Int. J. Appl. Math. Comput. Sci., 2026, Vol. 36, No. 2, 267–280
DOI: 10.61822/amcs-2026-0018

AN IMPROVED CHACHA ALGORITHM BASED ON QUANTUM RANDOM NUMBERS

CHAO LIU[a], SHUAI ZHAO[a,b,c,*], CHENHAO JIA[a], GENGRAN HU[a,c], TINGTING CUI[a,c]

[a] School of Cyberspace
Hangzhou Dianzi University
Hangzhou, Zhejiang, 310018, China
e-mail: zhaoshuai@hdu.edu.cn

[b] Pinghu Digital Technology Innovation Institute Co., Ltd.
Hangzhou Dianzi University
Jiaxing, Zhejiang, 314299, China

[c] Zhejiang Provincial Key Laboratory of Sensitive Data Security and Confidentiality Governance
Hangzhou, Zhejiang, 310018, China

Due to the merits of high efficiency and strong security against timing and side-channel attacks, ChaCha has been widely applied in real-time communication and data streaming scenarios. However, with the rapid development of AI-assisted cryptanalysis and quantum computing technologies, there are serious challenges to secure implementation of the ChaCha cipher. To further strengthen its security, we propose an improved variant based on quantum random numbers, i.e., quantum random number enhanced ChaCha (QRE-ChaCha). Specifically, the design XORs the initial constants with quantum random numbers and periodically injects the latter into selected state words during odd rounds to enhance diffusion. Compared with the original ChaCha, the present variant shows stronger resistance to differential attacks and generates a keystream with statistical randomness, thereby offering increased robustness against both classical and quantum attacks. To evaluate the security and performance of the present ChaCha, our analysis proceeds in three main parts. Firstly, we analyze its theoretical security in terms of quantum randomness and attack testing, and conduct differential cryptanalysis with an automated search method based on the Boolean satisfiability problem (SAT). Secondly, we subject the keystream generated by the cipher to randomness tests using the NIST Statistical Test Suite and the GM/T 0005-2021 randomness testing standard. Finally, we assess its encryption and decryption performance by measuring the encryption speed on files of various sizes. According to the results, the new ChaCha is significantly improved to resist differential attacks while maintaining the high efficiency of the original ChaCha cipher, and its keystream successfully passes statistical randomness tests using the NIST and GM/T 0005-2021 standards, meeting cryptographic application requirements.

Keywords: stream cipher, ChaCha, quantum random numbers, QRE-ChaCha.

1. Introduction

With the rapid development of information technology, the demand for data security has significantly increased, particularly in modern communication and storage systems. Among the various cryptographic algorithms, ChaCha has been widely adopted in critical protocols, such as Transport Layer Security (TLS) (Langley *et al.*, 2016), due to its exceptional speed and robust security design, securing vast amounts of data in transit and at rest. However, as quantum technologies and AI-assisted cryptanalysis techniques have achieved significant progress, the security of conventional cryptographic schemes, including ChaCha, faces more severe challenges. In response to these, prominence has been obtained in two major research directions: post-quantum cryptography (Bennett and Brassard, 2014; Pirandola *et al.*, 2020; Portmann and Renner, 2022; Li *et al.*, 2023; Renner, 2008; Lo *et al.*, 2005; 2012; Acín *et al.*, 2007), which focuses on developing new algorithms based on mathematical problems that are conjectured to be intractable for quantum computers, and quantum cryptography (Bennett and

*Corresponding author

Brassard, 2014; Pirandola *et al*., 2020; Portmann and Renner, 2022; Li *et al*., 2023; Renner, 2008; Lo *et al*., 2005; Acín *et al*., 2007; Lo *et al*., 2012), which is rooted in quantum physical principles. Nevertheless, both approaches have significant limitations in enhancing existing symmetric ciphers: post-quantum cryptography research is largely concerned with asymmetric cryptosystems, while in quantum cryptography the large-scale deployment of quantum key distribution (QKD) is currently impeded by technological and infrastructural constraints, despite ongoing research efforts to improve network efficiency through advanced routing optimizations (Szczepanik and Niemiec, 2025).

Due to the inherent uncertainty in quantum physics, quantum random numbers possess intrinsic randomness, which is an important branch of quantum cryptography research, and also one of the quantum technologies with mature applications at present. To enhance the security of classical cryptographic algorithms, it has been highly motivated to combine quantum random numbers with classical cryptographic protocols (Li *et al*., 2023), which on the one hand can extend the application scenarios of quantum random number generators, and on the other improve the security of classical cryptographic protocols. In this work, we draw on a similar idea to enhance the algorithm by applying quantum random numbers to the ChaCha stream cipher.

The security of that cipher relies heavily on the quality of its inputs, specifically the seed key and the nonce. In real-world implementations, these inputs are often generated by deterministic algorithms or classical physical noise sources. As noted in recent studies on quantum randomness (Li *et al*., 2023), relying on deterministic functions, such as hash ones or pseudo-random number generators (PRNGs), to simulate randomness creates a gap between theoretical security models and physical reality. These deterministic outputs lack intrinsic randomness (Dhara *et al*., 2014) and can potentially be predicted if the internal state is compromised or reset, leading to catastrophic failures like nonce reuse.

Unlike classical noise sources, which are governed by deterministic classical physics, quantum random number generators (QRNGs) rely on the intrinsic probabilistic nature of quantum mechanics, providing provable randomness and intrinsic unpredictability. Recent breakthroughs have further demonstrated the capability to generate traceable and certified randomness even in complex environments (Kavuri *et al*., 2025). QRNGs have no internal state to reset and no cycle to repeat. Each generated bit is an independent event. By using a high-entropy QRNG to generate the nonce, the selection is drawn from a space with true uniformity, and the probability of a collision due to physical randomness is cryptographically negligible. Therefore, using QRNGs to enhance existing symmetric ciphers offers a practical and immediate security upgrade.

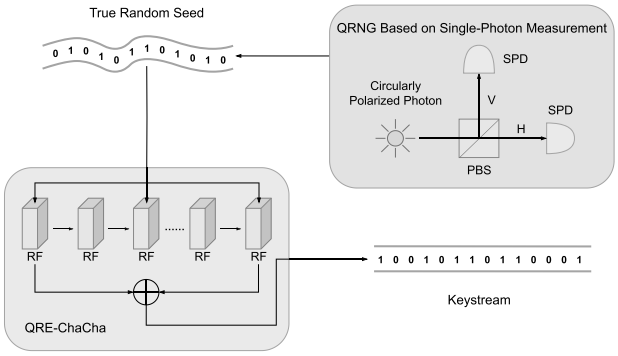

Fig. 1. QRE-ChaCha: an integration scheme of quantum random numbers with a classical cipher. SPD: the single-photon detector, PBS: the polarization beam splitter, RF: the round function.

As illustrated in Fig. 1, we propose an improved ChaCha cipher based on quantum random numbers that significantly improves its security while maintaining the original cipher's high performance. Unlike previous optimizations that focus on modifying the arithmetic round function, our approach injects intrinsic quantum entropy directly into the cipher's state. The proposed scheme increases the randomness and cryptanalytic resistance of the round function, strengthens the security of the generated keystream, and enhances the cipher's resilience against differential cryptanalysis and the resistance of its seeds to quantum attacks. The main contributions of this work are summarized as follows:

1. To overcome the deterministic nature of classical inputs and enhance resistance against differential cryptanalysis, we propose a quantum random number enhanced stream cipher, QRE-ChaCha, which improves the security of the traditional ChaCha cipher by intermittently injecting quantum random numbers into the initial seed and its round function. Specifically, for the initial state matrix, quantum random numbers are XORed with the initial constants; for the intermediate state matrix produced by odd-numbered rounds, quantum random numbers are selectively XORed with specific positions to enhance diffusion. The quantum random numbers are generated through physical processes based on the principles of quantum mechanics, providing true randomness and unpredictability, which significantly strengthens the cipher's resistance to cryptanalytic attacks.

2. To evaluate the security of the QRE-ChaCha cipher against differential cryptanalysis, we employed an automated search method based on the Boolean satisfiability problem (SAT).

