features of the original image are still retained. This demonstrates that the proposed algorithm is highly robust against Gaussian noise, ensuring that minor perturbations in the encrypted image do not prevent successful decryption. The test image 'Barbara' was obtained from the USC-SIPI Image Database[25] (Miscellaneous volume).

Salt and pepper noise

Salt and Pepper noise is a form of impulse noise that manifests as random black and white pixels scattered throughout the image. It typically arises due to sharp and sudden disturbances in the image signal. The original image is encrypted using the proposed quantum encryption method. Salt-and-pepper noise is then applied to the encrypted image. When the noisy encrypted image is decrypted, the image content remains mostly intact and identifiable despite the presence of visible noise artefacts. The ability of the algorithm to recover meaningful information even in the presence of severe impulse noise confirms its high resilience. The test image 'Barbara' was obtained from the USC-SIPI Image Database[25] (Miscellaneous volume).

The Table 22 presents the robustness of the encryption scheme under Salt and Pepper noise conditions using a standard test image. The original image is first encrypted into a noise-like format. After the addition of Salt and Pepper noise to the encrypted image, decryption is still able to recover the original content with minimal visual distortion. This demonstrates the resilience of the encryption algorithm against noise attacks confirms its effectiveness in maintaining data integrity under adverse conditions.

Computational complexity

The computational complexity of the proposed quantum image encryption algorithm is determined by quantum key generation, quantum bit-level scrambling and selective QFT operations utilizing Hadamard and Controlled-Rotation-Z gates. The algorithm uses NEQR for encoding grayscale images where an image of size $2^n \times 2^n$ requires $2n + 8$ qubits for both pixel position and intensity information. In the quantum key generation phase,

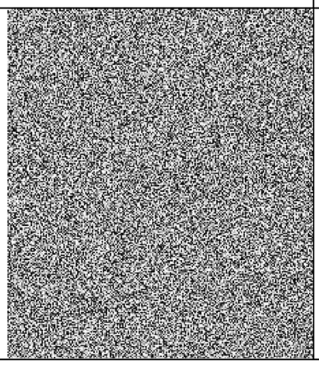
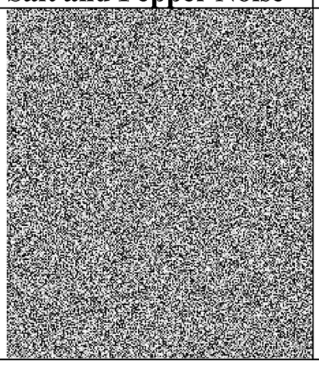
Original Image	Encrypted Image	Encrypted image with Salt and Pepper Noise	Decrypted Image with Salt and Pepper Noise
			

Table 22. Salt and pepper analysis.

Method	Encryption Technique	Randomness Source	Robustness analysis	Quantum Hardware Feasibility	Advantages of the Proposed Method
[2]	DNA encoding and CNOT-based scrambling	Pseudo-random classical DNA encoding	Moderate NPCR/UACI, entropy not optimal	Simulated only	Lacks quantum randomness and hardware support; the proposed method uses QHRNG and real hardware testing
[26]	Fractional wavelet, chaotic system and quantum transform	Classical chaos and mathematical transforms	Theoretical robustness; limited attack testing	Purely theoretical model	No real quantum implementation or randomness; proposed method integrates real QTRNG and QPRNG
[27]	2D quantum walk and quantum coding	Quantum walk patterns	High diffusion, theoretical analysis	No hardware implementation	High circuit complexity, no entropy or plaintext attack testing; proposed method shows hardware feasibility and full attack analysis
[28]	Bit-plane permutation and sine logistic map	Classical chaotic maps	Moderate NPCR/UACI, entropy not near ideal	Fully classical model	No quantum integration; proposed method uses selective QFT and quantum gates compatible with NISQ
Proposed Method	QHRNG key (QTRNG and QPRNG), Quantum bit-level scrambling, selective QFT and parallel processing.	True and pseudo quantum randomness (QHRNG)	NPCR > 99.5%, UACI ~ 33.5%, Entropy ~ 7.99, Avalanche effect ~ 50%, chi-square, PSNR and correlation ~ low, MSE ~ high	Implemented on IBM Qiskit Noisy and Noiseless simulations and real quantum hardware implementation of QHRNG	Combines QTRNG and QPRNG achieves high entropy, strong resistance to attack models supports real-time quantum hardware with polynomial complexity

