密码学报 ISSN 2095-7025 CN 10-1195/TN
Journal of Cryptologic Research

E-mail: jcr@cacrnet.org.cn
http://www.jcr.cacrnet.org.cn
Tel/Fax: +86-10-82789618

量子随机数增强的 ChaCha 密码算法*

刘 超, 赵 帅, 贾晨浩, 胡耿然, 崔婷婷

杭州电子科技大学 网络空间安全学院, 杭州 310018

通信作者: 赵帅, E-mail: zhaoshuai@hdu.edu.cn 崔婷婷, E-mail: cuitingting@hdu.edu.cn

摘 要: 本文提出了一种量子随机数增强的 ChaCha (QRE-ChaCha) 流密码算法. ChaCha 算法作为一种重要的流密码算法, 具有高效性和安全性, 在实时通信和数据流传输场景中得到了广泛应用. 随着对 ChaCha 算法的深入研究, 研究者们通过各种分析方法将 ChaCha 算法受攻击的轮数提升至 7 轮甚至更高, 同时随着量子信息技术的发展, 针对 ChaCha 算法的攻击可能更加容易. 为此, 在保持 ChaCha 算法高效性和适应性的基础上, 旨在进一步增强其安全性, 本文提出的 QRE-ChaCha 算法在 ChaCha 原轮函数的基础上使用量子随机数增强的状态矩阵作为轮函数的输入并对轮变换进行了改进, 从而提高其随机性以及抵抗经典攻击和量子攻击的能力. 为评估所提出算法的安全性, 文中基于可满足性问题 (SAT) 的自动化搜索方法对 QRE-ChaCha 算法进行了差分和线性密码分析测试, 并使用 NIST 随机性统计测试套件和 GM/T 0005-2021 随机性规范对算法生成的密钥流进行了随机性检测, 还通过测量 QRE-ChaCha 算法的加密速度与解密速度来评估其加解密性能. 测试结果表明, QRE-ChaCha 算法在保持 ChaCha 算法高效性的情况下, 其抗差分攻击与抗线性攻击的能力有显著的提升, 并且通过 NIST 与 GM/T 0005-2021 的随机性检测. 与此同时, 得益于量子随机数作为种子密钥, QRE-ChaCha 算法还具有一定抵抗量子攻击的能力.

关键词: 流密码; ChaCha; 量子随机数; QRE-ChaCha

中图分类号: TP309.7 **文献标识码:** A DOI:

中文引用格式: 刘超, 赵帅, 贾晨浩, 胡耿然, 崔婷婷. 量子随机数增强的 ChaCha 密码算法[J].

英文引用格式: LIU C, ZHAO S, JIA C H, HU G R, CUI T T. Quantum random number enhanced chacha cipher algorithm[J].

Quantum Random Number Enhanced ChaCha Cipher Algorithm

LIU Chao, ZHAO Shuai, JIA Chen-Hao, HU Geng-Ran, CUI Ting-Ting

School of Cyberspace Security, Hangzhou Dianzi University, Hangzhou 310018, China

Abstract: In this paper, a quantum random number-enhanced ChaCha (QRE-ChaCha) stream cipher algorithm is proposed. As an important stream cipher algorithm with high efficiency and security, ChaCha algorithm is widely used in real-time communication and data stream transmission scenarios. With the in-depth study of ChaCha algorithm, researchers have increased the number of attacked rounds of ChaCha algorithm to 7 or even higher by various analytical methods, and with the development of quantum information technic, the attack against ChaCha algorithm may be easier

* 基金项目: 浙江省自然科学基金 (No.LQ24A050005); 杭州电子科技大学科研启动基金 (No.KYSZ275623071)
Foundation: Zhejiang Provincial Natural Science Foundation of China (No.LQ24A050005); The Research Startup Foundation of Hangzhou Dianzi University (No.KYSZ275623071)

in the future. In order to further enhance the security of the ChaCha algorithm while maintaining its efficiency and adaptability, the QRE-ChaCha algorithm proposed in this paper uses a quantum random number-enhanced state matrix as the input of the wheel function, and improves the wheel transformations on the basis of the original wheel function of ChaCha, which improve its randomness and resistance to both classical and quantum attacks. In order to evaluate the security of the proposed algorithm, the differential and linear cryptanalysis of the QRE-ChaCha algorithm is tested based on the automated search method of the satisfiability problem (SAT), and the key stream generated by the algorithm is checked for randomness using the NIST Statistical Test Suite for Randomness and the randomness specification of GM/T 0005-2021, and the encryption speed of the QRE-ChaCha algorithm is measured by the measurement of the QRE-ChaCha algorithm. ChaCha algorithm is also evaluated by measuring its encryption and decryption speeds. The test results show that the QRE-ChaCha algorithm, by maintaining the high efficiency of the ChaCha algorithm, has a remarkable degree of improvement in its resistance to differential attacks and linear attacks, and passes the randomness tests of NIST and GM/T 0005-2021. Meanwhile, due to the quantum random number as the seed key, the QRE-ChaCha algorithm is also resistant to quantum attacks.

Key words: Stream cipher; ChaCha; Quantum random number; QRE-ChaCha

1 引言

随着信息技术的迅猛发展, 数据安全的需求日益增长. 在现代通信和存储系统中, 加密算法被广泛应用于保护敏感数据的安全性和机密性. 与此同时, 随着量子技术的不断突破, 计算机被赋予极强的处理能力, 传统加密算法的安全性正面临着新的挑战. 在这样的背景下, 基于不存在有效量子算法的困难数学问题提出的后量子密码 [1–5] 和基于量子物理提出的量子密码技术 [6–13] 成为了当前信息领域的前沿研究热点。量子随机数是量子密码研究的重要分支, 也是当前成熟应用的量子密码技术之一. 然而, 相比于经典随机数发生器, 量子随机数发生器还存在随机数生成码率不高的限制. 对此, 研究者们提出了一种新的思路 [9]: 将量子随机数与经典密码算法相结合, 一方面可以扩展量子随机数发生器的应用场景, 另一方面可以提高经典密码算法的抵抗量子攻击的能力.

