AN IMPROVED CHACHA ALGORITHM BASED ON QUANTUM RANDOM NUMBER

CHAO LIU[1], SHUAI ZHAO*[1,2,3], CHENHAO JIA[1], GENGRAN HU[1,3], AND TINGTING CUI[1,3]

Abstract. 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 the secure implementation of ChaCha cipher. To further strengthen the security of ChaCha cipher, 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 quantum random numbers 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 its encryption speed on files of various sizes. According to the results, the present 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.

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)[1], 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

2020 *Mathematics Subject Classification.* Primary: 94A60; Secondary: 68P25; Tertiary: 81P94.

Key words and phrases. Stream cipher, ChaCha, Quantum random numbers, QRE-ChaCha.

This work was supported by the Zhejiang Provincial Natural Science Foundation of China (Grant No. LQ24A050005); the Innovation Program for Quantum Science and Technology (Grant No. 2024ZD0302200).

*Corresponding author: Shuai Zhao.

techniques have achieved significant progress, the security of conventional cryptographic schemes, including ChaCha, faces more severe challenges. In response to these challenges, prominence has been obtained in two major research directions: post-quantum cryptography[2, 3, 4, 5, 6], which focuses on developing new algorithms based on mathematical problems that are conjectured to be intractable for quantum computers, and quantum cryptography[7, 8, 9, 10, 11, 12, 13, 14], 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.

Due to the inherent uncertainty in quantum physics, quantum random number possesses 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[10], which on the one hand can extend the application scenarios of quantum random number generators, and on the other hand can 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 ChaCha stream cipher.

Different from the block cipher, stream ciphers operate on data one bit at a time, enabling faster encryption and decryption as well as inherent parallelism. When the seed key and the pseudorandom numbers exhibit high randomness, the stream cipher can provide strong security guarantees[15]. One prominent stream cipher, ChaCha, designed by Daniel J. Bernstein in 2008 as an enhanced variant of Salsa20[16], exemplifies this design philosophy. Although it retains the structural foundation of Salsa20, ChaCha introduces a modified key matrix and a more intricate round function to enhance diffusion and bolster security against cryptanalytic attacks. This refinement, achieved without a significant performance penalty, has cemented ChaCha's status as a robust and efficient cipher, leading to its widespread adoption in real-time communication and data transmission scenarios[1]. Despite its high performance, widespread adoption, and a robust ARX-based (Addition, Rotation, XOR) design secure against various attacks, numerous security analyses have revealed potential vulnerabilities in the ChaCha cipher[17]. As reviewed in Section 2.2, existing optimization approaches largely focus on modifying the round function or adjusting internal parameters, but these methods often provide only marginal security improvements against key-recovery attacks, particularly those based on differential cryptanalysis.

To address this issue, as illustrated in Figure 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. By introducing intrinsic randomness from quantum random numbers into both the seeds and the round function, 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:

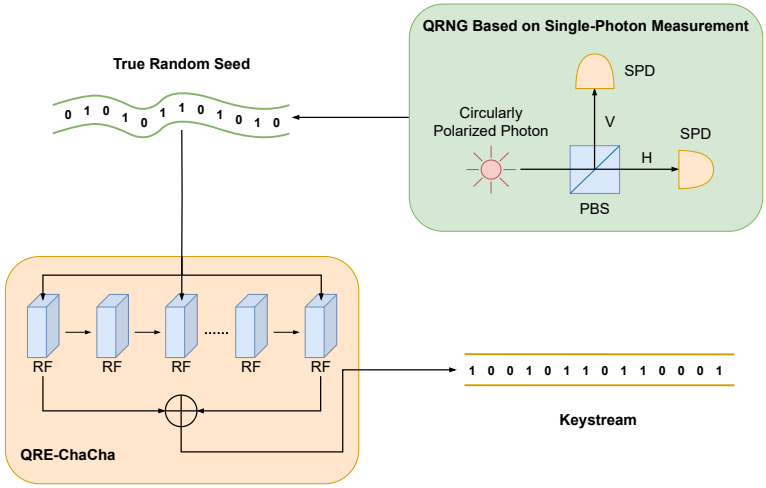

FIGURE 1. QRE-ChaCha: An integration scheme of quantum random numbers with a classical cipher. *SPD* denotes the single-photon detector, *PBS* denotes the polarization beam splitter, *RF* denotes the round function.

(1) To address the suboptimal resistance of existing ChaCha optimizations against differential cryptanalysis, we propose a quantum random number enhanced stream cipher, QRE-ChaCha (Quantum Random Number Enhanced 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 into 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 Figure 4, QRE-ChaCha achieves better differential probabilities for both 2-round and 3-round configurations compared to the original ChaCha cipher, 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 Figure 3, the encryption and decryption speeds of QRE-ChaCha are nearly identical to those of ChaCha.