According to the results in Fig. 3, QRE-ChaCha achieves better differential probabilities for both 2-round and 3-round configurations compared to the original ChaCha, indicating improved resistance to differential attacks. In addition, we conducted statistical randomness tests on the generated keystream using the NIST Statistical Test Suite and the Chinese national standard GM/T 0005-2021 for randomness evaluation. The results show that the keystreams generated by QRE-ChaCha passed both sets of tests, demonstrating compliance with cryptographic standards.

3. To evaluate the performance of the QRE-ChaCha cipher, we measured its encryption and decryption speeds and compared them with those of the original ChaCha cipher. The purpose of this test was to examine whether the enhanced security features of QRE-ChaCha lead to any significant performance degradation. According to the results, when the time overhead of quantum random number generation is excluded, since the random numbers can be pre-stored in the memory, as shown in Fig. 2, the encryption and decryption speeds of QRE-ChaCha are nearly identical to those of ChaCha.

4. Beyond its role as a cipher enhancement, the proposed QRE-ChaCha scheme can also be regarded as a quantum randomness expansion scheme, which utilizes a limited set of quantum random seeds to generate a significantly larger volume of cryptographically secure randomness through iterative encryption rounds.

The remainder of this paper is organized as follows. In Section 2, we provide a brief overview of related work and recent advances in the study of ChaCha and quantum random numbers. To support the proposed design, we introduce the necessary preliminaries in Section 3. In Section 4, we detail the QRE-ChaCha enhancement scheme. In Section 5, to analyze the security of QRE-ChaCha, we describe its theoretical properties and present the results of our differential cryptanalysis. In Section 6, we present the methodology and results of randomness testing. A performance evaluation is then provided in Section 7. Finally, we conclude this work in Section 8.

2. Related work

2.1. Quantum random number generation. Randomness is a foundational resource in cryptography. Unlike classical PRNGs, which are deterministic and computationally bounded, QRNGs exploit the fundamental indeterminacy of quantum physics to provide intrinsic unpredictability (Mannalatha *et al*., 2023; Herrero-Collantes and Garcia-Escartin, 2017). To mitigate trust issues in hardware, research has evolved toward device-independent (DI) and semi-device-independent (SDI) models (Ma *et al*., 2016), which guarantee randomness even with untrusted components. Recent breakthroughs have further extended these concepts to quantum networks and practical certifications. For instance, Zhao *et al*. (2026) proposed optimal protocols for certifying DI randomness in quantum networks, while Kavuri *et al*. (2025) demonstrated traceable random numbers derived from non-local quantum advantages, reinforcing the rigorous validation of physical entropy sources.

In terms of integrating these quantum sources into practical cryptographic systems, Iavich *et al.* (2020; 2021) developed hybrid QRNGs combining photon arrival times with classical post-processing to balance speed and entropy for cryptographic use. Stipcevic (2012) further emphasized the necessity for QRNGs in secure systems by comparing them with oscillator-based RNGs. Beyond hardware, alternative approaches like pseudo-quantum generators based on permutation pads (Kuang *et al*., 2021) and cloud-based quantum randomness services (Huang *et al*., 2021) have been proposed to widen accessibility.

It is worth noting that, in 2023, Li *et al.* proposed an enhanced zero-knowledge proof scheme based on device-independent quantum randomness, further demonstrating the feasibility of integrating quantum random numbers with classical cryptographic protocols. In their work, a quantum solution was introduced in the form of a quantum randomness service, which generates random numbers via loophole-free Bell tests and transmits them using post-quantum cryptographic authentication, thereby improving the overall security of the protocol. Inspired by similar principles, the present study introduces quantum random numbers into the round function of the ChaCha cipher to enhance its security, leading to the design of QRE-ChaCha.

2.2. ChaCha cipher. Since its design by Bernstein (2008), ChaCha has effectively replaced Salsa20 in many high-performance scenarios. Consequently, it has been subjected to extensive security analysis. Early evaluations focused on implementation security, including side-channel resistance (Najm *et al*., 2018), fault attacks (Kumar *et al*., 2017), and authenticated encryption security in multi-user settings (Procter, 2014; Degabriele *et al*., 2021), confirming its robustness relative to the Advanced Encryption Standard (AES) (Centellas Claros *et al*., 2022) and its rotational properties (Barbero *et al*., 2023).

The primary cryptanalytic threat to ChaCha remains differential cryptanalysis. Since the introduction of probabilistic neutral bits (PNBs) by Aumasson

et al. (2008), attack techniques have rapidly evolved. Subsequent researchers improved key-recovery methods using high-probability differential paths (Shi *et al*., 2013; Choudhuri and Maitra, 2016) and refined PNB attacks (Deepthi and Singh, 2018). Later advancements include optimized mask selection using MILP (Bellini *et al*., 2023) and higher-order differential-linear attacks (Ghafoori and Miyaji, 2024), which have progressively reduced the security margin of reduced-round variants.

To counter these threats, several structural optimizations have been set forth. For example, Mahdi *et al*. (2021) proposed the so-called Super ChaCha by modifying rotation processes to resist cryptanalysis, while others have tailored ChaCha for IoT constraints (Jain *et al*., 2022), lightweight video encryption (Maolood *et al*., 2022), or structural enhancements like EChaCha20 (Kebande, 2023). However, most existing improvements, such as ARX operations, focus on modifying the arithmetic round function. Few approaches address the fundamental deterministic nature of the initial state generation, leaving the cipher vulnerable if the seed entropy is compromised, which is a gap this work aims to fill.

3. Preliminaries

To facilitate understanding, we first summarize the relevant notational conventions in Table 1, followed by a concise overview of the ChaCha state structure required for our proposed scheme.

ChaCha is an ARX-based stream cipher operating on a 512-bit internal state. The state is organized as a 4×4 matrix of 32-bit words. The initial state $X^{(0)}$ is composed of 128 bits of constants ($c_0 \dots c_3$), a 256-bit key ($k_0 \dots k_7$), a 32-bit counter (t_0), and a 96-bit nonce ($\nu_0 \dots \nu_2$), arranged as follows:

$$\begin{aligned} X^{(0)} &= \begin{pmatrix} x_0^{(0)} & x_1^{(0)} & x_2^{(0)} & x_3^{(0)} \\ x_4^{(0)} & x_5^{(0)} & x_6^{(0)} & x_7^{(0)} \\ x_8^{(0)} & x_9^{(0)} & x_{10}^{(0)} & x_{11}^{(0)} \\ x_{12}^{(0)} & x_{13}^{(0)} & x_{14}^{(0)} & x_{15}^{(0)} \end{pmatrix} \\ &= \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & \nu_0 & \nu_1 & \nu_2 \end{pmatrix}. \end{aligned} \tag{1}$$

The cipher evolves this state through R rounds (typically 20). Each round applies the quarter-round function to all 16 words, mixing them via modular addition, XOR, and rotation operations. To avoid redundancy with the standard literature, we omit the detailed arithmetic steps of the quarter-round function here and refer readers to the work of Bernstein (2008) for the specific ARX sequences.

Table 1. Notational conventions.

Symbol	Definition
X	4×4 ChaCha matrix composed of 16 words
$X^{(0)}$	Initial state matrix of ChaCha/QRE-ChaCha
$X^{(R)}$	Matrix after R rounds
$x \boxplus y$	Modular addition ($x + y \mod 2^{32}$)
$x \boxminus y$	Modular subtraction ($x - y \mod 2^{32}$)
$x \oplus y$	Bitwise XOR
$x \lll n$	Left rotation by n bits
Δx	XOR difference

After R rounds of mixing, the final 512-bit pseudorandom keystream block Z is generated by adding the final state $X^{(R)}$ to the initial state $X^{(0)}$ word-wise:

$$Z = X^{(0)} + X^{(R)}. \tag{2}$$

This keystream is then XORed with the plaintext to produce the ciphertext.

4. Quantum random number enhanced ChaCha

In order to strengthen the security of the original ChaCha cipher by introducing true randomness, the proposed quantum random number enhanced ChaCha (QRE-ChaCha) modifies the round transformation mechanism. Specifically, quantum random numbers are first XORed with the constant words in the initial state matrix. Then, after each odd-numbered round, additional quantum random numbers are XORed into the first 128 bits of the intermediate state matrix.