流密码算法是一种基于比特异或和伪随机数生成器的加密技术, 它以流的方式生成密钥流, 并将密钥流与明文进行异或运算来实现加密. 与分组密码算法不同, 流密码算法可以实现逐比特的加密和解密操作, 具有较高的加解密速度和可并行性, 在种子密钥及随机数具有高随机性的情况下, 可以保证较高的安全性 [14]. 因此, 针对流密码算法提升其种子密钥及随机数的随机性是提升算法安全性的一种可操作的思路.

ChaCha 算法作为一种重要的流密码算法, 在近年来备受关注. 它由丹尼尔 · J · 伯恩斯坦 (Daniel J. Bernstein) 于 2008 年设计, 是 Salsa20 算法的改进版本 [16]. ChaCha 算法虽然基于 Salsa20 算法的结构, 但通过引入不同的种子密钥矩阵和更复杂的轮变换, 在不减慢加密速度的情况下实现更强的扩散性能, 以提供更高的安全性和性能. 它以其高效性和安全性在实时通信和数据流传输场景中得到了广泛应用, 正如 ChaCha20-Poly1305 流密码套件在传输层安全 (TLS) 和数据报传输层安全 (DTLS) 协议中的使用 [15].

尽管 ChaCha 算法已经得到广泛应用和研究, 但随着对 ChaCha 算法的不断深入研究, 研究者们针对算法进行了一些优化. 这些优化的目标包括提高算法的抗攻击性、降低算法的计算和存储需求以及发掘新的应用场景等 [17–19]. 本文通过引入量子随机数并对轮变换进行改进, 提出一种保持原有算法高性能特点的量子随机数增强的 ChaCha 密码算法方案. 量子随机数增强的 ChaCha 密码算法在保持 ChaCha 算法高效性和适应性的基础上, 在种子密钥和轮变换中引入量子随机数, 进一步增强 ChaCha 算法的抵抗差分攻击和量子攻击的能力. 本文提出的主要贡献可概括如下:

(1) 提出了一种量子随机数增强的 ChaCha 流密码算法 (QRE-ChaCha, Quantum Random Number Enhanced), 该算法通过间隔在轮函数中引入量子随机数, 优化了传统 ChaCha 算法的安全性和抗攻击能力. 具体来说, 对于初始状态矩阵, 量子随机数通过异或组件作用于初始状态矩阵中的初始

常量; 对于奇数轮输出的中间状态矩阵, 量子随机数通过异或组件对矩阵中的部分位置进行增强. QRE-ChaCha 密码算法中引入的量子随机数可以对已知的密码攻击分析方法提供更强的抵抗能力. 量子随机数本身由量子物理过程生成, 其不可预测性和高熵特性使得它们非常适合作为密码系统的种子密钥, 同时将量子随机数通过异或操作作用于算法轮变换的过程, 可以使量子随机数的真随机性快速扩散至整个状态矩阵, 显著地增加输出密钥流的随机性和不可预测性, 使得攻击者难以找到有效的差分路径, 并增强密码系统对量子攻击的抵抗能力.

(2) 基于可满足性问题 (Boolean satisfiability problem, SAT) 的自动化搜索方法对所提出的 QRE-ChaCha 密码算法进行了差分分析测试, 以评估其安全性. 根据测试结果, QRE-ChaCha 算法的 2 轮差分概率与 3 轮差分概率均更优于 ChaCha 算法. 除此之外, 本文使用 NIST 统计测试套件和基于《GM/T 0005—2021 随机性检测规范》的测试方法对算法生成的密钥流进行随机性检测. 根据测试结果, 本文提出的 QRE-ChaCha 算法所生成的密钥流均通过了上述两项测试, 并且部分测试结果优于 ChaCha 算法, 可以认为 QRE-ChaCha 算法能够提供更加均匀随机的密钥流.

(3) 还通过测量 QRE-ChaCha 算法的加密速度与解密速度来评估算法性能, 并将其与 ChaCha 算法进行比较. 该测试目的是比较 QRE-ChaCha 在提升部分安全性的条件下与 ChaCha 算法的性能是否有较大的差距. 根据测试结果, 在没有考虑量子随机数生成过程中的时间消耗的情况下, QRE-ChaCha 算法的加解密速度几乎与 ChaCha 算法一致.

(4) QRE-ChaCha 算法能够实现量子随机数的扩张, 弥补现阶段量子随机数发生器成码率低的不足, 进一步扩展量子随机数的应用场景.

本文剩余内容分为以下几个部分: 第2节简要概述了 ChaCha 密码算法研究的相关工作和进展; 第3节详细描述本文提出的优化方案; 第4节为本文所提出算法的安全性测试分析与结果; 第5节为本文所提出算法的随机性测试; 第6节为本文所提出算法的性能评估测试分析与结果; 第7节总结全文并对未来的研究方向进行展望.