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.** In the field of quantum random number generation, ensuring the security and reliability of randomness sources has attracted a lot of research interest. As summarized by Mannalatha et al.[18] and Herrero-Collantes et al.[19], quantum random number generators (QRNGs) offer intrinsic unpredictability and true randomness based on quantum physics, which are fundamentally inexplicable by classical theories. Recent advances have led to the development of device-independent (DI) and semi-device-independent (SDI) QRNG models, which aim to minimize or eliminate trust assumptions on the quantum devices. These models strengthen resistance against side-channel attacks by ensuring that the randomness generation process cannot be tampered with or predicted even when parts of the device are untrusted. Ma et al.[20] further classified QRNGs into practical, SDI, and DI types, and analyzed their trade-offs in terms of implementation complexity, generation rate, and provable security. Overall, DI and SDI QRNGs have become key approaches for constructing secure randomness sources under different security assumptions, particularly suitable for enhancing cryptographic primitives that demand high levels of entropy and adversarial robustness.

In terms of application domains for quantum random numbers, Iavich et al.[21] proposed a novel QRNG in 2020 based on photon arrival time, aiming to generate fast and high-quality quantum random numbers at a low cost to meet the demands of cryptographic applications. To combine the inherent randomness of quantum processes with the efficiency of classical pseudorandom number generation, they further introduced a hybrid QRNG[22] in 2021, integrating techniques from time-of-arrival QRNGs, photon-counting QRNGs, and attenuated-pulse QRNGs. It was specifically designed for use in cryptographic algorithms. Experimental results demonstrated its ability to produce high-quality random numbers at a relatively high throughput, making it highly effective in conventional cryptographic applications. Similarly, Stipcevic et al.[23] focused explicitly on the cryptographic context, emphasizing the significance of random number generators in secure systems. By comparing free-running oscillator-based RNGs with QRNGs, their work explored the role of randomness in quantum cryptography, discussed various post-processing

techniques, and proposed evaluation methods, thereby offering guidance on practical QRNG deployment in cryptographic settings.

Distinct from hardware-based QRNGs, Kuang et al.[24] proposed a pseudo-quantum random number generator (pQRNG) based on quantum algorithms. Leveraging the high-entropy properties of quantum permutation spaces, the proposed method uses Quantum Permutation Pad (QPP) techniques to generate pseudorandom numbers with strong unpredictability. Without the need for physical quantum integration, the pQRNG can be readily embedded into classical computing systems, providing a high-quality deterministic randomness source for cryptographic and other applications.

To support real-world adoption of quantum random numbers, Huang et al.[25] developed a practical cloud-based quantum random number service by integrating QRNGs with Alibaba Cloud infrastructure. This platform provides quantum-grade randomness for a variety of applications, including cryptographic systems.

It is worth noting that in 2023, JianWei Pan et al.[10] 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 the QRE-ChaCha.

2.2. **ChaCha cipher.** As a member of the stream cipher family, the ChaCha has been widely adopted due to its high performance and simplicity. To date, most studies have focused on structural analysis and cryptanalytic attacks against the cipher[26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37], while relatively fewer works have explored the optimization of ChaCha's design and performance[38, 39, 40, 41].

With regard to the analysis of the ChaCha, Najm et al.[26] compared AES and ChaCha20 on microcontrollers in order to assess the side-channel resistance of ChaCha. Their results showed that, although ChaCha20 exhibits stronger resistance to side-channel attacks than AES, it incurs higher computational overhead. To reveal potential weaknesses of ChaCha20 in terms of fault resistance, Kumar et al.[27] were the first to introduce a practical fault attack against the ChaCha20 stream cipher and proposed four differential fault analysis techniques. To lay a theoretical foundation for the practical deployment of ChaCha20-Poly1305, Procter et al.[15] analyzed the security of combining ChaCha20 with the Poly1305 authenticator in IETF protocols and provided a security reduction for the authenticated encryption scheme. To further analyze the security of ChaCha20-Poly1305 in multi-user environments, Degabriele et al.[28] proposed an enhanced analytical framework, established tighter security bounds and adversarial models, and identified security limits of the scheme under such settings. To study the rotational properties of the ChaCha permutation, Barbero et al.[29] proposed a distinguishing method based on permutation calls and derived upper and lower bounds on the probability of rotation propagation. However, the authors noted that no effective cryptanalytic applications of these results to the ChaCha stream cipher have yet been found. To compare AES, 3DES, and ChaCha20 in terms of performance

and efficiency, Claros et al.[30] designed an experiment and found that ChaCha20 achieved the fastest encryption and decryption speeds.

Innovative analytical approaches to the ChaCha can be traced back to the introduction of the probabilistic neutral bits (PNB) concept by Aumasson et al.[31] in 2008. This concept was first applied to differential cryptanalysis of the Salsa20 and ChaCha stream ciphers, leading to successful attacks and marking the first time this technique was used in cryptanalysis. Subsequently, to reduce the time and data complexity of attacks against reduced-round variants of Salsa20 and ChaCha, Shi et al.[32] proposed an improved key-recovery method by introducing new types of distinguishers and identifying high-probability second-order differential paths. To evaluate the differential security of Salsa and ChaCha using a mixed analytical framework, Choudhuri et al.[33] developed a hybrid model that applies original nonlinear functions in the initial rounds and linear approximations in the later rounds, concluding that 12 rounds suffice to protect a 256-bit key. To correct flaws in prior analyses and extend attacks to other reduced-round versions, Deepthi et al.[34] conducted a thorough reanalysis of ChaCha20 and Salsa20 and successfully mounted attacks on Salsa20/7, ChaCha6, and ChaCha7. To enhance the effectiveness of PNB-based differential attacks, Miyashita et al.[35] introduced an optimized PNB differential attack, identifying the best differential biases and combinations of neutral bits for ChaCha. Later, to improve the precision of differential-linear cryptanalysis, Bellini et al.[36] expanded the search space, optimized mask selection between differential and linear components, and utilized MILP tools to construct distinguishers against 7, 7.5, and 5 rounds of ChaCha. To explore the potential of higher-order differential-linear attacks on ChaCha, Ghafoori et al.[37] analyzed multiple round configurations and discovered new differential biases and linear approximations, further advancing the complexity and scope of ChaCha cryptanalysis.