The quantum random numbers used in our scheme are derived from a QRNG based on the principles of quantum optics, specifically the measurement of vacuum fluctuations or single-photon arrival times. Unlike classical thermal noise, which is theoretically deterministic if environmental variables are known, the outcome of a quantum measurement is intrinsically unpredictable as dictated by the axioms of quantum mechanics. As discussed by Dhara *et al*. (2014), this intrinsic nature ensures that the observed randomness is not merely a lack of information but a fundamental physical property. This property ensures that even if the internal state of the classical computer is compromised, the future output of the QRNG remains unpredictable.

A direct query to a hardware QRNG for every round operation would introduce significant latency. To mitigate this, we adopt a buffered integration strategy. Quantum random numbers are pre-fetched and stored in a secure, protected memory buffer. The encryption algorithm accesses this buffer via direct memory mapping, ensuring that the throughput of QRE-ChaCha is limited

Algorithm 1. QRE-ChaCha.

Input: Input parameters Matrix X, Rounds R, QRN Q
Output: Output Keystream Z

for $r = 0$ **to** $R - 1$ **do**
  if r is odd **then**
    $(x_0^{(r+1)}, x_4^{(r+1)}, x_8^{(r+1)}, x_{12}^{(r+1)}) \leftarrow QR(x_0^{(r)}, x_4^{(r)}, x_8^{(r)}, x_{12}^{(r)})$
    $(x_1^{(r+1)}, x_5^{(r+1)}, x_9^{(r+1)}, x_{13}^{(r+1)}) \leftarrow QR(x_1^{(r)}, x_5^{(r)}, x_9^{(r)}, x_{13}^{(r)})$
    $(x_2^{(r+1)}, x_6^{(r+1)}, x_{10}^{(r+1)}, x_{14}^{(r+1)}) \leftarrow QR(x_2^{(r)}, x_6^{(r)}, x_{10}^{(r)}, x_{14}^{(r)})$
    $(x_3^{(r+1)}, x_7^{(r+1)}, x_{11}^{(r+1)}, x_{15}^{(r+1)}) \leftarrow QR(x_3^{(r)}, x_7^{(r)}, x_{11}^{(r)}, x_{15}^{(r)})$
  end if
  if r is even **then**
    $(x_0^{(r)}, x_1^{(r)}, x_2^{(r)}, x_3^{(r)}) \leftarrow (x_0^{(r)}, x_1^{(r)}, x_2^{(r)}, x_3^{(r)}) \oplus (q_0^{(r)}, q_1^{(r)}, q_2^{(r)}, q_3^{(r)})$
    $(x_0^{(r+1)}, x_5^{(r+1)}, x_{10}^{(r+1)}, x_{15}^{(r+1)}) \leftarrow QR(x_0^{(r)}, x_5^{(r)}, x_{10}^{(r)}, x_{15}^{(r)})$
    $(x_1^{(r+1)}, x_6^{(r+1)}, x_{11}^{(r+1)}, x_{12}^{(r+1)}) \leftarrow QR(x_1^{(r)}, x_6^{(r)}, x_{11}^{(r)}, x_{12}^{(r)})$
    $(x_2^{(r+1)}, x_7^{(r+1)}, x_8^{(r+1)}, x_{13}^{(r+1)}) \leftarrow QR(x_2^{(r)}, x_7^{(r)}, x_8^{(r)}, x_{13}^{(r)})$
    $(x_3^{(r+1)}, x_4^{(r+1)}, x_9^{(r+1)}, x_{14}^{(r+1)}) \leftarrow QR(x_3^{(r)}, x_4^{(r)}, x_9^{(r)}, x_{14}^{(r)})$
  end if
end for
return $Z = X^{(0)} + X^{(R)}$

only by the CPU's arithmetic speed, not by the generation rate of the QRNG device. As illustrated in Fig. 2, the quantum random number generator produces truly random bits and stores them in a quantum random number memory module, which is subsequently accessed by the QRE-ChaCha cipher. It is worth noting that, since this work focuses solely on the optimization of the ChaCha cipher, the quantum random number generation process itself is not illustrated in detail in Fig. 2, as it merely serves as a service module invoked by the algorithm and can be pre-stored in the memory module.

The detailed optimization strategy of QRE-ChaCha is summarized as follows:

- For the initial state matrix, QRE-ChaCha replaces the 128-bit (4-word) constant (c_0, c_1, c_2, c_3) with the bitwise XOR of this constant and a 128-bit quantum random number q, while the remaining parts use a 256-bit (8-word) secret key, a 96-bit (3-word) nonce, and a 32-bit counter as input, as shown in Eqn. (3), where q denotes the quantum random number:

$$X^{(0)} = \begin{pmatrix} x_0^{(0)} & x_1^{(0)} & x_2^{(0)} & x_3^{(0)} \\ x_4^{(0)} & x_5^{(0)} & x_6^{(0)} & x_7^{(0)} \\ x_8^{(0)} & x_9^{(0)} & x_{10}^{(0)} & x_{11}^{(0)} \\ x_{12}^{(0)} & x_{13}^{(0)} & x_{14}^{(0)} & x_{15}^{(0)} \end{pmatrix}$$

$$= \begin{pmatrix} c_0 \oplus q_0^{(0)} & c_1 \oplus q_1^{(0)} & c_2 \oplus q_2^{(0)} & c_3 \oplus q_3^{(0)} \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & \nu_0 & \nu_1 & \nu_2 \end{pmatrix}. \tag{3}$$

- For each odd-numbered round of the quarter-round (QR) function, QRE-ChaCha modifies the input state matrix (i.e., the output state matrix of the preceding even-numbered round) by XORing the first four words $(x_0^{(r-1)}, x_1^{(r-1)}, x_2^{(r-1)}, x_3^{(r-1)})$ with four quantum random words of the same length $(q_0^{(r-1)}, q_1^{(r-1)}, q_2^{(r-1)}, q_3^{(r-1)})$, and then replacing the original words with the resulting values. This operation enhances the diffusion and randomness of the round function, as shown in Eqn. (4). Here, r starts from 0, so $X^{(r)}$ refers to the input state matrix of an odd-numbered round when r is even:

$$X^{(r=\text{even})} = \begin{pmatrix} x_0^{(r)} & x_1^{(r)} & x_2^{(r)} & x_3^{(r)} \\ x_4^{(r)} & x_5^{(r)} & x_6^{(r)} & x_7^{(r)} \\ x_8^{(r)} & x_9^{(r)} & x_{10}^{(r)} & x_{11}^{(r)} \\ x_{12}^{(r)} & x_{13}^{(r)} & x_{14}^{(r)} & x_{15}^{(r)} \end{pmatrix}$$

$$= \begin{pmatrix} x_0^{(r)} \oplus q_0^{(r)} & x_1^{(r)} \oplus q_1^{(r)} & x_2^{(r)} \oplus q_2^{(r)} & x_3^{(r)} \oplus q_3^{(r)} \\ x_4^{(r)} & x_5^{(r)} & x_6^{(r)} & x_7^{(r)} \\ x_8^{(r)} & x_9^{(r)} & x_{10}^{(r)} & x_{11}^{(r)} \\ x_{12}^{(r)} & x_{13}^{(r)} & x_{14}^{(r)} & x_{15}^{(r)} \end{pmatrix}. \tag{4}$$

The complete QRE-ChaCha is described as Algorithm 1.

In addition, to ensure the injected quantum randomness effectively propagates across the entire state matrix, we assume that the difference of the quantum random numbers is not equal to the difference of the corresponding input word to the quarter-round function,

$$\Delta q_i^{(r)} \neq \Delta x_a^{(r)},$$

where $i = 0, 1, 2, 3$, r denotes the r-th round of the QRE-ChaCha cipher and $x_a^{(r)}$ represents the first input word of the quarter-round function. This constraint is an integral part of the algorithm design.