2 相关工作

2.1 量子随机数在密码中的应用

在量子随机数领域, 量子随机数的生成一直是研究的热点. Mannalatha 等人 [20] 关于量子随机数生成器 (QRNGs) 进行了综合性回顾, 着重讨论了它们在经典世界中无法实现的各种可能特征 (如设备独立性、半设备独立性), 并从量子力学的一组分层公理的角度探讨了随机性的起源, 对可用的 QRNGs 进行了分类, 并对每个类别的技术挑战进行了分析. 同样, Ma 等人 [21] 对量子随机数生成 (QRNG) 的不同类型及其基本原理提供了概览, 阐述了实用 QRNG、自检测 QRNG 和半自检测 QRNG 三种主要类型, 以及它们各自的优缺点. 还介绍了一些常见的 QRNG 实现方案, 如基于单光子探测器、宏观光电探测器、真空噪声和放大自发辐射. Herrero-Collantes 等人 [22] 进一步深入探讨了 QRNG 的发展历程、不同实现技术以及随机性评估方法. 从基于放射性衰变的早期 QRNG 设备出发, 介绍了利用光的量子态收集熵的多种新型方法. 重点分析了基于噪声、光学和非光学技术的 QRNG, 说明了量子力学如何通过量子现象提供真正的随机性.

对于量子随机数的应用领域, Iavich 等人于 2020 年提出了一种新型基于光子到达时间的 QRNG [23], 又于 2021 年提出一种结合了时间到达 QRNG、光子计数 QRNG 和衰减脉冲 QRNG 等技术混合的 QRNG [24]. 前者目标是以较低成本生成快速而高质量的量子随机数, 以满足密码学应用的需求, 后者专门用于加密算法领域, 生成器结合了量子随机性和经典伪随机数生成器, 以增强随机数的质量和生成码率, 实验证明, 能够以较高的速率生成具有高质量的随机数, 将其用于传统密码领域效果十分出色. 同样, Stipčevi 等人 [25] 将视角聚焦于密码学领域, 阐述了随机数生成器在密码学中的重要性. 通过比较基于自由运行振荡器和 QRNG 的两种随机数生成方法, 着眼于量子密码学应用场景, 探讨了随机数的作用、后处理技术以及评估方法, 为 QRNG 在密码学中的具体应用提供了指引. 与物理 QRNG 不同, Kuang 等

人 [27] 提出了一种基于量子算法的伪量子随机数生成器 (pQRNG). pQRNG 利用了量子排列空间的高熵特性, 通过量子排列垫 (QPP) 技术生成具有良好不可预测性的伪随机, pQRNG 无需进行物理集成, 即可与任何经典计算系统集成, 为密码学和其他领域提供高质量的确定性随机数源. 将量子随机数的应用落地, Huang 等人 [26] 提出了一个实际可用的量子随机数云平台, 将 QRNG 与阿里巴巴的云服务器相结合, 为密码学等应用提供随机数支持.

在量子随机数认证领域, Iavich 等人 [28] 考虑到 QRNG 认证的重要性, 提出了一种新的 QRNG 认证方法, 结合了自检测和设备无关的量子随机数生成方法, 提供了安全高效的认证途径, 提高 QRNG 系统的效率. Drahi 等人 [29] 提出了一种从不可信光源中生成经认证的量子随机数的方法, 通过对光源特征化和模拟, 确保生成的随机数具备高度随机性和安全性, 通过实验表明, 能以很高速率产生具有组合安全性的量子随机数, 同时克服了对光源的信任困难, 为密码学等领域提供可靠的随机数源.

值得注意的是 2023 年 Jian-Wei Pan 等人 [9] 提出了基于设备无关量子随机性的增强型零知识证明, 进一步验证了将量子随机数与经典密码算法结合的可行性. 文中提出了一种量子解决方案, 即实现一个量子随机服务. 该服务通过无漏洞的贝尔测试生成随机数, 并通过后量子密码认证进行传递, 以此排除了哈希函数的理想假设, 增强了协议的安全性. 与此类似的思想同样可以应用于密码算法的优化,QRE-ChaCha 算法通过将量子随机数引入轮变换中, 以增强算法的安全性.

2.2 针对 ChaCha 算法的分析与优化

2008 年, 丹尼尔 · J · 伯恩斯坦 (Daniel J. Bernstein) 设计出了 ChaCha20 算法作为 Salsa20 算法的改进版本 [16], 该算法作为流密码家族中的一员, 保持了高性能和简单算法的优点, 因此有着广泛的应用. 但是到目前为止, 大部分关于 ChaCha 的研究工作都集中在对算法结构的分析和攻击 [14,15,30–32,34–43], 而对于 ChaCha 算法的优化研究 [17–19,43] 相对较少.

对于 ChaCha 算法的分析工作, Langley 等人 [15] 介绍了在传输层安全性 (TLS) 和数据报传输层安全性 (DTLS) 协议中使用 ChaCha20 流密码和 Poly1305 认证器的方法, 这种组合不仅解决了 RC4 的安全性问题, 还提供了高效的加解密性能. Najm 等人 [30] 比较了在微控制器上使用的 AES 和 ChaCha20 两种密码算法在抵抗侧信道攻击方面的差异, 与 AES 相比, ChaCha20 展现出了更优的抵抗侧信道攻击的特性, 但同时开销更大. Kumar 等人 [31] 首次在文献中提出了针对 ChaCha20 流密码的实际故障攻击研究, 提出了四种差分故障分析攻击, 并在实验中验证了这些攻击的有效性, 揭示了 ChaCha20 在抵御故障攻击方面的潜在弱点. Procter 等人 [14] 分析了将 Bernstein 的 ChaCha20 流密码与 Poly1305 认证器组合作为 IETF 协议中的认证加密方案的安全性, 并通过安全约简证明了该组合方案的安全性, 为 ChaCha20-Poly1305 的实际应用奠定了理论基础. Tordsson 等人 [32] 针对 AES-GCM 和 ChaCha20-Poly1305 的分区预言攻击, 并提出了改进方法, 通过解决线性方程组问题展示了这些攻击的有效性, 同时提出了使用非域环的替代方案以增加攻击难度. Degabriele 等人 [33] 针对 ChaCha20-Poly1305 在多用户环境下的安全性进行了深入研究, 提出了改进的安全性分析方法, 建立了更严格的安全边界和攻击模型, 揭示了该加密方案在多用户环境下的安全性界限. Barbero 等人 [34] 对 ChaCha 置换的旋转特性进行了理论分析, 推导了旋转在置换中传播的概率上下界, 并提出了一种通过调用置换进行区分的方法, 但作者也指出目前尚未发现将这些结果应用于 ChaCha 流密码密码分析的方法. Centellas Claros 等人 [35] 比较了三种对称加密算法 AES、3DES 和 ChaCha20 在性能和效率方面的表现, 结果显示 ChaCha20 在加密解密速度上最快.