In the domain of ChaCha optimization, several efforts have been made to enhance its security and adaptability across different application scenarios. To improve the security of ChaCha while ensuring low power consumption for IoT devices, Mahdi et al.[38] proposed an enhanced variant called Super ChaCha, which modifies the rotation process and the input update order to increase resistance against cryptanalytic attacks. Similarly, to enhance data privacy on the Internet of Things platforms, Jain et al.[39] presented an optimized version of ChaCha20 tailored for secure communication in constrained environments. To strengthen the security of ChaCha20 through structural improvements, Kebande et al.[40] introduced EChaCha20, which enhances the quarter-round function (QR-F) and incorporates 32-bit input words along with ARX operations. To apply ChaCha in lightweight multimedia encryption, Maolood et al.[41] proposed a novel video encryption scheme that integrates the ChaCha20 stream cipher with hybrid chaotic mapping theory, achieving efficiency and lightweight security suitable for real-time video processing.

3. Preliminaries

To facilitate subsequent reading and understanding, we first present the relevant notational conventions, as summarized in Table 1. Then, a brief overview of the ChaCha cipher is provided.

ChaCha is a pseudorandom function based on the ARX structure. It updates its internal state matrix by performing four modular additions, four XOR operations, and four bitwise rotations. Compared to Salsa, ChaCha updates each word twice

TABLE 1. Notation Conventions.

Symbol	Definition
X	A 4×4 ChaCha matrix composed of 16 words
$X^{(0)}$	Initial state matrix of ChaCha/QRE-ChaCha
$X'^{(0)}$	Related matrix with a one-bit difference at position $x_{i,j}$
$X^{(R)}$	ChaCha/QRE-ChaCha matrix after R rounds
$X^{(r)}$	ChaCha/QRE-ChaCha matrix after r rounds, where $R > r$
$x_i^{(R)}$	The i-th word of the state matrix $X^{(R)}$
$x_{i,j}^{(R)}$	The j-th bit of the i-th word in $X^{(R)}$
p	Probability of a differential path
$x \boxplus y$	Modular addition of words x and y
$x \boxminus y$	Modular subtraction of words x and y
$x \oplus y$	Bitwise XOR of words x and y
$x \lll n$	Left rotation of word x by n bits
Δx	XOR difference between word x and x'
ChaCha n	ChaCha stream cipher at round n
QRE-ChaCha n	QRE-ChaCha stream cipher at round n

per round rather than once, which enhances diffusion across the state matrix and improves resistance to cryptanalysis. ChaCha follows the same design principles as Salsa, using 32 bits (one word) to construct a 512-bit (16-word) initial state matrix, which consists of four constant words ($c_1 = 0x61707865$, $c_2 = 0x3320646e$, $c_3 = 0x79622d32$, $c_4 = 0x6b206574$), eight seed key words, three nonce words, and one counter word.

As an iterative stream cipher, the number of rounds in ChaCha can be selected based on the desired level of security and performance. For maximum security, 20 rounds are used; for maximum speed, 8 rounds may be chosen; and for a trade-off between the two, 12 rounds are commonly applied[16]. In each round, the ChaCha round function (R) is composed of four quarter-round functions (QR), each of which takes four input words $(x_a^{(r)}, x_b^{(r)}, x_c^{(r)}, x_d^{(r)})$.

The initial state matrix of ChaCha is shown in Equation (1):

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

In Equation (1), c denotes the constant, k denotes the seed key, t denotes the counter, and v denotes the nonce.

The overall structure of the quarter-round function is illustrated in Figure 2. Since the four input words of each quarter-round function in a round are distinct, ChaCha's round function allows all four quarter-rounds to be executed in parallel. For odd-numbered rounds (column rounds), the quarter-round functions operate on the following four column vectors respectively: $(x_0^{(r)}, x_4^{(r)}, x_8^{(r)}, x_{12}^{(r)})$, $(x_1^{(r)}, x_5^{(r)}, x_9^{(r)}, x_{13}^{(r)})$, $(x_2^{(r)}, x_6^{(r)}, x_{10}^{(r)}, x_{14}^{(r)})$, $(x_3^{(r)}, x_7^{(r)}, x_{11}^{(r)}, x_{15}^{(r)})$. For even-numbered