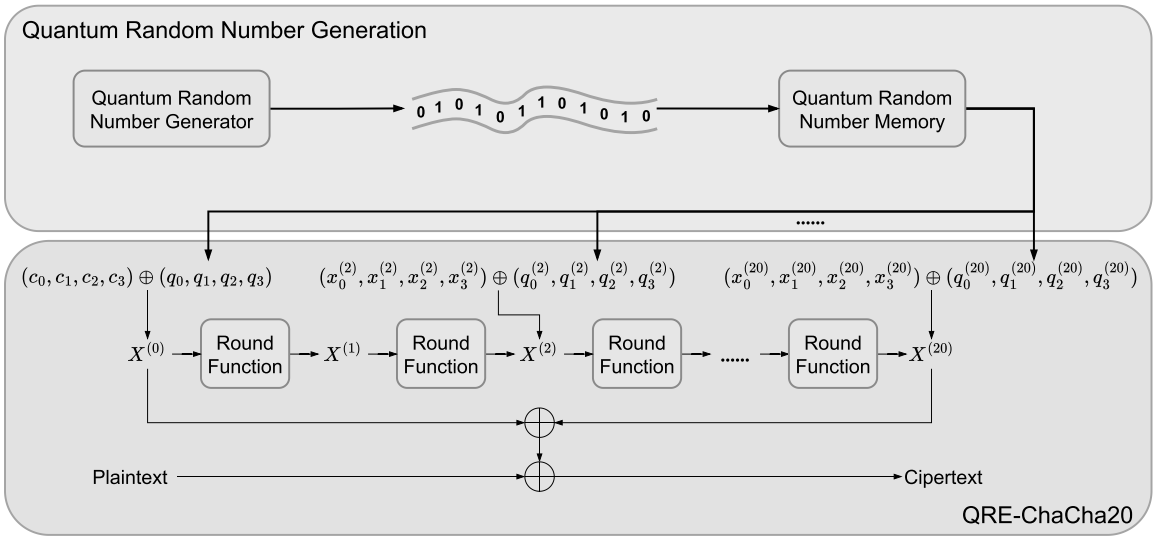

Fig. 2. Overall process of the 20-round QRE-ChaCha cipher.

Without loss of generality, it is important to note that the quantum random numbers used in the cipher are assumed to be confidential to adversaries. The distribution method of these quantum random numbers is not specified in this work. In practice, to ensure the security of quantum random number transmission, a recommended approach is to adopt the method used in the work of Li *et al*. (2023), where the generated random numbers are authenticated using a post-quantum cryptographic (PQC) signature scheme. For the 20-round version of the algorithm, a total of 1280 bits of random numbers need to be distributed to the user through a PQC scheme during the establishment of the secure communication channel. Such an overhead is affordable for PQC. For instance, the public key length of the CRYSTALS-Kyber (ML-KEM) algorithm is 6400 bits for the lowest security level (Kyber512) (NIST, 2024). This approach provides a certain level of resistance against quantum attacks during the distribution process.

5. Security analysis

5.1. Quantum randomness analysis. The design of QRE-ChaCha is inspired by the non-interactive zero-knowledge proof (NIZKP) protocol based on device-independent quantum randomness, as proposed by Li *et al*. (2023). In this work, the security of classical cryptographic schemes is improved by leveraging the intrinsic unpredictability of quantum-generated randomness. Similarly, we incorporate quantum random numbers into the ChaCha cipher to strengthen its cryptographic security.

Quantum random numbers are generated based on fundamental principles of quantum mechanics, which are inherently unpredictable and truly random. In contrast, conventional cryptographic systems typically rely on pseudorandom number generators (PRNGs). Although PRNGs can produce uniformly distributed outputs, they are fundamentally deterministic and lack intrinsic randomness. This distinction is critical, since, as Dhara *et al*. (2014) argued, observed randomness does not necessarily imply full intrinsic randomness unless certified by quantum non-locality or similar physical principles. Moreover, as pointed out by Herrero-Collantes and Garcia-Escartin (2017), uniformity alone is far from sufficient in modern cryptographic applications. Random numbers are now expected to satisfy at least two additional properties: unpredictability (forward security) and backward security. Since PRNGs are inherently deterministic, they cannot provide true randomness and thus may fall short in satisfying these essential security requirements.

For quantum random numbers, their true randomness originates from quantum processes that disrupt coherent superposition states (Ma *et al*., 2016). In the most widely used practical QRNGs based on photonic systems, a single photon can carry one quantum bit (qubit), which may be viewed as a linear superposition of the classical bit values 0 and 1, expressed as $(|0\rangle + |1\rangle)/\sqrt{2}$. Upon measurement, the qubit collapses into either 0 or 1 with equal probability (50%), thus producing a genuinely random binary outcome. In practical implementations, as the example shown in the QRNG part of Fig. 1, the photon is initially prepared in a superposition of horizontal (H) and vertical (V) polarization states, denoted as $(|H\rangle + |V\rangle)/\sqrt{2}$. A polarization beam splitter (PBS) is used to transmit horizontally polarized photons and reflect vertically polarized ones. Two single-photon detectors (SPDs), positioned at the output ports of the PBS, are

used to measure the outcome. This configuration enables the generation of random bits with theoretically perfect randomness, where the unpredictability is fundamentally guaranteed by the laws of quantum physics.

This intrinsic unpredictability is particularly critical for preventing nonce reuse attacks, a major vulnerability in stream ciphers like ChaCha. As summarized by Henry (2024), QRNGs produce truly random and non-reproducible values, making it computationally infeasible to predict future outputs based on past values. In contrast, PRNGs generate sequences based on an initial seed; if the seed is weak or reused, or the system state is reset, the nonce sequence repeats, rendering the cipher insecure. By deriving the nonce or initial injection from a QRNG, the probability of a collision is physically negligible (approaching 2^{-128}), independent of the computational state. Furthermore, QRNGs use quantum entropy sources, where any attempt to probe or tamper with the system inherently disturbs the quantum state, thus making such attacks detectable. Recent experimental demonstrations, such as the traceable randomness generation by Kavuri *et al*. (2025), have shown that it is possible to verify the origin and quality of randomness even in complex setups. It is also important to note that while quantum sources provide intrinsic entropy, practical implementations utilize post-processing algorithms and real-time health monitoring to filter out potential biases or statistical anomalies, ensuring the output strictly conforms to a uniform distribution.

Therefore, the true randomness provided by QRNGs significantly enhances the strength of cryptographic keys, reduces the effectiveness of statistical attacks and cryptanalysis that exploit key generation patterns, and adds an extra layer of security to cryptographic systems. Additionally, as shown by Pandey and Jenef (2024), QRNGs demonstrate substantial advantages in randomness, uniformity, and resistance to correlation-based attacks through comprehensive statistical evaluations, including the NIST, Diehard, and ENT test suites, particularly in scenarios requiring high bit-rate random number generation.

In summary, by integrating quantum random numbers into both the seed initialization and the round function, QRE-ChaCha leverages the intrinsic unpredictability and high entropy of quantum randomness. This design not only improves the statistical quality of the generated keystreams, but also enhances the cipher's robustness against differential cryptanalysis and potential quantum adversaries, thereby reinforcing its theoretical cryptographic security.

5.2. Differential cryptanalysis. In this section, to demonstrate the enhanced security of the proposed QRE-ChaCha against differential attacks, we conduct a detailed cryptanalysis of its reduced-round versions and present the corresponding analysis process and experimental results. Despite the wide range of available cryptanalytic techniques, differential analysis remains one of the most widely adopted methods in symmetric cipher evaluation. Due to practical limitations such as computational resources and device constraints, our differential analysis focuses on the 2-round and 3-round versions of QRE-ChaCha. Nevertheless, the results are sufficient to demonstrate that proposed cipher exhibits improved resistance to differential attacks.

Theoretically, the improved resistance of QRE-ChaCha against differential cryptanalysis stems from the probabilistic decorrelation introduced by intrinsic randomness, which fundamentally alters the differential propagation model. In classical differential analysis, an attacker exploits a deterministic dependency where a fixed input difference Δx propagates through the round function to an output difference Δy with probability $\Pr(\Delta x \to \Delta y) > 2^{-n}$.