针对 ChaCha 算法创新式的分析方法起源于 Aumasson 等人 [36] 于 2008 提出的中性位概念, 并将其应用于 Salsa20 和 ChaCha 流密码的新的差分密码分析方法, 并成功地展示了对它们的攻击, 这是密码分析中首次应用这一概念. 随后, Shi 等人 [37] 提出了改进的密钥恢复攻击方法, 针对降低轮数的 Salsa20 和 ChaCha 加密算法, 通过引入新的区分器类型和发现高概率的二阶差分路径, 展示了比之前方法更小的时间和数据复杂度的攻击结果. Choudhuri 等人 [38] 提出了一个混合模型来评估 Salsa 和 ChaCha 流密码的差分密码分析, 通过在初始轮中使用原始的非线性函数和后续轮中使用线性化的函数, 推导出对于 256 位密钥, 只需考虑 12 轮即可提供安全性. Deepthi 等人 [39] 对 ChaCha20 和 Salsa20 两种流密码进行了深入分析和攻击研究, 发现了之前研究中的错误, 并成功对较低轮数版本 (Salsa20/7、ChaCha6 和 ChaCha7) 进行了攻击. Miyashita 等人 [40] 在原有 PNB 方法的基础上提出了一种优化的基于概率中性位差分攻击方法, 针对 ChaCha 流密码进行了分析, 并找到了最佳的差分偏置和概率中性位组合, 进一步

改进了现有的攻击方法. 后来, Bellini 等人 [41] 提出了一种优化的 ChaCha 流密码的差分线性密码分析方法, 通过考虑更大的搜索空间、优化差分和线性部分之间的掩码选择以及使用精心设计的 MILP 工具, 成功构建了新的差分线性区分器, 大幅提升了对 ChaCha 密码算法的攻击效率, 并提出了对 ChaCha 的针对 7 轮和 7.5 轮的区分器以及针对 5 轮的差分线性区分器. 最后, Ghafoori 等人 [42] 研究了 ChaCha 流密码的高阶差分线性密码分析方法, 并针对其不同轮数进行了分析和攻击, 发现了新的差分偏置和线性逼近, 提高了攻击复杂度.

在 ChaCha 算法的优化领域, Mahdi 等人 [19] 提出了一种名为 Super ChaCha 的增强 ChaCha 算法, 该算法通过改变旋转过程和输入的更新顺序来提高安全性. Super ChaCha 在保持低功耗的同时, 提供了更高的安全性, 适用于资源受限的物联网设备. 应用环境相似的, Jain 等人 [18] 提出了一种优化的 Chacha20 算法, 用于增强物联网设备上的数据隐私. Kebande 等人 [17] 提出了另一种优化的 Chacha20 算法, 称为 EChacha20. 该算法通过增强四分之一轮函数 (QR-F) 并引入 32 位输入字和 ARX 操作来提高安全性. 在算法应用层面, Maolood 等人 [43] 提出了一种新型的轻量级视频加密方法, 该方法基于 ChaCha20 流密码和混合混沌映射理论.

随着对 ChaCha 算法的深入分析, 尽管该算法尚未被真正破解, 但其安全性正逐步受到挑战. 现有的攻击方法虽然未能彻底破解 ChaCha, 但已经揭示了其潜在的弱点, 这促使研究者必须不断探索新的安全增强手段. 同时, 虽然针对 ChaCha 算法的优化工作已取得一定进展, 但主要集中在轮变换结构的优化和算法参数的调整上, 这些改进在提升安全性方面的效果有限. 鉴于此, 本文提出创新的量子随机数增强型 ChaCha 算法. 通过量子随机数的引入, 不仅能够在轮变换中引入高熵随机性, 从而增加算法的不可预测性和抵抗分析攻击的能力, 还能够增强算法种子密钥的安全性, 提高密钥空间的复杂性, 具有一定抵抗量子攻击的能力.

3 量子随机数增强的 ChaCha 算法

本文基于量子随机数的 ChaCha 算法优化方案体现两个方面, 一是在初始状态矩阵中, 量子随机数通过异或组件作用于初始状态矩阵中的初始常量, 以增强种子密钥的随机性; 二是在奇数轮输出的中间状态矩阵中, 量子随机数通过异或组件对矩阵中的部分位置进行增强, 使量子随机数的真随机性通过轮变换过程快速扩散到整个状态矩阵. 在这一部分中, 首先简要描述 ChaCha 算法, 其次详细介绍本文提出的量子随机数增强的 ChaCha 算法优化方案. 为了方便后续的阅读和理解, 表1首先给出了符号约定.