Table 23. Advantages of the proposed method over existing methods.

the QTRNG utilizes H and CRZ gates contributes a complexity of $\mathrm{O}\left(n^2\right)$ due to entanglement among the qubit pairs. The QPRNG utilizes Hadamard and phase gate along with Double-Controlled-Not gates is also contributes the complexity of $\mathrm{O}\left(n^2\right)$. The QHRNG integrates outputs from both QTRNG and QPRNG, initialising QTRN bits to enhance randomness, thereby contributing to the overall algorithmic complexity of $\mathrm{O}\left(n^2\right)$. The bit-level scrambling process involves quantum operations such as Controlled-SWAP and CNOT gate transformations. These operations are applied to each pixel's bits and scale linearly with the image size leading to a complexity of $\mathrm{O}(M \times N)$, where $M \times N$ represents the total number of pixels. The selective QFT component applies a Hadamard gate to each of the 24 qubits, resulting in a linear complexity of $\mathrm{O}(n)$. The controlled rotation gates are applied between each pair of quadratic growth leads to a complexity of $\mathrm{O}\left(n^2\right)$. Therefore, the total quantum computational cost for the selective QFT with 24 qubits is 24 (Hadamard gates) + 276 (CRZ gates) = 300. The overall quantum computational complexity of the proposed encryption algorithm is $\mathrm{O}\left(n^2\right)$ ensures scalable and efficient for quantum implementations. In classical encryption methods exhibit a complexity of $\mathrm{O}\left(2^{2n}\right)$ due to the exponential growth associated with bitwise operations on each pixel. This highlights the advantage of the quantum approach in handling large-scale image encryption tasks with improved efficiency.

Comparative analysis

The Table 23 presents a comparative evaluation of the proposed quantum-enhanced encryption scheme against existing methods. While prior approaches predominantly rely on classical randomness, theoretical robustness, and lack practical hardware implementation, the proposed method stands out by integrating both quantum true and pseudo-randomness achieves significantly higher robustness metrics. The table provides a detailed analysis, highlighting the advantages of the proposed method in terms of entropy, security performance and hardware feasibility.

Discussion

The experimental evaluation demonstrates the practicality of the proposed quantum image encryption scheme integrating QTRNG, QPRNG and QHRNG-based key generation frameworks. The QHRNG method combining the inherent unpredictability of QTRNG with the deterministic structure of QPRNG consistently outperformed standalone random generators in terms of randomness quality, entropy and resistance to attacks across both noiseless (*BasicSimulator*) and noisy (*AerSimulator*) environments. The NPCR values consistently exceeded 99.56%, indicating strong sensitivity to single-pixel modifications, and UACI values near 33.5% confirm significant pixel intensity changes after encryption. These metrics reflect the proposed encryption method and confirm high robustness against differential attacks. The Avalanche effect value of around 50% validates that even minimal changes in the input image propagate unpredictability across the measured encrypted image, confirming a strong diffusion. The entropy values near the ideal value of 8 confirm that the encrypted images exhibit perfect randomness. This is particularly evident with QHRNG key encryption; entropy peaked at 7.9974, especially in standard Mandril textured images. The correlation coefficients near zero and flat histograms in encrypted outputs signify the effective removal of spatial and intensity-based dependencies, strengthening the resistance to statistical analysis. Chi-square test values remained far below the critical threshold, and deviation from ideality values was close to zero, confirming uniform pixel distribution for post-encryption. The key sensitivity analysis further established the reliability of the encryption scheme. A one-bit key alteration led to drastically altered decrypted outputs, reinforcing the strength of the proposed encryption algorithm against brute-force attacks. The increasing key space with the number of qubits and shots makes exhaustive key search infeasible, even Grover's search algorithm considered. This proposed encryption method is resilient under chosen plaintext