However, QRE-ChaCha transforms the cipher into a probabilistic encryption scheme. For any differential pair of states (x, x^*) with difference $\Delta x = x \oplus x^*$, the state update injects two statistically independent quantum random vectors q and q^*, respectively. The state update becomes $x' = x \oplus q$ and $x^{*'} = x^* \oplus q^*$. Consequently, the effective difference entering the next transformation is governed by

$$\Delta x' = x' \oplus x^{*'} = (x \oplus x^*) \oplus (q \oplus q^*) = \Delta x \oplus \Delta q, \quad (5)$$

where $\Delta q = q \oplus q^*$ represents the differential quantum noise. Since q and q^* are derived from a true random number generator (TRNG), Δq is uniformly distributed over the state space. Under the design constraint $\Delta q \neq \Delta x$, the term Δq acts as a random mask that prevents the cancellation of the injected entropy.

This introduces a randomization of the differential input. The probability of observing a specific output difference Δy is no longer determined by a single path, but is the convolution of the cipher's differential probability profile with the distribution of the quantum noise:

$$\Pr_{\text{QRE}}[\Delta x \to \Delta y] = \sum_{\delta \in \{0,1\}^w} \Pr_{\text{ChaCha}}[(\Delta x \oplus \delta) \to \Delta y] \times \Pr[\Delta q = \delta]. \quad (6)$$

Here, δ represents the realization of the quantum difference. Because Δq is unpredictable and varies per execution, the high-probability differential trails where $\Pr_{\text{ChaCha}}$ is high for a specific input are essentially averaged out by the uniform distribution of Δq. This forces the effective differential probability to decay exponentially towards the uniform probability 2^{-w}, rendering classical differential distinguishers ineffective.

To quantitatively evaluate this resistance, we adopt the theoretical framework proposed by Fu *et al*. (2016), which models the differential characteristics and linear approximations of modular addition operations in ARX ciphers using linear inequalities under the assumptions of independently distributed inputs and independent rounds . Based on this framework, we utilize the Boolean satisfiability problem (SAT)-based automated search method to explore optimal differential characteristics for the reduced-round versions of QRE-ChaCha. The specific differential model and testing tool used are based on the open-source CryptoSMT project developed by Kölbl (n.d.). In addition, to obtain the quantum random numbers required for the experiments, we utilized the API provided by the ETH ZÜRICH QRNG. It is worth noting that, while the specific access link for this service has become inactive as of the time of writing, the experimental data derived from it remains valid. For reproducibility purposes, researchers may utilize other publicly available quantum randomness services that operate on comparable physical principles, such as the ANU QRNG (ANU, 2011) or Cisco Outshift QRNG (Cisco, 2024). It is important to emphasize that these online sources are used here solely for conceptual and theoretical validation. In a practical, high-security deployment, the system would not rely on public APIs but would instead integrate a dedicated, on-premise hardware quantum random number generator to ensure both confidentiality and high-throughput generation.

In order to analyze the specific contribution of quantum random numbers to the security of QRE-ChaCha, we selected 10 independently generated pairs of quantum random numbers and computed the differential values between each pair. These differentials were used as fixed constraints in the automated differential search process. Based on this setup, we determined the optimal differential trails and corresponding differential probabilities after two and three rounds of iteration. The upper bounds of the differential probabilities are illustrated in Fig. 3. The upper bound of the 2-round differential probability remains stable at approximately 2^{-4}, while the 3-round differential probability fluctuates between 2^{-24} and 2^{-53}. Furthermore, we computed the average over the 10 sets to obtain representative results, with the 2-round average upper bound at approximately 2^{-4} and the 3-round counterpart at around 2^{-25}.

Based on the final search results and subsequent averaging, as shown in Table 2, the 2-round average upper bound on the differential trail probability for the QRE-ChaCha is approximately 2^{-4}, while that of the 3-round case is approximately 2^{-25}. Therefore, the number of effective differential trails (with $p > 2^{-512}$) does not exceed $3{\times}20 + 2{\times}3 = 66$ rounds. The upper bound on the 20-round differential trail probability is approximately 2^{-154}. Under the same testing conditions

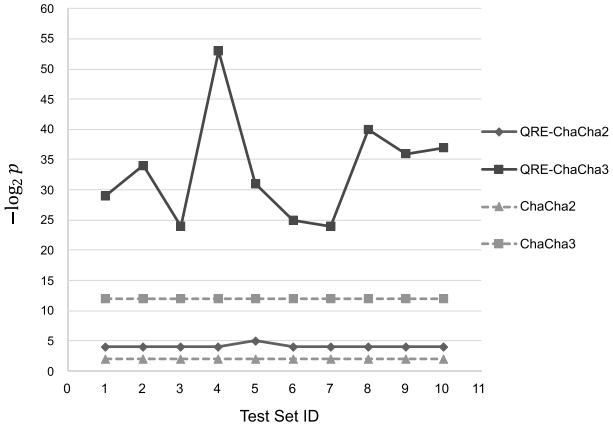

Fig. 3. Differential probabilities of 2-round and 3-round QRE-ChaCha (10 sets).

Table 2. Average differential probabilities of QRE-ChaCha and ChaCha.

Algorithm	Rounds	$\log_2 p$
QRE-ChaCha	2	-4
	3	-25
ChaCha	2	-2
	3	-12

and methodology, the 2-round differential trail probability upper bound for the original ChaCha is 2^{-2}, and for 3 rounds it is 2^{-12}. Accordingly, the number of effective differential trails ($p > 2^{-512}$) for ChaCha does not exceed $3{\times}42 + 2{\times}4 = 134$ rounds, and the upper bound on the 20-round probability is 2^{-74}. Hence, under the analysis framework adopted in this work, QRE-ChaCha demonstrates significantly stronger resistance against differential cryptanalysis compared to the original ChaCha.

6. Statistical randomness testing

To further validate the quality of QRE-ChaCha's output, we comprehensively evaluate the randomness characteristics using the NIST Statistical Test Suite and the Chinese cryptographic randomness test standard in this section. Specifically, the tests are performed using NIST SP 800-22 Rev. 1 (Rukhin *et al*., 2010), and the open-source randomness toolkit project (Trisia, 2022).

The NIST Statistical Test Suite is a widely adopted standard consisting of 15 tests designed to assess the randomness of binary sequences generated by hardware- or software-based cryptographic random or pseudorandom number generators. The Chinese cryptographic randomness test conforms to the national standard GM/T 0005-2021: Specification for Randomness Testing (CISTC, 2021), which also includes 15 tests.

Among them, 11 are consistent with those in the NIST suite, including the frequency test, block frequency test, runs test, longest run of ones in a block test, binary matrix rank test, discrete Fourier transform test, Maurer's universal statistical test, linear complexity test, overlapping template matching test, approximate entropy test, and cumulative sums test. Although the same tests are used, the two suites may differ slightly in implementation details. In addition, the Chinese standard includes four specialized test items: the poker test, run distribution test, binary derivation test, and autocorrelation test. Together, these two test suites ensure that the generated random number sequences exhibit strong statistical randomness, thereby meeting the requirements of cryptographic and other randomness-dependent applications. The test results demonstrate that QRE-ChaCha introduces no flaws in the randomness of its keystream. The overall testing procedure adopted in this work is summarized as follows:

- The test is based on the 8-round version of QRE-ChaCha, which is the minimum number of rounds permitted under our evaluation criteria. Using randomly generated seed keys, we encrypt identical plaintexts with QRE-ChaCha8 to produce $10{,}000$ keystream sequences, each with a length of 10^6 bits.

- These $10{,}000$ keystream sequences are subjected to both the NIST Statistical Test Suite and the Chinese National Cryptographic Randomness Test Suite. During testing, all parameters recommended by the NIST and the Chinese standard are adopted. The significance level for both tests is set to 0.01 and the uniformity significance level to 0.0001. To address the necessity of a baseline comparison, we performed the exact same tests on ChaCha8. The results are analyzed to determine whether the keystreams generated by QRE-ChaCha exhibit statistically strong randomness. The comparative test results from both suites are summarized in Tables 3 and 4.

To constrain the file size during testing, only $1{,}000$ keystream sequences, each with a length of 10^6 bits, were used for the NIST randomness evaluation. Table 3 presents selected results. For the Non-overlapping Template test, only the result corresponding to the template with the minimum number of samples passing the significance level is shown. Results from the Random Excursions and Random Excursions Variant tests are omitted due to the large number of sub-tests, but the keystreams successfully passed all items within these two test categories.