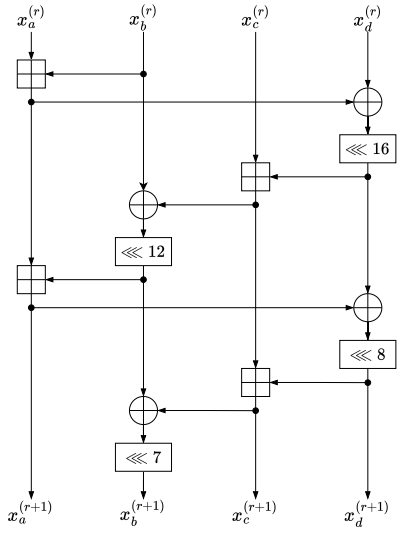

FIGURE 2. The quarter-round function structure of the ChaCha cipher.

rounds (diagonal rounds), the quarter-round functions are applied to the following four diagonal vectors respectively: $(x_0^{(r)}, x_5^{(r)}, x_{10}^{(r)}, x_{15}^{(r)})$, $(x_1^{(r)}, x_6^{(r)}, x_{11}^{(r)}, x_{12}^{(r)})$, $(x_2^{(r)}, x_7^{(r)}, x_8^{(r)}, x_{13}^{(r)})$, $(x_3^{(r)}, x_4^{(r)}, x_9^{(r)}, x_{14}^{(r)})$.

Each quarter-round updates the internal state matrix $X^{(r)}$ by sequentially applying the operations defined in Equation (2) to its four input words.

$$\begin{cases} x_{a'}^{(r)} = x_a^{(r)} \boxplus x_b^{(r)} & x_{d'}^{(r)} = x_d^{(r)} \oplus x_{a'}^{(r)} & x_{d''}^{(r)} = x_{d'}^{(r)} \lll 16 \\ x_{c'}^{(r)} = x_c^{(r)} \boxplus x_{d''}^{(r)} & x_{b'}^{(r)} = x_b^{(r)} \oplus x_{c'}^{(r)} & x_{b''}^{(r)} = x_{b'}^{(r)} \lll 12 \\ x_a^{(r+1)} = x_{a'}^{(r)} \boxplus x_{b''}^{(r)} & x_{d'''}^{(r)} = x_{d''}^{(r)} \oplus x_a^{(r+1)} & x_d^{(r+1)} = x_{d'''}^{(r)} \lll 8 \\ x_c^{(r+1)} = x_{c'}^{(r)} \boxplus x_d^{(r+1)} & x_{b'''}^{(r)} = x_{b''}^{(r)} \oplus x_c^{(r+1)} & x_b^{(r+1)} = x_{b''}^{(r)} \lll 7 \end{cases} . \quad (2)$$

For the n-round version of ChaCha, the final 512-bit pseudorandom keystream block Z is computed as:

$$Z = X^{(0)} + X^{(n)}.$$

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.

As illustrated in Figure 3, 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

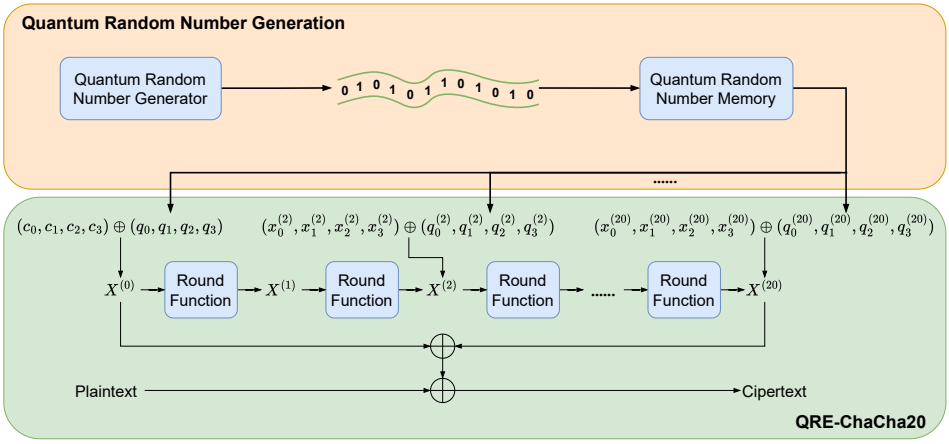

FIGURE 3. The overall process of the 20-round QRE-ChaCha cipher.

this work focuses solely on the optimization of the ChaCha cipher, the quantum random number generation process itself is not illustrated in detail in Figure 3, 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 Equation (3), where q denotes the quantum random number.
- 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 Equation (4). Here, r starts from 0, so $X^{(r)}$ refers to the input state matrix of an odd-numbered round when r is even. The complete QRE-ChaCha algorithm is described in Algorithm 1.

$$
\begin{aligned}
X^{(0)} &= \begin{pmatrix} x_0^{(0)} & x_1^{(0)} & x_2^{(0)} & x_3^{(0)} \\ x_4^{(0)} & x_5^{(0)} & x_6^{(0)} & x_7^{(0)} \\ x_8^{(0)} & x_9^{(0)} & x_{10}^{(0)} & x_{11}^{(0)} \\ x_{12}^{(0)} & x_{13}^{(0)} & x_{14}^{(0)} & x_{15}^{(0)} \end{pmatrix} \\
&= \begin{pmatrix} c_0 \oplus q_0^{(0)} & c_1 \oplus q_1^{(0)} & c_2 \oplus q_2^{(0)} & c_3 \oplus q_3^{(0)} \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & \nu_0 & \nu_1 & \nu_2 \end{pmatrix}.
\end{aligned} \tag{3}
$$

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

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 $Z = X^{(0)} + X^{(R)}$

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

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

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.