Metrics	QTRNG	QPRNG	QHRNG
Average P-Value (NIST SP800-22)	0.7012	0.6448	**0.7595**
Average Min-Entropy (NIST SP800-90B)	0.8932	0.8637	**0.9171**
Variability across Restart Experiment	High	Medium	**Highest**
Autocorrelation	$\leq$ 0.19	$\leq$ 0.17	$\leq$ **0.12**
NPCR (%)	99.56–99.66	99.58–99.62	**99.66**
UACI (%)	33.43	33.39–33.50	**33.54**
Avalanche Effect (%)	49.91	49.96	**50.15**
PSNR (dB)	8.92–9.60	9.00-9.61	**9.62–9.68**
MSE (dB)	8200–9500	7900–8262	**7003**
Information Entropy	7.9967	7.9950	**7.9974**
Correlation Coefficient	0.0044	0.0039	**0.0026**
Histogram Uniformity	Moderate	High	**Highest**
Chi-Square Value	0.4201	1.5454	**0.4921**
Linear Complexity	Moderate	High	**Very High**
Approximate Entropy	0.221	0.195	**0.173**
Serial Test P-Value	0.641	0.605	**0.707**

Table 24. Quantitative analysis of proposed different random number generators using quantum simulators.

and known plaintext attacks, where the intruder attempts to exploit patterns between known input and output pairs. The stable PSNR and MSE values across different simulators show noise resilience and a desirable trait for strong encryption. The quantum computational complexity of $O\left(n^2\right)$ ensures efficient and scalable even image size and qubit count increases. The simulation on real IBM quantum hardware (*ibm_torino*) confirms the practical deployability of the QTRNG, QPRNG and QHRNG method. This hardware implementation validates the feasibility of all proposed quantum circuits on real quantum processors demonstrates its practicality for secure cryptographic applications.

Quantitative comparison of QTRNG, QPRNG and QHRNG

The performance claims of the proposed QHRNG with a quantitative comparison with QTRNG and QPRNG model is presented in Table 24. The QTRNG relies solely on quantum measurement induced randomness and offering high unpredictable but limited scalability and noise resilience. The QPRNG uses deterministic quantum gate sequences to generate complex patterns from fixed initial states provides structural randomness but lacking true quantum entropy. The QHRNG integrates both approaches by using QTRNG generated seed to initialize a gate driven pseudo-random circuit combines the entropy of quantum indeterminacy with the complexity of entangled quantum evolution. This hybrid architecture leads to enhance randomness, improved entropy and greater resistance to differential and statistical attacks. Table 24 summarizes the performance of each method across a range of statistical, cryptographic and image encryption metrics demonstrating the superior performance of QHRNG.

Potential applications and future direction

The proposed quantum encryption framework combines QTRNG, QPRNG, and QHRNG to provide robust, scalable and highly secure cryptographic solutions for next-generation applications. QTRNG provides quantum superposition and quantum measurement collapses to generate highly unpredictable entropy random sequences, making them highly suitable for Quantum Key Distribution (QKD), blockchain security and authentication protocols. QPRNG are used in repeatable randomness for synchronization, simulation and secure algorithm development. QHRNG effectively combines the advantage of QTRNG bits as seeds in QPRNG circuits, maintaining unpredictability in random sequences. The hybrid model uses quantum cloud security, multi-user encrypted communication systems, and secure routing in quantum-enhanced IoT devices. The selective QFT in the quantum encryption method introduces a spectral transformation that obscures the image data in the quantum domain. This method extends to secure satellite imagery transmission, protection of medical image records, classified surveillance data encryption and secure biometric template storage. QHRNG adds an adaptive layer of proposed encryption, ensuring resilience under dynamic quantum environments and classical attack vendors. This method can be developed on NISQ (Noisy Intermediate-Scale Quantum) systems using real quantum hardware from the IBM quantum platform. Future work will integrate quantum error correction codes and mitigation techniques to address gate errors and decoherence. These advancements will facilitate the shift from simulated quantum encryption systems to real-time quantum cryptographic hardware applications, enabling scalable and secure quantum communication networks.