As presented in Tables 3 and 4, both ChaCha8 and QRE-ChaCha8 successfully pass all test items with high P-values. Statistically, the output distributions of both algorithms are indistinguishable from a uniform random source. This result confirms that the integration of quantum random numbers via XOR injection preserves the excellent statistical properties of the original ChaCha cipher without introducing any bias or structural flaws.

However, it is worth noting that statistical test suites operate on the generated bit strings and can only evaluate observed randomness. They cannot distinguish between a deterministic PRNG and a true quantum source. The fact that the P-values are similar does not imply that the security levels are identical. The true advantage of QRE-ChaCha lies in its intrinsic randomness, which is verified by the physical generation process of the QRNG, as discussed in Sections 4 and 5. While the original ChaCha's output is computationally determined by its seed and thus vulnerable if the seed is compromised, the QRE-ChaCha stream contains entropy that is theoretically unpredictable due to the laws of quantum physics. Therefore, the similar test results serve to validate the robustness of our modification, while the security enhancement is derived from the unpredictable nature of the quantum source itself.

7. Performance evaluation

In this section, to quantify the computational efficiency of QRE-ChaCha, we measure its performance by timing the encryption of fixed-size files. The testing environment is configured as follows: an AMD Ryzen 7 5700U processor with Radeon Graphics, clocked at 1.80 GHz, running a 64-bit Windows 10 Enterprise Edition (version 22H2), with 16 GB of RAM in an x64 architecture. The encryption and decryption operations were implemented at the software level using the C programming language.

In the performance evaluation, we used the 8-round versions of both QRE-ChaCha and ChaCha for encryption and decryption comparison tests, with the 20-round version of ChaCha included as a reference. For each cipher, randomly generated seed keys were used to encrypt the same files of sizes 10 MB, 20 MB, 30 MB, 40 MB, and 50 MB. Each file size was tested five times, and the average encryption time was taken as the final performance metric. The results are summarized in Table 5. Based on the performance test results, the encryption time of QRE-ChaCha8 is almost identical to that of ChaCha8.

It is worth noting that the performance evaluation presented in this work does not account for the time required to generate quantum random numbers. This is because modern QRNGs are capable of delivering secure random numbers at rates exceeding 20 Gbps (Bertapelle *et al*., 2025; Bian *et al*., 2025), which can be pre-stored in the memory module of local devices or online servers, rendering their impact on overall encryption time negligible. Therefore, the performance tests conducted

Table 3. NIST randomness test results of the QRE-ChaCha8 and ChaCha8 keystreams (1,000 sets).

NIST test item	QRE-ChaCha8		ChaCha8		Result
	Pass count	P-value	Pass count	P-value	
Frequency	982	0.187581	988	0.467322	Pass
Block Frequency	988	0.751866	993	0.388990	Pass
Cumulative Sums	983	0.435430	988	0.353733	Pass
Runs	994	0.062821	990	0.781106	Pass
Longest Run of Ones	984	0.747898	989	0.664168	Pass
Rank	990	0.784927	991	0.763677	Pass
FFT	986	0.803720	987	0.281232	Pass
Non-overlapping Template	982	0.940080	980	0.989055	Pass
Overlapping Template	990	0.117432	991	0.286836	Pass
Universal Statistical	990	0.012829	989	0.236810	Pass
Approximate Entropy	990	0.345650	992	0.408275	Pass
Serial	992	0.899171	986	0.383827	Pass
Linear Complexity	987	0.115387	986	0.794391	Pass

in this work focus solely on the algorithmic structure. The results demonstrate that the integration of quantum randomness does not degrade the encryption or decryption efficiency of QRE-ChaCha. On the contrary, the algorithm retains the high performance of ChaCha while achieving an improvement in security.

8. Conclusion

We proposed an innovative stream cipher, QRE-ChaCha, which enhances the security of the classical ChaCha cipher by introducing quantum random numbers. The core enhancement was achieved through a two-stage process. Firstly, quantum random numbers were XORed into the initial constant block. This operation injects intrinsic entropy into the state initialization, effectively preventing the nonce reuse vulnerability caused by deterministic PRNG failures in classical systems. Secondly, during each odd-numbered round, the first 128 bits of the intermediate state were further strengthened with additional quantum random numbers, also via XOR. These periodic injections, combined with the strong diffusion properties of the round function, ensure that the true randomness from the quantum source permeates the entire state matrix, thereby improving the cipher's overall cryptographic security.

In terms of security analysis, this work first examined the theoretical security advantages of QRE-ChaCha by analyzing the physical principles of quantum random number generation and attack resilience. Subsequently, differential cryptanalysis was performed using an SAT-based automated search approach. Compared with the original ChaCha cipher, QRE-ChaCha demonstrates significantly enhanced resistance to differential attacks. Specifically, the upper bounds of the average differential trail probabilities for two and three rounds of QRE-ChaCha are 2^{-4} and 2^{-25}, respectively, whereas those of the original ChaCha are 2^{-2} and 2^{-12}. For 20 rounds, QRE-ChaCha achieves an upper bound of 2^{-154}, considerably lower than the 2^{-74} bound observed in ChaCha.

Regarding keystream randomness, we evaluated QRE-ChaCha using both the NIST Statistical Test Suite and the GM/T 0005-2021 standard. Comparative testing shows that QRE-ChaCha successfully passes all tests just as the original ChaCha does. This result confirms that, while QRE-ChaCha maintains the excellent observed randomness of the original cipher, it provides superior intrinsic randomness and forward security derived from the physical quantum source, a property that classical algorithms cannot theoretically achieve.

For performance evaluation, this work measured the encryption speed of QRE-ChaCha on input files of varying sizes to assess its overall encryption and decryption efficiency. Experimental results indicated that QRE-ChaCha maintains high performance. Specifically, the encryption time of QRE-ChaCha8 is nearly identical to that of ChaCha8, suggesting that the integration of quantum random numbers introduces no noticeable performance overhead. This observation holds consistently across encryption tasks involving files ranging from 10 MB to 50 MB in size.

In summary, the proposed QRE-ChaCha cipher preserves the high performance of the original ChaCha while significantly enhancing its security through the integration of quantum random numbers. The improvements are evident in both the cipher's resistance to differential attacks and the high randomness quality of its keystream. Performance evaluations further confirm the cipher's practical viability. Moreover, QRE-ChaCha can also be regarded as a quantum randomness expansion

Table 4. GM/T 0005-2021 randomness test results of QRE-ChaCha8 and ChaCha8 keystreams (10,000 sets).

GM/T 0005-2021 test item	QRE-ChaCha8		ChaCha8		Result
	Pass count	P-value	Pass count	P-value	
Single Bit Frequency	9884	0.862398	9892	0.438383	Pass
Block Frequency ($m = 10000$)	9902	0.969009	9898	0.588514	Pass
Poker Test ($m = 4$)	9889	0.469806	9892	0.636911	Pass
Poker Test ($m = 8$)	9910	0.362434	9899	0.601351	Pass
Overlapping Template ($m = 3$, P1)	9890	0.978538	9897	0.768750	Pass
Overlapping Template ($m = 3$, P2)	9901	0.211848	9897	0.127393	Pass
Overlapping Template ($m = 5$, P1)	9918	0.610070	9906	0.279706	Pass
Overlapping Template ($m = 5$, P2)	9915	0.906880	9913	0.225644	Pass
Total Runs	9915	0.113239	9907	0.401375	Pass
Run Distribution	9900	0.399442	9888	0.645657	Pass
Max Run of 1s ($m = 10000$)	9900	0.386748	9873	0.044797	Pass
Max Run of 0s ($m = 10000$)	9902	0.650860	9898	0.140054	Pass
Binary Derivation ($k = 3$)	9905	0.699313	9897	0.090826	Pass
Binary Derivation ($k = 7$)	9889	0.669151	9907	0.633579	Pass
Autocorrelation ($d = 1$)	9915	0.073281	9907	0.390374	Pass
Autocorrelation ($d = 2$)	9898	0.187378	9895	0.714252	Pass
Autocorrelation ($d = 8$)	9902	0.128354	9896	0.762307	Pass
Autocorrelation ($d = 16$)	9893	0.846168	9898	0.456314	Pass
Matrix Rank	9902	0.008056	9884	0.040318	Pass
Cumulative Sums (Forward)	9885	0.394370	9897	0.135487	Pass
Cumulative Sums (Backward)	9889	0.447116	9896	0.640035	Pass
Approximate Entropy ($m = 2$)	9890	0.981469	9897	0.802608	Pass
Approximate Entropy ($m = 5$)	9915	0.216485	9902	0.981258	Pass
Linear Complexity ($m = 500$)	9882	0.526907	9878	0.878465	Pass
Linear Complexity ($m = 1000$)	9887	0.155238	9896	0.586861	Pass
Maurer Universal ($L = 7$, $Q = 1280$)	9892	0.621922	9874	0.017151	Pass
DFT ($m = 500$)	9892	0.294959	9890	0.331564	Pass

Table 5. Encryption time comparison of QRE-ChaCha8, ChaCha8, and ChaCha20.