3.1 ChaCha 算法描述

ChaCha 是基于模加循环移位异或 (ARX) 运算的伪随机函数, 通过执行四次加法、四次异或运算和四次循环移位来更新其状态矩阵, 与 Salsa 相比, ChaCha 对每个字都更新两次而不是一次, 并且速度更快. ChaCha 遵循与 Salsa 相同的基本设计原则, 以 32 个比特为一个字单元并以 512 比特作为初始状态矩阵, 其中包括了 4 个字的常量 ($c1 = 0x61707865$, $c2 = 0x3320646e$, $c3 = 0x79622d32$, $c4 = 0x6b206574$), 8 个字的种子密钥, 3 个字的随机数以及 1 个字的计数器. ChaCha 作为一种迭代的流密码, 其循环迭代的轮数取决于应用程序的要求, 需要最大安全性可取 20 轮、需要最大速度可取 8 轮或需要在速度和安全性之间平衡可取 12 轮 [43]. 在每一轮中, ChaCha 的轮函数 (R) 由 4 个四分之一轮函数 (QR) 组成, 每个四分之一轮函数需要 4 个字的输入 $(x_a^{(r)}, x_b^{(r)}, x_c^{(r)}, x_d^{(r)})$. 对于 ChaCha 的初始状态矩阵如公式1所示:

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

公式 (1) 中, c 表示常量, k 表示种子密钥, t 表示计数器, v 表示随机数.

由于一轮中的四分之一轮函数输入的 4 个字各不相同, 所以 ChaCha 轮函数支持四个四分之一轮函数同时执行. 对于奇数轮次 (列操作轮次), 四分之一轮函数分别对 4 个列向量进行操作: $(x_0^{(r)}, x_4^{(r)}, x_8^{(r)}, x_{12}^{(r)})$, $(x_1^{(r)}, x_5^{(r)}, x_9^{(r)}, x_{13}^{(r)})$, $(x_2^{(r)}, x_6^{(r)}, x_{10}^{(r)}, x_{14}^{(r)})$, $(x_3^{(r)}, x_7^{(r)}, x_{11}^{(r)}, x_{15}^{(r)})$. 而对于

表 **1** 符号约定
Table 1 Symbol Definition

符号	定义
X	由 16 个字组成的 4×4 ChaCha 矩阵
$X^{(0)}$	ChaCha/QRE-ChaCha 的初始状态矩阵
$X'^{(0)}$	$x_{i,j}$ 位置有一位差分的关联矩阵
$X^{(R)}$	R 轮后的 ChaCha/QRE-ChaCha 矩阵
$X^{(r)}$	r 轮后的 ChaCha/QRE-ChaCha 矩阵, 其中 $R > r$
$x_i^{(R)}$	状态矩阵 $X^{(R)}$ 的第 i 个字
$x_{i,j}^{(R)}$	状态矩阵 $X^{(R)}$ 的第 i 个字的第 j 位
p	差分路线的概率
$x \boxplus y$	字 x 和字 y 模加
$x \boxminus y$	字 x 和字 y 模减
$x \oplus y$	字 x 和字 y 按位异或
$x \lll n$	字 x 循环左移 n 位
Δx	字 x 和字 x' 的异或差分
ChaCha n	第 n 轮的 ChaCha 流密码
QRE-ChaCha n	第 n 轮的 QRE-ChaCha 流密码

偶数轮次 (对角操作轮次), 四分之一轮函数分别对 4 个对角向量进行操作: $(x_0^{(r)}, x_5^{(r)}, x_{10}^{(r)}, x_{15}^{(r)})$, $(x_1^{(r)}, x_6^{(r)}, x_{11}^{(r)}, x_{12}^{(r)})$, $(x_2^{(r)}, x_7^{(r)}, x_8^{(r)}, x_{13}^{(r)})$, $(x_3^{(r)}, x_4^{(r)}, x_9^{(r)}, x_{14}^{(r)})$. 四分之一轮函数对于 4 个输入字通过顺序执行如公式 (2) 所示的运算来更新内部状态矩阵 $X^{(r)}$, 整体流程结构见图1.

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

对于 ChaCha 的 n 轮版本最终所生成的 512 比特伪随机密钥块 Z 可以表示为:

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

3.2 量子随机数增强的 ChaCha 算法优化方案

本文提出一种量子随机数增强的 ChaCha 算法, 在对 ChaCha 算法的基本原理进行了深入研究之后, 设计了一个新的轮变换算法来更新状态矩阵: 首先, 使用量子随机数通过异或组件作用于初始状态矩阵的初始常量; 其次, 在奇数轮输出的中间状态矩阵中, 使用量子随机数通过异或组件作用于起始的 128bit 数据. 如图2所示, 其中量子随机数生成部分使用量子随机数生成器 (QRNG) 输入源, 随机数提取器 (Ext) 从原始的随机数源中提取出真正的随机数并将其存储进量子随机数存储器中, 以供 QRE-ChaCha 算法调用. 由于本文关注 ChaCha 算法的优化, 图2中并未将详细的量子随机数生成与分发过程表示出来, 事实上, 为了确保随机数的传输安全, 系统可以使用基于哈希的后量子密码 (PQC) 签名算法对生成的随机数进行认证, 此方法在随机数分发过程中能够实现一定的抗量子攻击能力. 根据量子随机数增强特性, 本文把这种算法称为 QRE-ChaCha 密码算法, 其优化方案总结如下:

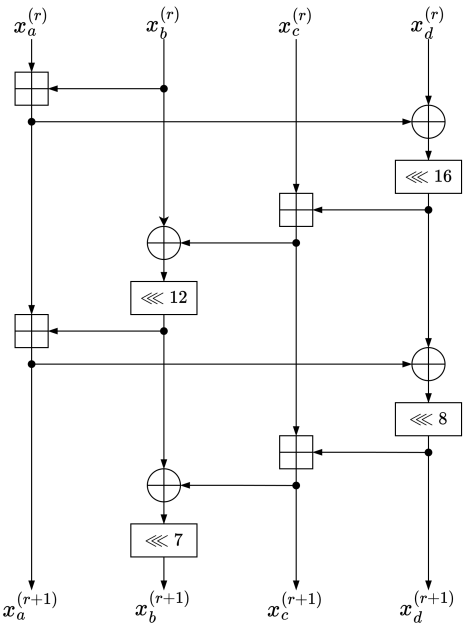

图 1　ChaCha 算法四分之一轮函数结构
Figure 1　ChaCha algorithm quarter-round function structure

- QRE-ChaCha 是一个伪随机函数 (PRF), 其初始输入依旧采用 512 比特矩阵, 但与 ChaCha 算法不同的是, 它将 128 比特常量与生成的 128 比特量子随机数比特进行异或来替换原来位置上的 128 比特常量, 其余采用 256 比特 (8 个字) 密钥、96 比特 (3 个字) 随机数和 32 比特 (1 个字) 计数器作为输入, 如公式 (3) 所示, 其中 q 表示量子随机数.