The Table 25 highlights the applications of QTRNG, QPRNG and QHRNG. QTRNG ensures high-entropy randomness for secure communication and cryptography. QPRNG supports controlled randomness in simulations and testing. QHRNG combines both for robust, adaptive security across domains like IoT, cloud authentication, medical imaging and cybersecurity.

Proposed Method	Application Domain	Practical Usage	Benefits
QTRNG	Quantum Cryptography	Quantum Key Distribution, One-time pads	High entropy, non-deterministic randomness, ideal for key generation
	Secure Communication	Satellite communication encryption, Military-grade communication	It prevents eavesdropping due to quantum measurement collapse.
	Blockchain & Smart Contracts	Generation of Cryptographic tokens	It enhances randomness in blockchain processes for tamper-proof transaction
QPRNG	Simulation & Modelling	Randomized quantum algorithms, Quantum Monte Carlo simulations	Deterministic control with seeded randomness suitable for repeated simulations
	Game Development & Random Testing	Quantum-enhanced random behaviour in game mechanism and fuzzy testing	Controlled randomness for test reproducibility
QHRNG	Secure Multi-party Communication	Quantum cloud authentication, Key sharing between users	It combines the unpredictability of QTRNG and control of QPRNG for optimized security
	Quantum IoT Networks	Secure routing and data encryption in IoT edge devices	Lightweight and adaptive quantum security solution
	Quantum-Enhanced Cybersecurity	Random challenge-response authentication, Protocol obfuscation	Adaptive encryption with increased resilience
Quantum Image encryption	Medical Imaging	Secure sensitive diagnostic images, Telemedicine image transfer	It protects patient data against theft and tampering
	Biometric Authentication Systems	Encryption of facial, fingerprint and iris images	Secure storage and transmission of personal biometric templates
	Secure Multimedia Sharing	Secure photo and video exchange over quantum internet	Resists statistical and correlation-based attacks
	Military & Surveillance Imaging	Confidential drone footage, satellite surveillance	Tamper-resistant encryption for real-time secure visual communication
	Cloud-based Quantum Storage	Quantum image databases with secure access	It ensures data privacy even under cloud compromise.

Table 25. Potential applications and implications.

Conclusion

This research proposes a secure and efficient quantum image encryption scheme that integrates QTRNG and QPRNG into a hybrid quantum key framework and validates the randomness through the Restart experiment, Autocorrelation, NIST statistical and entropy test. The quantum images are encrypted using quantum bit-level scrambling and selective QFT to ensure strong confusion and diffusion across the quantum image data. Extensive noisy and noiseless simulation and hardware-based testing demonstrated the proposed method's high resistance to statistical, differential and noise-based attacks. Metrics such as NPCR, UACI, entropy, chi-square deviation and correlation coefficients confirmed the encryption's effectiveness in achieving pixel randomness and eliminating structural redundancy. Furthermore, key sensitivity analysis and plaintext attack evaluations confirmed that even minor variations in the key or input led to significantly different encrypted outputs. The hybrid approach outperformed standalone QTRNG and QPRNG methods by balancing unpredictability and structural control. It proved adaptable for practical applications, including secure quantum cloud storage, medical imaging, satellite communication, and biometric protection. The encryption algorithm also maintained a polynomial quantum computational complexity making it scalable for high-resolution image data. Future work will incorporate quantum error correction and mitigation techniques to overcome hardware noise and gate imperfections for practical quantum cryptographic systems.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Code and Data availability

The source code and execution log files supporting this study are available at: https://github.com/Gururaja-TS/Quantum-secure-image-encryption-using-Hybrid-QTRNG-and-QPRNG.git.

Received: 7 January 2025; Accepted: 2 January 2026
Published online: 06 February 2026