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 [10], where the generated random numbers are authenticated using a post-quantum cryptographic (PQC) signature

scheme. 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 in [10]. 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 in our work.

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, as pointed out in [19], 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[20]. 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 Figure 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.

In terms of attack testing against random number generators, as summarized in [42], QRNGs produce truly random and non-reproducible values, making it impossible to predict future outputs based on past values. In contrast, PRNGs generate sequences based on an initial seed; if the seed is weak or compromised, the entire sequence becomes predictable and vulnerable to attack. QRNGs, in contrast, use quantum entropy sources, where any attempt to probe or tamper with the system inherently disturbs the quantum state, thus making such attacks detectable. 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 in [43], QRNGs demonstrate substantial advantages in randomness, uniformity, and resistance to correlation-based attacks through comprehensive statistical evaluations, including 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 QRE-ChaCha exhibits improved resistance to differential attacks.

To evaluate the resistance of the QRE-ChaCha against differential cryptanalysis, we adopt the theoretical framework proposed by Kai 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[44]. 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 Stefan Kölbl[45]. In addition, we used the quantum random number API provided by ETH ZÜRICH[46] to obtain the quantum random numbers required for the experiments.

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 2 and 3 rounds of iteration. The upper bounds of the differential probabilities are illustrated in Figure 4. 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 $Pr > 2^{-512}$) does not exceed $3\times20+2\times3 = 66$ rounds. The upper bound on the 20-round differential trail probability is approximately 2^{-154}. Under the same testing conditions and methodology, the 2-round differential trail probability upper bound for the original

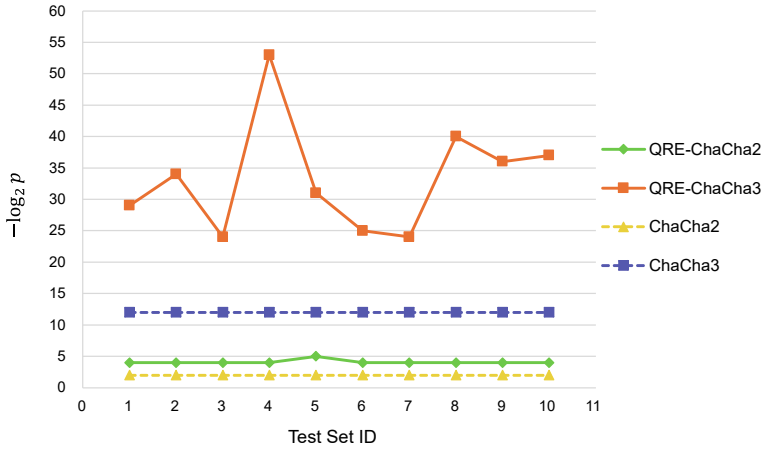

FIGURE 4. Differential probabilities of 2-round and 3-round QRE-ChaCha (10 Sets)

ChaCha is 2^{-2}, and for 3 rounds, it is 2^{-12}. Accordingly, the number of effective differential trails ($Pr > 2^{-512}$) for ChaCha does not exceed $3\times42 + 2\times4 = 134$ 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.

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

6. STATISTICAL RANDOMNESS TESTING

To further validate the quality of the QRE-ChaCha's output, we comprehensively evaluate the randomness characteristics of the QRE-ChaCha using the NIST Statistical Test Suite and the Chinese Cryptographic Randomness Test Standard in this section. Specifically, the tests are performed using the NIST SP 800-22 Rev. 1 [47], and the open-source randomness toolkit project[48].

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[49], 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 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 the QRE-ChaCha introduces no flaws in the randomness of its keystream. The overall testing procedure adopted in this work is summarized as follows:

- This test is based on the 8-round version of the 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 1,000,000 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 NIST and the Chinese standard are adopted. The significance level for both tests is set to 0.01, and the uniformity significance level is set to 0.0001. The results are analyzed to determine whether the keystreams generated by QRE-ChaCha exhibit statistically strong randomness. The test results from both suites are summarized in Table 3 and Table 4.