- QRE-ChaCha 算法使用量子随机数与常量 (c_0, c_1, c_2, c_3) 进行异或操作, 将其结果作为种子密钥矩阵的一部分. 此外, 优化了 ChaCha 算法中的轮函数而没有改变具体四分之一轮函数 (QR) 的内部变换操作, 其通过将奇数轮次的轮函数输入状态矩阵 (即偶数轮次的输出状态矩阵) 中 4 个 32 比特字数据 $(x_0^{(r-1)}, x_1^{(r-1)}, x_2^{(r-1)}, x_3^{(r-1)})$, 与 4 个相同长度的量子随机数比特 $(q_0^{(r-1)}, q_1^{(r-1)}, q_2^{(r-1)}, q_3^{(r-1)})$ 分别进行逐位异或操作, 并将结果替换原有相同位置的 4 个 32 比特字数据, 由此操作来实现算法轮函数的改进, 如公式 (4) 所示, 其中 r 从 0 开始, 故 $X^{(r)}$ 中 r 为偶数时表示奇数轮次的轮函数输入状态矩阵. 综上, 整体 QRE-ChaCha 算法如算法1所描述.

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

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

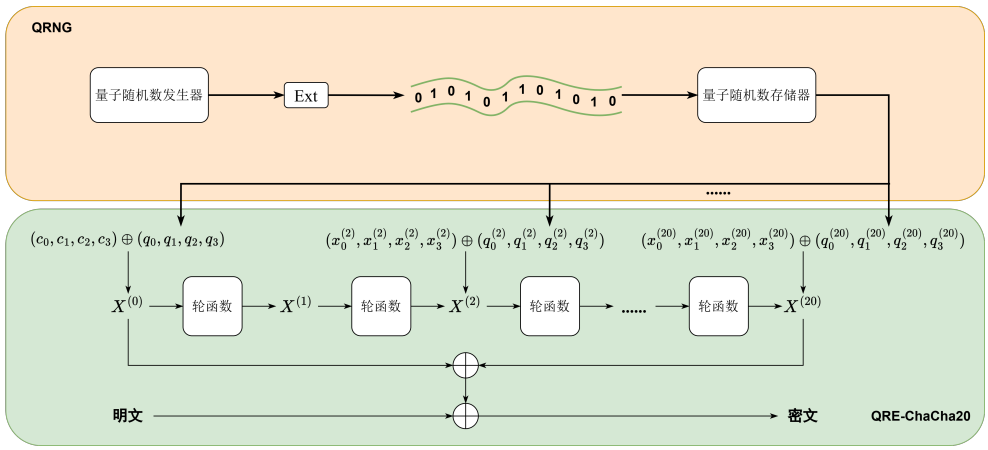

图 2 QRE-ChaCha 算法 20 轮版本整体流程. Ext 表示随机数提取器.

Figure 2 Overall flow of QRE-ChaCha algorithm version 20 rounds. Ext is a random number extractor.

Algorithm 1: QRE-ChaCha 算法

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**

    $(x_0^{(r+1)},x_4^{(r+1)},x_8^{(r+1)},x_{12}^{(r+1)})=QR(x_0^{(r)},x_4^{(r)},x_8^{(r)},x_{12}^{(r)})$;

    $(x_1^{(r+1)},x_5^{(r+1)},x_9^{(r+1)},x_{13}^{(r+1)})=QR(x_1^{(r)},x_5^{(r)},x_9^{(r)},x_{13}^{(r)})$;

    $(x_2^{(r+1)},x_6^{(r+1)},x_{10}^{(r+1)},x_{14}^{(r+1)})=QR(x_2^{(r)},x_6^{(r)},x_{10}^{(r)},x_{14}^{(r)})$;

    $(x_3^{(r+1)},x_7^{(r+1)},x_{11}^{(r+1)},x_{15}^{(r+1)})=QR(x_3^{(r)},x_7^{(r)},x_{11}^{(r)},x_{15}^{(r)})$;

  end

  if *r is even* **then**

    $(x_0^{(r)},x_1^{(r)},x_2^{(r)},x_3^{(r)})=(x_0^{(r)},x_1^{(r)},x_2^{(r)},x_3^{(r)})\oplus(q_0^{(r)},q_1^{(r)},q_2^{(r)},q_3^{(r)})$;

    $(x_0^{(r+1)},x_5^{(r+1)},x_{10}^{(r+1)},x_{15}^{(r+1)})=QR(x_0^{(r)},x_5^{(r)},x_{10}^{(r)},x_{15}^{(r)})$;

    $(x_1^{(r+1)},x_6^{(r+1)},x_{11}^{(r+1)},x_{12}^{(r+1)})=QR(x_1^{(r)},x_6^{(r)},x_{11}^{(r)},x_{12}^{(r)})$;

    $(x_2^{(r+1)},x_7^{(r+1)},x_8^{(r+1)},x_{13}^{(r+1)})=QR(x_2^{(r)},x_7^{(r)},x_8^{(r)},x_{13}^{(r)})$;