References

1. Ali, Z. A. et al. A comprehensive review of quantum image encryption methods: design characteristics, cryptographic properties, and AI integration. *Quantum Inf. Process.* **23**, 335. https://doi.org/10.1007/s11128-024-04563-y (2024).
2. Zhou, S. A quantum image encryption method based on DNACNot. *IEEE Access.* **8**, 178336–178344. https://doi.org/10.1109/ACCESS.2020.3027964 (2020).
3. Zhao, J. et al. Color image encryption scheme based on alternate quantum walk and controlled rubik's cube. *Sci. Rep.* **12**, 14253. https://doi.org/10.1038/s41598-022-18079-x (2022).
4. Liu, X., Xiao, D. & Xiang, Y. Quantum image encryption using intra and inter bit permutation based on logistic map. *IEEE Access.* **7**, 6937–6946. https://doi.org/10.1109/ACCESS.2018.2889896 (2019).
5. Li, X., Li, T., Wu, J., Xie, Z. & Shi, J. Joint image compression and encryption based on sparse bayesian learning and bit-level 3D Arnold Cat maps. *PLoS ONE.* **14**, e0224382. https://doi.org/10.1371/journal.pone.0224382 (2019).
6. Shafique, A. et al. A hybrid encryption framework leveraging quantum and classical cryptography for secure transmission of medical images in IoT-based telemedicine networks. *Sci. Rep.* **14**, 1–24. https://doi.org/10.1038/s41598-024-82256-3 (2024).
7. Ye, G., Jiao, K., Huang, X., Goi, B. & Yap, W. An image encryption scheme based on public key cryptosystem and quantum logistic map. *Sci. Rep.* **10**, 1–19. https://doi.org/10.1038/s41598-020-78127-2 (2020).
8. Yang, Y., Xu, P., Yang, R., Zhou, Y. & Shi, W. Quantum hash function and its application to privacy amplification in quantum key distribution, pseudo-random number generation and image encryption. *Sci. Rep.* **6**, 1–14. https://doi.org/10.1038/srep19788 (2016).
9. Argillander, J. et al. Quantum random number generation based on a perovskite light-emitting diode. *Commun. Phys.* **6**, 1–7. https://doi.org/10.1038/s42005-023-01280-3 (2023).
10. Yang, Y. & Zhao, Q. Novel pseudo-random number generator based on quantum random walks. *Sci. Rep.* **6**, 1–11. https://doi.org/10.1038/srep20362 (2016).
11. Satre, S. M. & Joshi, B. Quantum image cryptography based on continuous chaotic maps. *Microsyst. Technol.* https://doi.org/10.1007/s00542-024-05764-2 (2024).
12. Wu, W. & Wang, Q. Quantum image encryption based on Baker map and 2D logistic map. *Int. J. Theor. Phys.* **61**, 64. https://doi.org/10.1007/s10773-022-04979-1 (2022).
13. Ye, G., Jiao, K. & Huang, X. Quantum logistic image encryption algorithm based on SHA-3 and RSA. *Nonlinear Dyn.* **104**, 2807–2827. https://doi.org/10.1007/s11071-021-06422-2 (2021).
14. Hu, W. W. et al. Quantum image encryption algorithm based on generalized Arnold transform and logistic map. *CCF Trans. HPC.* **2**, 228–253. https://doi.org/10.1007/s42514-020-00043-8 (2020).
15. Liu, X., Xiao, D. & Liu, C. Three-level quantum image encryption based on Arnold transform and logistic map. *Quantum Inf. Process.* **20**, 23. https://doi.org/10.1007/s11128-020-02952-7 (2021).
16. Jacak, M. M. et al. Quantum generators of random numbers. *Sci. Rep.* **11**, 16108. https://doi.org/10.1038/s41598-021-95388-7 (2021).
17. Chamon, C. et al. Fast pseudorandom quantum state generators via inflationary quantum gates. *Npj Quantum Inf.* **10**, 37. https://doi.org/10.1038/s41534-024-00831-y (2024).
18. IBM. *Qiskit*. Available at: https://www.ibm.com/quantum/qiskit
19. IBM. *IBM Quantum Documentation*. Available at: https://docs.quantum.ibm.com/
20. Zhang, Y. et al. NEQR: a novel enhanced quantum representation of digital images. *Quantum Inf. Process.* **12**, 2833–2860. https://doi.org/10.1007/s11128-013-0567-z (2013).
21. Gururaja, T. S. et al. Implementation of RQFT-QTRNG Using Quantum Gates in the IBM Quantum Lab. In *Harnessing Quantum Cryptography for Next-Generation Security Solutions*, edited by N. K. Chaubey and N. Chaubey, 413–438. IGI Global, (2025). https://doi.org/10.4018/979-8-3693-9220-1.ch014