TABLE 3. NIST randomness test results for 1000 sets of QRE-ChaCha8 keystreams

NIST Test Item	Pass Count	P-Value
Frequency	982	0.187581
Block Frequency	988	0.751866
Cumulative Sums	983	0.435430
Runs	994	0.062821
Longest Run of Ones	984	0.747898
Rank	990	0.784927
FFT	986	0.803720
Non-overlapping Template	982	0.940080
Overlapping Template	990	0.117432
Universal Statistical	990	0.012829
Approximate Entropy	990	0.345650
Serial	992	0.899171
Linear Complexity	987	0.115387

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

TABLE 4. Randomness test results of 10,000 sets of QRE-ChaCha8 keystreams using GM/T 0005-2021

GM/T 0005-2021 Test Item	Pass Count	P-Value
Single Bit Frequency	9884	0.862398
Block Frequency ($m = 10000$)	9902	0.969009
Poker Test ($m = 4$)	9889	0.469806
Poker Test ($m = 8$)	9910	0.362434
Overlapping Template ($m = 3$, P1)	9890	0.978538
Overlapping Template ($m = 3$, P2)	9901	0.211848
Overlapping Template ($m = 5$, P1)	9918	0.610070
Overlapping Template ($m = 5$, P2)	9915	0.906880
Total Runs	9915	0.113239
Run Distribution	9900	0.399442
Max Run of 1s ($m = 10000$)	9900	0.386748
Max Run of 0s ($m = 10000$)	9902	0.650860
Binary Derivation ($k = 3$)	9905	0.699313
Binary Derivation ($k = 7$)	9889	0.669151
Autocorrelation ($d = 1$)	9915	0.073281
Autocorrelation ($d = 2$)	9898	0.187378
Autocorrelation ($d = 8$)	9902	0.128354
Autocorrelation ($d = 16$)	9893	0.846168
Matrix Rank	9902	0.008056
Cumulative Sums (Forward)	9885	0.394370
Cumulative Sums (Backward)	9889	0.447116
Approximate Entropy ($m = 2$)	9890	0.981469
Approximate Entropy ($m = 5$)	9915	0.216485
Linear Complexity ($m = 500$)	9882	0.526907
Linear Complexity ($m = 1000$)	9887	0.155238
Maurer Universal ($L = 7$, $Q = 1280$)	9892	0.621922
Discrete Fourier Transform ($m = 500$)	9892	0.294959

According to the test results, the keystreams generated by QRE-ChaCha8 successfully passed both the NIST and the GM/T 0005-2021 randomness test suites. This indicates that the keystreams exhibit strong randomness properties and meet the requirements for cryptographic applications. Furthermore, the QRE-ChaCha explicitly considers randomness enhancement during its design. By optimizing the original ChaCha structure and incorporating quantum random numbers, additional confusion is introduced, thereby significantly improving the statistical quality of the generated keystreams. The output sequences exhibit excellent distribution uniformity, with no observable statistical patterns exploitable by adversaries. This effectively strengthens resistance against statistical and cryptanalytic attacks, and further demonstrates that the integration of quantum random numbers contributes to enhancing the overall security of the cipher.

7. Performance Evaluation

In this section, to quantify the computational efficiency of the 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 5 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.

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

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[50, 51], which can be pre-stored in the memory module, rendering their impact on overall encryption time negligible. Therefore, the performance tests conducted 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

As a conclusion, we propose an innovative stream cipher, QRE-ChaCha, which enhances the security of the classical ChaCha cipher by introducing quantum random numbers. The core enhancement is achieved through a two-stage process. Firstly, quantum random numbers are XORed into the initial constant block of the state matrix to bolster the randomness of the seed key. Secondly, during each odd-numbered round, the first 128 bits of the intermediate state are 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 examines 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 a 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 2 and 3 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, this work evaluated QRE-ChaCha using both the NIST Statistical Test Suite and the GM/T 0005-2021 standard for randomness testing. The results confirm that the keystreams generated by QRE-ChaCha exhibit strong statistical randomness and meet the requirements for cryptographic applications.

For performance evaluation, this work measures the encryption speed of QRE-ChaCha on input files of varying sizes to assess its overall encryption and decryption efficiency. Experimental results indicate 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 10MB to 50MB in size.

In summary, the proposed QRE-ChaCha cipher preserves the high performance of the original ChaCha cipher 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 scheme, offering new prospects for broader applications of quantum-generated randomness in cryptographic and computational contexts.

References

[1] Adam Langley, Wan-Teh Chang, Nikos Mavrogiannopoulos, Joachim Strombergson, and Simon Josefsson, *ChaCha20-Poly1305 cipher suites for transport layer security (TLS)*, RFC 7905, June 2016.

[2] Daniel J Bernstein and Tanja Lange, *Post-quantum cryptography*, Nature **549** (2017), no. 7671, 188–194.

[3] Daniel J Bernstein, Nadia Heninger, Paul Lou, and Luke Valenta, *Post-quantum RSA*, Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings 8, Springer, 2017, pp. 311–329.

[4] David Jao and Luca De Feo, *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*, Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4, Springer, 2011, pp. 19–34.

[5] Daniele Micciancio, *Lattice-based cryptography*, Encyclopedia of Cryptography and Security, Springer, 2011, pp. 713–715.

[6] ZhengFeng Ji, YouMing Qiao, Fang Song, and Aaram Yun, *General linear group action on tensors: A candidate for post-quantum cryptography*, Theory of cryptography conference, Springer, 2019, pp. 251–281.

[7] Charles H Bennett and Gilles Brassard, *Quantum cryptography: Public key distribution and coin tossing*, Theoretical computer science **560** (2014), 7–11.