File size	Encryption time (s)		
	QRE-ChaCha8	ChaCha8	ChaCha20
10 MB	0.1037854	0.1051830	0.2025330
20 MB	0.2096104	0.2115156	0.4061916
30 MB	0.3118018	0.3147998	0.6116970
40 MB	0.4168038	0.4228406	0.8162400
50 MB	0.5273160	0.5308238	1.0211580

scheme, offering new prospects for broader applications of quantum-generated randomness in cryptographic and computational contexts.

Acknowledgment

This work was supported by the Quantum Science and Technology–National Science and Technology Major Project (grant no. 2024ZD0302200) and the Zhejiang Provincial Natural Science Foundation of China (grant no. LQ24A050005).

References

Acín, A., Brunner, N., Gisin, N., Massar, S., Pironio, S. and Scarani, V. (2007). Device-independent security of quantum cryptography against collective attacks, *Physical Review Letters* **98**(23): 230501, DOI: 10.1103/PhysRevLett.98.230501.

ANU (2011). Quantum random number generator, Australian National University, Canberra, `https://qrng.anu.edu.au/`.

Aumasson, J.-P., Fischer, S., Khazaei, S., Meier, W. and Rechberger, C. (2008). New features of Latin dances: Analysis of Salsa, ChaCha, and Rumba, *Fast Software Encryption: 15th International Workshop, FSE 2008, Lausanne, Switzerland*, pp. 470–488, DOI: 10.1007/978-3-540-71039-4_30.

Barbero, S., Bellini, E. and Makarim, R.H. (2023). Rotational analysis of chacha permutation, *Advances in Mathematics of Communications* **17**(6): 1422–1439, DOI: 10.3934/amc.2021057.

Bellini, E., Gerault, D., Grados, J., Makarim, R.H. and Peyrin, T. (2023). Boosting differential-linear cryptanalysis of ChaCha7 with MILP, *IACR Transac-*

tions on Symmetric Cryptology **2023**(2): 189–223, DOI: 10.46586/tosc.v2023.i2.189-223.

Bennett, C.H. and Brassard, G. (2014). Quantum cryptography: Public key distribution and coin tossing, *Theoretical Computer Science* **560**: 7–11, DOI: 10.1016/j.tcs.2014.05.025.

Bernstein, D.J. (2008). ChaCha, a variant of Salsa20, *Workshop Record of SASC, Lausanne, Switzerland,* pp. 3–5, https://cr.yp.to/chacha/chacha-20080120.pdf.

Bernstein, D.J., Heninger, N., Lou, P. and Valenta, L. (2017). Post-quantum RSA, *Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands*, pp. 311–329, DOI: 10.1007/978-3-319-59879-6_18.

Bernstein, D.J. and Lange, T. (2017). Post-quantum cryptography, *Nature* **549**(7671): 188–194, DOI: 10.1038/nature23461.

Bertapelle, T., Avesani, M., Santamato, A., Montanaro, A., Chiesa, M., Rotta, D., Artiglia, M., Sorianello, V., Testa, F., De Angelis, G., Contestabile, G., Vallone, G., Romagnoli, M. and Villoresi, P. (2025). High-speed source-device-independent quantum random number generator on a chip, *Optica Quantum* **3**(1): 111–118, DOI: 10.1364/OPTICAQ.529746.

Bian, Y., Yang, J., Jiang, H., Huang, W., Su, Q., Yu, S., Zhang, L., Zhang, Y. and Xu, B. (2025). 20 Gbps real-time source-independent quantum random number generator based on a silicon photonic chip, *Optics Letters* **50**(4): 1216–1219, DOI: 10.1364/OL.544982.

Centellas Claros, L.S., Blanco Coca, L. and Sandoval Alcocer, J.P. (2022). Comparative study of the symmetric cryptography algorithms AES, 3DES and ChaCha20, *Acta Nova* **10**(3): 283–302.

Choudhuri, A.R. and Maitra, S. (2016). Differential cryptanalysis of Salsa and ChaCha—An evaluation with a hybrid model, *Cryptology ePrint Archive*, 2016/377.

Cisco (2024). Outshift quantum random number generator, Cisco, San Jose, https://outshift.cisco.com/quantum-random-number-generator.

CISTC (2021). Randomness testing specification: GM/T 0005-2021, Cryptography Industry Standardization Technical Committee, Beijing, https://std.samr.gov.cn/hb/search/stdHBDetailed?id=E66CC4F6F8D78B7FE05397BE0A0A6C55.

Deepthi, K.K. and Singh, K. (2018). Cryptanalysis of Salsa and ChaCha: Revisited, *International Conference on Mobile Networks and Management, Copenhagen, Denmark,* pp. 324–338, DOI: 10.10007/978-3-319-90775-8_26.

Degabriele, J.P., Govinden, J., Günther, F. and Paterson, K.G. (2021). The security of ChaCha20-Poly1305 in the multi-user setting, *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (online), pp. 1981–2003, DOI: 10.1145/3460120.3484814.

Dhara, C., de la Torre, G. and Acín, A. (2014). Can observed randomness be certified to be fully intrinsic?, *Physical Review Letters* **112**(10): 100402, DOI: 10.1103/PhysRevLett.112.100402.

Fu, K., Wang, M., Guo, Y., Sun, S. and Hu, L. (2016). MILP-based automatic search algorithms for differential and linear trails for speck, *Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany*, pp. 268–288, DOI: 10.1007/978-3-662-52993-5_14.

Ghafoori, N. and Miyaji, A. (2024). Higher-order differential-linear cryptanalysis of ChaCha stream cipher, *IEEE Access* **12**: 13386–13399, DOI: 10.1109/ACCESS.2024.3356868.

Henry, E. (2024). The role of quantum random number generation in enhancing encryption security, *SSRN* 4966139, DOI: 10.2139/ssrn.4966139.

Herrero-Collantes, M. and Garcia-Escartin, J.C. (2017). Quantum random number generators, *Reviews of Modern Physics* **89**(1): 015004, DOI: 10.1103/RevModPhys.89.015004.

Huang, L., Zhou, H., Feng, K. and Xie, C. (2021). Quantum random number cloud platform, *npj Quantum Information* **7**(1): 107.

Iavich, M., Kuchukhidze, T., Iashvili, G. and Gnatyuk, S. (2021). Hybrid quantum random number generator for cryptographic algorithms, *Radioelectronic and Computer Systems* (4): 103–118, DOI: 10.32620/reks.2021.4.09.

Iavich, M., Kuchukhidze, T., Okhrimenko, T. and Dorozhynskyi, S. (2020). Novel quantum random number generator for cryptographical applications, *2020 IEEE International Conference on Problems of Infocommunications: Science and Technology (PIC S&T), Kharkiv, Ukraine*, pp. 727–732, DOI: 10.1109/PICST51311.2020.9467951.

Jain, D.K., Mohan, P., Lakshmanna, K. and Nanda, A.K. (2022). Enhanced data privacy in cyber-physical system using improved Chacha20 algorithm, *Research Square*, rs-1558846, DOI: 10.21203/rs.3.rs-1558846/v1.

Jao, D. and De Feo, L. (2011). Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies, *Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan*, pp. 19–34, DOI: 10.1007/978-3-642-25405-5_2.

Ji, Z., Qiao, Y., Song, F. and Yun, A. (2019). General linear group action on tensors: A candidate for post-quantum cryptography, *Theory of Cryptography Conference, Nuremberg, Germany*, pp. 251–281, DOI: 10.1007/978-3-030-36030-6_11.