    $(x_3^{(r+1)},x_4^{(r+1)},x_9^{(r+1)},x_{14}^{(r+1)})=QR(x_3^{(r)},x_4^{(r)},x_9^{(r)},x_{14}^{(r)})$;

  end

end

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

QRE-ChaCha 的优化方案启发于文献 [9] 提出的一种基于设备无关的量子随机数的非交互式零知识证明 (NIZKP) 协议, 该协议利用了设备无关的量子随机数来增强安全性. 类似地, 本文将量子随机数生成器生成的量子随机数应用在 ChaCha 算法中来增强算法的安全性. 其原理在于, 量子随机数基于量子力学原理, 如量子纠缠和量子非局域性, 本质上是不可预测的, 具有真正的随机性. 而在传统密码学中, 随机数一般使用基于确定性算法的伪随机数生成器, 本质上是确定性的, 不具有真随机性. 其次, 依据不同的安全场景, 量子随机数已经发展出了设备无关、半设备无关和设备依赖的协议. 例如设备无关性意味着它不依赖于设备的物理实现细节, 这有助于抵御侧信道攻击, 攻击者无法通过分析设备的物理特性来预测量子随机数. 进一步地, 使用量子随机数替换 ChaCha 算法中部分状态矩阵, 依靠于量子随机数的真随机性与替换规则, QRE-ChaCha 的安全性与随机性在理论上会有显著的提升.

4 QRE-ChaCha 安全性分析

本文在这一个部分, 针对 QRE-ChaCha 算法进行了差分分析, 并给出了分析过程与实验结果, 尽管针对密码算法的分析方法众多, 但差分分析依然是最主流的密码分析方法, 出于设备条件等因素的限制, 本文针对 QRE-ChaCha 算法的差分分析只进行了 2 轮和 3 轮, 但这已经足够说明 QRE-ChaCha 算法的抗差分性能有所提高.

为了评估 QRE-ChaCha 算法的抗差分分析能力, 本文基于 Kai Fu 等人于 2016 年提出的 ARX 密码中在模加法独立输入和独立轮的假设下用线性不等式描述模加法的差分特征和线性逼近的理论 [44], 使用可满足性问题 (Boolean satisfiability problem, SAT) 的计算机自动化搜索方法, 对短轮数的 QRE-ChaCha 最优差分路线进行了搜索. 在本文所使用的测试方法中, 将约定量子随机数的差分均不等于与其异或的四分之一轮函数输入, 如公式 (5) 所示, 若不进行此约束, 测试结果将无意义.

$$\Delta q_i^{(r)} \neq \Delta x_a^{(r)}, \tag{5}$$

其中 $i = 0, 1, 2, 3$, r 代表 QRE-ChaCha 算法的第 r 轮, $x_a^{(r)}$ 表示四分之一轮函数输入的第一个字参数.

在实验中, 我们采用了苏黎世联邦理工学院提供的量子随机数 API 接口 [45] 来获取量子随机数. 为了评估量子随机数对算法安全性的具体贡献, 本文选取了 10 对独立生成的量子随机数, 并计算了每对随机数之间的差分值. 这些差分值被用作自动化搜索过程中的固定差分约束条件. 基于此, 我们计算了算法在经过 2 轮和 3 轮迭代后的最优差分路径和差分概率, 其算法差分概率上界如图3所示. QRE-ChaCha 算法的 2 轮差分概率上界稳定在 2^{-4}, 3 轮差分概率上界在 2^{-24} 与 2^{-53} 之间波动. 进一步地, 对 10 组差分概率进行了平均处理, 以获得一个代表性的结果, 其 2 轮平均差分概率上界约为 2^{-4}, 3 轮约为 2^{-25}.

为了确保对比的严谨性, 本实验采用了与 QRE-Chacha 算法相同的自动化搜索方法, 对 Chacha 算法本身在 2 轮和 3 轮迭代后的差分概率进行了评估. 这种对比方法有助于我们更准确地量化量子随机数在提高算法安全性方面的实际效益. 然而, 由于测试设备性能的限制, 本实验的自动化搜索仅能提供 QRE-ChaCha 算法在 3 轮迭代后的差分概率的一个上限估计. 尽管未能得到最优的差分路径和差分概率, 但其上限估计已能说明最终结果.

由最终搜索结果并进行平均处理后可知 (如表2所示), QRE-ChaCha 算法的 2 轮平均差分路线概率上界约为 2^{-4}, 3 轮平均差分路线概率上界约为 2^{-25}, 因此 QRE-ChaCha 的有效差分路线 ($Pr > 2^{-512}$) 不超过 $3{\times}20 + 2{\times}3 = 66$ 轮, 同样可知 20 轮的概率上界为 2^{-154}. 在相同测试条件与测试方法下, ChaCha 算法的 2 轮差分路线概率上界为 2^{-2}, 3 轮差分路线概率上界为 2^{-12}, 因此 ChaCha 的有效差分路线 ($Pr > 2^{-512}$) 不超过 $3{\times}42 + 2{\times}4 = 134$ 轮, 20 轮的概率上界为 2^{-74}. 所以, 在本文这样的研究方法下, QRE-ChaCha 算法的抗差分性能要优于 ChaCha 算法.