[8] Stefano Pirandola, Ulrik L Andersen, Leonardo Banchi, Mario Berta, Darius Bunandar, Roger Colbeck, Dirk Englund, Tobias Gehring, Cosmo Lupo, Carlo Ottaviani, et al., *Advances in quantum cryptography*, Advances in optics and photonics **12** (2020), no. 4, 1012–1236.

[9] Christopher Portmann and Renato Renner, *Security in quantum cryptography*, Reviews of Modern Physics **94** (2022), no. 2, 025008.

[10] ChengLong Li, KaiYi Zhang, XingJian Zhang, KuiXing Yang, Yu Han, SuYi Cheng, HongRui Cui, WenZhao Liu, MingHan Li, Yang Liu, et al., *Device-independent quantum randomness–enhanced zero-knowledge proof*, Proceedings of the National Academy of Sciences **120** (2023), no. 45, e2205463120.

[11] Renato Renner, *Security of quantum key distribution*, International Journal of Quantum Information **6** (2008), no. 01, 1–127.

[12] Hoi-Kwong Lo, XiongFeng Ma, and Kai Chen, *Decoy state quantum key distribution*, Physical review letters **94** (2005), no. 23, 230504.

[13] Antonio Acín, Nicolas Brunner, Nicolas Gisin, Serge Massar, Stefano Pironio, and Valerio Scarani, *Device-independent security of quantum cryptography against collective attacks*, Physical Review Letters **98** (2007), no. 23, 230501.

[14] Hoi-Kwong Lo, Marcos Curty, and Bing Qi, *Measurement-device-independent quantum key distribution*, Physical review letters **108** (2012), no. 13, 130503.

[15] Gordon Procter, *A security analysis of the composition of ChaCha20 and Poly1305*, Cryptology ePrint Archive, Paper 2014/613, 2014, `https://eprint.iacr.org/2014/613c`.

[16] Daniel J Bernstein et al., *ChaCha, a variant of Salsa20*, Workshop record of SASC, Citeseer, 2008, `https://cr.yp.to/chacha/chacha-20080120.pdf`, pp. 3–5.

[17] Sabyasachi Dey, Tapabrata Roy, and Santanu Sarkar, *Revisiting design principles of Salsa and ChaCha*, Advances in Mathematics of Communications **13** (2019), no. 4, 689–704.

[18] Vaisakh Mannalatha, Sandeep Mishra, and Anirban Pathak, *A comprehensive review of quantum random number generators: Concepts, classification and the origin of randomness*, Quantum Information Processing **22** (2023), no. 12, 439.

[19] Miguel Herrero-Collantes and Juan Carlos Garcia-Escartin, *Quantum random number generators*, Reviews of Modern Physics **89** (2017), no. 1, 015004.

[20] XiongFeng Ma, Xiao Yuan, Zhu Cao, Bing Qi, and Zhen Zhang, *Quantum random number generation*, npj Quantum Information **2** (2016), no. 1, 1–9.

[21] Maksim Iavich, Tamari Kuchukhidze, Tetyana Okhrimenko, and Serhii Dorozhynskyi, *Novel quantum random number generator for cryptographical applications*, 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), IEEE, 2020, pp. 727–732.

[22] Maksim Iavich, Tamari Kuchukhidze, Giorgi Iashvili, and Sergiy Gnatyuk, *Hybrid quantum random number generator for cryptographic algorithms*, Radioelectronic and Computer Systems (2021), no. 4, 103–118.

[23] Mario Stipcevic, *Quantum random number generators and their applications in cryptography*, Advanced photon counting techniques VI, vol. 8375, SPIE, 2012, pp. 20–34.

[24] Randy Kuang, Dafu Lou, Alex He, Chris McKenzie, and Michael Redding, *Pseudo quantum random number generator with quantum permutation pad*, 2021 IEEE international conference on quantum computing and engineering (QCE), IEEE, 2021, pp. 359–364.

[25] LeiLei Huang, HongYi Zhou, Kai Feng, and ChongJin Xie, *Quantum random number cloud platform*, npj Quantum Information **7** (2021), no. 1, 107.

[26] Zakaria Najm, Dirmanto Jap, Bernhard Jungk, Stjepan Picek, and Shivam Bhasin, *On comparing side-channel properties of AES and ChaCha20 on microcontrollers*, 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), IEEE, 2018, pp. 552–555.

[27] SV Dilip Kumar, Sikhar Patranabis, Jakub Breier, Debdeep Mukhopadhyay, Shivam Bhasin, Anupam Chattopadhyay, and Anubhab Baksi, *A practical fault attack on arx-like ciphers with a case study on chacha20*, 2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), IEEE, 2017, pp. 33–40.

[28] Jean Paul Degabriele, Jérôme Govinden, Felix Günther, and Kenneth G Paterson, *The security of ChaCha20-Poly1305 in the multi-user setting*, Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 1981–2003.