Kavuri, G.A., Palfree, J., Reddy, D.V., Zhang, Y., Bienfang, J.C., Mazurek, M.D., Alhejji, M.A., Siddiqui, A.U., Cavanagh, J.M., Dalal, A., Abellán, C., Amaya, W., Mitchell, M.W., Stange, K.E., Beale, P.D., Brandão, L.T.A.N., Booth, H., Peralta, R., Nam, S.W., Mirin, R.P., Stevens, M.J., Knill, E. and Shalm, L.K. (2025). Traceable random numbers from a non-local quantum advantage, *Nature* **642**(8069): 916–921, DOI: 10.1038/s41586-025-09054-3.

Kebande, V.R. (2023). Extended-ChaCha20 stream cipher with enhanced quarter round function, *IEEE Access* **11**: 114220–114237, DOI: 10.1109/ACCESS.2023.3324612.

Kuang, R., Lou, D., He, A., McKenzie, C. and Redding, M. (2021). Pseudo quantum random number generator with quantum permutation pad, *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, (online), pp. 359–364, DOI: 10.1109/QCE52317.2021.00053.

Kumar, S.D., Patranabis, S., Breier, J., Mukhopadhyay, D., Bhasin, S., Chattopadhyay, A. and Baksi, A. (2017). A practical fault attack on ARX-like ciphers with a case study on ChaCha20, *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), Taipei, Taiwan*, pp. 33–40, DOI: 10.1109/FDTC.2017.14.

Langley, A., Chang, W.-T., Mavrogiannopoulos, N., Strombergson, J. and Josefsson, S. (2016). ChaCha20-Poly1305 cipher suites for transport layer security (TLS), *RFC* 7905, DOI: 10.17487/RFC7905.

Li, C., Zhang, K., Zhang, X., Yang, K., Han, Y., Cheng, S., Cui, H., Liu, W., Li, M., Liu, Y., Bai, B., Dong, H.-H., Zhang, J., Ma, X., Yu, Y., Fan, J., Zhang, Q. and Pan, J.-W. (2023). Device-independent quantum randomness-enhanced zero-knowledge proof, *Proceedings of the National Academy of Sciences* **120**(45): e2205463120, DOI: 10.1073/pnas.2205463120.

Lo, H.-K., Curty, M. and Qi, B. (2012). Measurement-device-independent quantum key distribution, *Physical Review Letters* **108**(13): 130503, DOI: 10.1103/PhysRevLett.108.130503.

Lo, H.-K., Ma, X. and Chen, K. (2005). Decoy state quantum key distribution, *Physical Review Letters* **94**(23): 230504, DOI: 10.1103/PhysRevLett.94.230504.

Ma, X., Yuan, X., Cao, Z., Qi, B. and Zhang, Z. (2016). Quantum random number generation, *npj Quantum Information* **2**(1): 1–9, DOI: 10.1038/npjqi.2016.21.

Mahdi, M.S., Hassan, N.F. and Abdul-Majeed, G.H. (2021). An improved chacha algorithm for securing data on IoT devices, *SN Applied Sciences* **3**(4): 429, DOI: 10.1007/s42452-021-04425-7.

Mannalatha, V., Mishra, S. and Pathak, A. (2023). A comprehensive review of quantum random number generators: Concepts, classification and the origin of randomness, *Quantum Information Processing* **22**(12): 439, DOI: 10.1007/s11128-023-04175-y.

Maolood, A.T., Gbashi, E.K. and Mahmood, E.S. (2022). Novel lightweight video encryption method based on ChaCha20 stream cipher and hybrid chaotic map, *International Journal of Electrical & Computer Engineering* **12**(5): 4988–5000, DOI: 10.11591/ijece.v12i5.pp4988-5000.

Micciancio, D. (2011). Lattice-based cryptography, *in* H.C.A. van Tilborg and S. Jajodia (Eds), *Encyclopedia of Cryptography and Security*, Springer, Boston, pp. 713–715.

Najm, Z., Jap, D., Jungk, B., Picek, S. and Bhasin, S. (2018). On comparing side-channel properties of AES and ChaCha20 on microcontrollers, *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Chengdu, China*, pp. 552–555, DOI: 10.1109/APCCAS.2018.8605653.

NIST (2024). Module-lattice-based key-encapsulation mechanism standard, *Federal Information Processing Standards Publication FIPS 203*, National Institute of Standards and Technology, Gaithersburg, DOI: 10.6028/NIST.FIPS.203.

Pandey, S.K. and Jenef, R. (2024). A comparative study and analysis of quantum random number generator with true random number generator, *2024 16th International Conference on Communication Systems & Networks, Bengaluru, India*, pp. 1000–1005, DOI: 10.1109/COMSNETS59351.2024.10426934.

Pirandola, S., Andersen, U.L., Banchi, L., Berta, M., Bunandar, D., Colbeck, R., Englund, D., Gehring, T., Lupo, C., Ottaviani, C., Pereira, J.L., Razavi, M., Shaari, J.S., Tomamichel, M., Usenko, V.C., Vallone, G., Villoresi, P. and Wallden, P. (2020). Advances in quantum cryptography, *Advances in Optics and Photonics* **12**(4): 1012–1236, DOI: 10.1364/AOP.361502.

Portmann, C. and Renner, R. (2022). Security in quantum cryptography, *Reviews of Modern Physics* **94**(2): 025008, DOI: 10.1103/RevModPhys.94.025008.

Procter, G. (2014). A security analysis of the composition of ChaCha20 and Poly1305, *Cryptology ePrint Archive*, Paper 2014/613, `https://eprint.iacr.org/2014/613`.

Renner, R. (2008). Security of quantum key distribution, *International Journal of Quantum Information* **6**(01): 1–127, DOI: 10.1142/S0219749908003256.

Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A. Dray, J. and Vo, S. (2010). A statistical test suite for random and pseudorandom number generators for cryptographic applications, *NIST Special Publication (SP) 800-22, Rev. 1*, National Institute of Standards and Technology, Gaithersburg, DOI: 10.6028/NIST.SP.800-22r1a.

Shi, Z., Zhang, B., Feng, D. and Wu, W. (2013). Improved key recovery attacks on reduced-round Salsa20 and ChaCha, *International Conference on Information Security and Cryptology, Seoul, South Korea*, pp. 337–351, DOI: 10.1007/978-3-642-37682-5_24.

Kölbl, S. (n.d.). CryptoSMT: An easy to use tool for cryptanalysis of symmetric primitives, `https://github.com/kste/cryptosmt`.

Stipcevic, M. (2012). Quantum random number generators and their applications in cryptography, *Advanced Photon Counting Techniques VI, Baltimore, USA,* pp. 20–34, DOI: 10.1117/12.91992.

Szczepanik, W. and Niemiec, M. (2025). Optimizing routing in quantum key distribution networks using the artificial fish swarm algorithm, *International Journal of Applied Mathematics and Computer Science* **35**(4): 667–675, DOI: 10.61822/amcs-2025-0047.

Trisia (2022). Randomness, Version 1.5.0, `https://github.com/Trisia/randomness`.

Zhao, S., Wang, R. and Zhao, Q. (2026). Certifying optimal device-independent quantum randomness in quantum networks, *arXiv* 2601.18534, DOI: 10.48550/arXiv.2601.18534.

Chao Liu holds a BS degree from the School of Cyberspace at Hangzhou Dianzi University. His research interests include quantum computing and quantum cryptography.

Shuai Zhao is currently a lecturer with the School of Cyberspace at Hangzhou Dianzi University. His research interests include quantum information physics and quantum cryptography.

Chenhao Jia holds an MS degree from the School of Cyberspace at Hangzhou Dianzi University. He is currently working toward the PhD degree at Shandong University. His research interests include the analysis and design of symmetric cryptography.

Gengran Hu is currently an associate professor with the School of Cyberspace at Hangzhou Dianzi University. His research interests include lattice-based cryptography, blockchain technology, and data privacy protection.

Tingting Cui is currently an associate professor with the School of Cyberspace at Hangzhou Dianzi University. Her research interests include the analysis and design of symmetric cryptography.

Received: 20 October 2025
Revised: 10 February 2026
Accepted: 7 March 2026