22. Kumar, V., Rayappan, J. B. B., Amirtharajan, R. & Praveenkumar, P. Quantum true random number generation on ibm's cloud platform. *J. King Saud Univ. - Comput. Inform. Sci.* **34**, 6453–6465. https://doi.org/10.1016/j.jksuci.2022.01.015 (2022).
23. Bassham, L. E. et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications. *NIST Special Publication* 800-22r1a (2010). https://doi.org/10.6028/NIST.SP.800-22r1a
24. Turan, M. S. et al. Recommendation for the entropy sources used for random bit generation. *NIST Spec. Publ.* **800-90B** https://doi.org/10.6028/nist.sp.800-90b (2018).
25. SIPI Database. *Image 30* (miscellaneous). Available at: https://sipi.usc.edu/database/
26. Yan, X., Teng, L. & Su, Y. A novel chaotic image encryption is based on fractional wavelet decomposition and quantum transform model. *Phys. Scr.* **99**, 055217. https://doi.org/10.1088/1402-4896/ad368b (2024).
27. Hao, W., Zhang, T., Chen, X. & Zhou, X. A hybrid NEQR image encryption cryptosystem using two-dimensional quantum walks and quantum coding. *Sig. Process.* **205**, 108890. https://doi.org/10.1016/j.sigpro.2022.108890 (2023).
28. Liu, X., Xiao, D. & Liu, C. Quantum image encryption algorithm based on bit-plane permutation and sine logistic map. *Quantum Inf. Process.* **19**, 239. https://doi.org/10.1007/s11128-020-02739-w (2020).
29. Gururaja, T. S. & Pravinkumar, P. Enhanced quantum image encryption using DNA–QTRNG and Sattolo-RQFT shuffling. *J. Supercomput.* **81**, 667. https://doi.org/10.1007/s11227-025-07085-1 (2025).
30. Alblehai, F. et al. Cascading quantum walks with Chebyshev map for designing a robust medical image encryption algorithm. *Sci. Rep.* **15**, 6685. https://doi.org/10.1038/s41598-025-90725-6 (2025).
31. Gururaja, T. S. & Pravinkumar, P. Quantum Phase True Random Number Generation for Secure Sustainable 6G Communication, 2025 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), Chennai, India, 2025, pp. 1–7. https://doi.org/10.1109/WiSPNET64060.2025.11005305
32. Salehi, R., Razaghi, M. & Fotouhi, B. Hybrid Hadamard and controlled-Hadamard based quantum random number generators in IBM QX. *Phys. Scr.* **97** (6), 065101. https://doi.org/10.1088/1402-4896/ac6762 (2022).
33. Li, Y. et al. Quantum random number generator using a cloud superconducting quantum computer based on source-independent protocol. *Sci. Rep.* **11**, 23873. https://doi.org/10.1038/s41598-021-03286-9 (2021).

34. Kumar, V. Experimenting with quantum true random number generators on NISQ computers using high-level quantum programming. *Int. J. Theor. Phys.* **64**, 238. https://doi.org/10.1007/s10773-025-06104-4 (2025).
35. Guo, X. et al. Enhancing extractable quantum entropy in vacuum-based quantum random number generator. *Entropy* **20** (11), 819. https://doi.org/10.3390/e20110819 (2018).

Acknowledgements
The Authors would like to Thank and Acknowledge Department of Telecommunications, India for their financial support through Telecom Technology Development Fund (TTDF) (SDUTTDF/6G/239) and SASTRA Deemed to be University, India, for providing the necessary resources and support.

Author contributions
Gururaja T S and Padmapriya Pravinkumar conceptualized and designed the study. Gururaja T S conducted the experiments and performed data analysis. Gururaja T S drafted the Manuscript. Padmapriya Pravinkumar provided critical feedback, helped write and edit the manuscript and gave guidance throughout the project. All authors reviewed and approved the final manuscript.

Funding
The Authors would like to Thank and Acknowledge Department of Telecommunications, India for their financial support through Telecom Technology Development Fund (TTDF) (SDUTTDF/6G/239).

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to P.P.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.