[29] Stefano Barbero, Emanuele Bellini, and Rusydi Makarim, *Rotational analysis of ChaCha permutation*, arXiv preprint arXiv:2008.13406 (2020).
[30] Leonardo Sergio Centellas Claros, Leticia Blanco Coca, and Juan Pablo Sandoval Alcocer, *Comparative study of the symmetric cryptography algorithms AES, 3DES and ChaCha20*, Acta Nova **10** (2022), no. 3, 283–302, http://www.scielo.org.bo/pdf/ran/v10n3/1683-0789-ran-10-03-283.pdf.
[31] Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier, and Christian Rechberger, *New features of Latin dances: analysis of Salsa, ChaCha, and Rumba*, Fast Software Encryption: 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers 15, Springer, 2008, pp. 470–488.
[32] ZhenQing Shi, Bin Zhang, DengGuo Feng, and WenLing Wu, *Improved key recovery attacks on reduced-round Salsa20 and ChaCha*, International Conference on Information Security and Cryptology, Springer, 2012, pp. 337–351.
[33] Arka Rai Choudhuri and Subhamoy Maitra, *Differential cryptanalysis of Salsa and ChaCha – an evaluation with a hybrid model*, Cryptology ePrint Archive, Paper 2016/377, 2016, https://eprint.iacr.org/2016/377.
[34] Kakumani KC Deepthi and Kunwar Singh, *Cryptanalysis of Salsa and ChaCha: revisited*, International Conference on Mobile Networks and Management, Springer, 2017, pp. 324–338.
[35] Shotaro Miyashita, Ryoma Ito, and Atsuko Miyaji, *PNB-focused differential cryptanalysis of ChaCha stream cipher*, Australasian Conference on Information Security and Privacy, Springer, 2022, pp. 46–66.
[36] Emanuele Bellini, David Gerault, Juan Grados, Rusydi H Makarim, and Thomas Peyrin, *Boosting differential-linear cryptanalysis of ChaCha7 with MILP*, IACR Transactions on Symmetric Cryptology **2023** (2023), no. 2, 189–223.
[37] Nasratullah Ghafoori and Atsuko Miyaji, *Higher-order differential-linear cryptanalysis of ChaCha stream cipher*, IEEE Access **12** (2024), 13386–13399.
[38] Mohammed Salih Mahdi, Nidaa Falih Hassan, and Ghassan H Abdul-Majeed, *An improved chacha algorithm for securing data on IoT devices*, SN Applied Sciences **3** (2021), no. 4, 429.
[39] Deepak Kumar Jain, Prakash Mohan, Kuruva Lakshmanna, and Ashok Kumar Nanda, *Enhanced data privacy in cyber-physical system using improved Chacha20 algorithm*, PREPRINT (Version 1) available at Research Square (2022).
[40] Victor R Kebande, *Extended-Chacha20 stream cipher with enhanced quarter round function*, IEEE Access **11** (2023), 114220–114237.
[41] Abeer Tariq Maolood, Ekhlas Khalaf Gbashi, and Eman Shakir Mahmood, *Novel lightweight video encryption method based on ChaCha20 stream cipher and hybrid chaotic map*, International Journal of Electrical & Computer Engineering (2088-8708) **12** (2022), no. 5, 4988–5000.
[42] Elizabeth Henry, *The role of quantum random number generation in enhancing encryption security*, Available at SSRN 4966139 (2024).
[43] Shashi Kant Pandey and R Jenef, *A comparative study and analysis of quantum random number generator with true random number generator*, 2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS), IEEE, 2024, pp. 1000–1005.
[44] Kai Fu, MeiQin Wang, YingHua Guo, SiWei Sun, and Lei Hu, *MILP-based automatic search algorithms for differential and linear trails for speck*, Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers 23, Springer, 2016, pp. 268–288.
[45] Stefan Kölbl, *CryptoSMT: An easy to use tool for cryptanalysis of symmetric primitives*, https://github.com/kste/cryptosmt, Accessed: 2024-05-27.
[46] ETH ZÜRICH, *Quantum random number generator*, http://qrng.ethz.ch/live/, Accessed: 2024-05-23.
[47] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, et al., *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, Tech. Report NIST Special Publication (SP) 800-22, Rev. 1, National Institute of Standards and Technology, 2010.
[48] Trisia, *Randomness Version 1.5.0*, https://github.com/Trisia/randomness, Accessed: 2024-05-23.

[49] Cryptography Industry Standardization Technical Committee, *Randomness testing specification: GM/T 0005-2021*, China Standards Press, 2021, https://std.samr.gov.cn/hb/search/stdHBDetailed?id=E66CC4F6F8D78B7FE05397BE0A0A6C55, Accessed: 2024-05-23.

[50] Tommaso Bertapelle, Marco Avesani, Alberto Santamato, Alberto Montanaro, Marco Chiesa, Davide Rotta, Massimo Artiglia, Vito Sorianello, Francesco Testa, Gabriele De Angelis, et al., *High-speed source-device-independent quantum random number generator on a chip*, Optica Quantum **3** (2025), no. 1, 111–118.

[51] YiMing Bian, Jie Yang, HaoYuan Jiang, Wei Huang, Qi Su, Song Yu, Lei Zhang, YiChen Zhang, and BingJie Xu, *20 Gbps real-time source-independent quantum random number generator based on a silicon photonic chip*, Optics Letters **50** (2025), no. 4, 1216–1219.

[1]School of Cyberspace, Hangzhou Dianzi University, Hangzhou, 310018, China

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

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

Email address: zhaoshuai@hdu.edu.cn