表 **2** QRE-ChaCha 与 ChaCha 算法 (平均) 差分路线概率
Table 2 QRE-ChaCha and ChaCha algorithms (average) differential route probabilities

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

5 QRE-ChaCha 随机性测试

为了全面评估 QRE-ChaCha 算法的随机性特性, 本文使用 NIST 随机性测试套件与国密随机性测试套件, 分别来源于 NIST 信息技术实验室的 NIST 随机数统计测试套件 [46] 与 randomness 工具库开源项目 [47]. NIST 随机性测试套件是由美国国家标准与技术研究院 (National Institute of Standards and

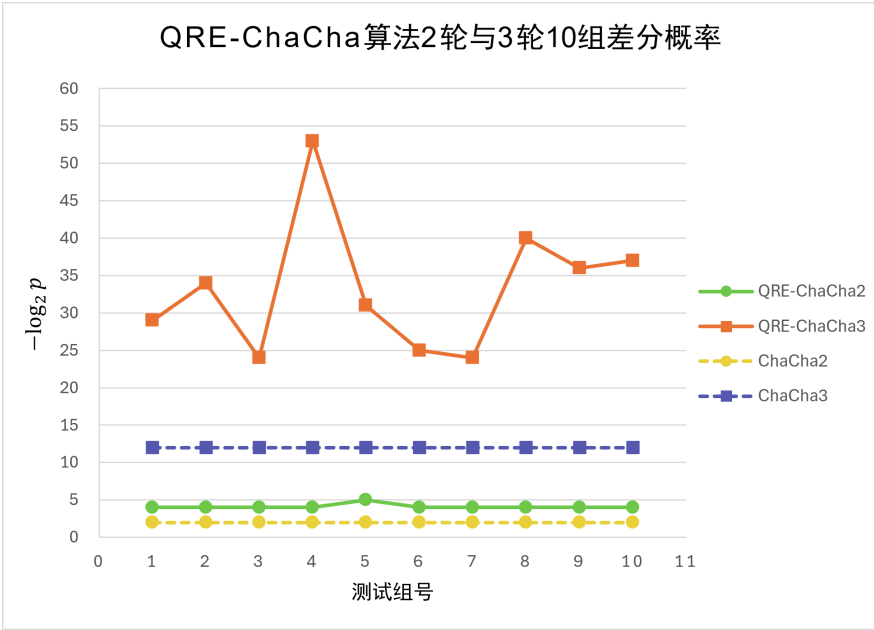

图 3 QRE-ChaCha 算法 2 轮与 3 轮 10 组差分概率
Figure 3 QRE-ChaCha algorithm 2-round vs. 3-round 10-group difference probabilities

Technology, NIST) 发布的, 它包含 15 个测试项, 用于测试由基于硬件或软件的加密随机数或伪随机数生成器生成的二进制序列的随机性.

国密随机性测试则符合《GM/T 0005—2021 随机性检测规范》[48], 它也包含 15 个检测项, 其中 11 个与 NIST 的检测项相同, 包括单比特频数检测、块内频数检测、游程总数检测、块内最大 1 游程检测、矩阵秩检测、离散傅里叶变换检测、Maurer 通用统计检测、线性复杂度检测、重叠子序列检测、近似熵检测、累加和检测.

此外, 国密还有 4 个专有检测项, 包括扑克检测、游程分布检测、二元推导检测、自相关检测. 通过这两个随机性测试能够确保生成的随机数序列具有良好的随机性, 以满足密码学和其他需要随机数的应用的需求, 证明 QRE-ChaCha 算法在密钥流随机性方面没有缺陷. 本文的整体测试流程可以总结如下:

- 本文基于 8 轮的 QRE-ChaCha 算法, 这是我们约定的可以使用的最小轮数版本, 使用随机生成的种子密钥对完全相同的明文进行 QRE-ChaCha8 加密, 生成 10000 组 1000000 比特长度的密钥流序列.
- 针对这 10000 组密钥流序列分别进行 NIST 随机性测试和国密随机性测试, 在测试过程中, 均使用 NIST 与国密推荐的测试参数, 两项测试所选取的显著性水平均取 0.01, 分布均匀性显著性水平均取 0.0001. 根据测试的结果来分析 QRE-ChaCha 算法生成密钥是否具有随机性. 两种测试套件的测试结果与对比如表3和表4所示.

为了限制测试文件的大小, 在 NIST 随机性测试过程中只使用 1000 组 1000000 比特长度的密钥流序列进行随机性测试, 表3给出了部分测试的结果, NonOverlappingTemplate 只选取了大于等于显著性水平样本数最小的一项测试结果, RandomExcursions 与 RandomExcursionsVariant 测试结果较多并未列出, 但密钥流均通过了此三项测试中的所有测试项.

根据测试结果显示, QRE-ChaCha8 所生成的密钥流均通过了 NIST 随机性测试与国密随机性测试, 从通过显著性水平的样本数角度看 QRE-ChaCha8 的随机性要略好于 ChaCha8. 这证明密钥流序列具有良好的随机性, 满足密码学应用的要求, 并且 QRE-ChaCha 算法本身在设计过程中就充分考虑了随机性问题, 通过对 ChaCha 原始算法的优化, 注入了量子随机数可以增加更多的混淆, 从而显著增强了算法输

出密钥流的随机性. 密钥流具备极佳的随机分布性质, 序列中不存在明显的统计规律可被利用. 这从根本上保证了 QRE-ChaCha 算法抵御了各种统计攻击和密码分析攻击, 说明增加量子随机数的方式可以一定程度上增加算法的安全性.

表 **3** 1000 组 QRE-ChaCha8 密钥流 NIST 随机性测试结果
Table 3 1000 sets of QRE-ChaCha8 keystream NIST randomness test results

NIST 随机性测试	≥ 显著性水平样本数		分布均匀性	
	QRE-ChaCha8	**ChaCha8**	**QRE-ChaCha8**	**ChaCha8**
Frequency	982	988	0.187581	0.467322
BlockFrequency	988	993	0.751866	0.388990
CumulativeSums	983	988	0.435430	0.353733
Runs	994	990	0.062821	0.781106
LongestRun	984	989	0.747898	0.664168
Rank	990	991	0.784927	0.763677
FFT	986	987	0.803720	0.281232
NonOverlappingTemplate	982	980	0.940080	0.989055
OverlappingTemplate	990	991	0.117432	0.286836
Universal	990	989	0.012829	0.236810
ApproximateEntropy	990	10	0.345650	0.653773
Serial	992	992	0.899171	0.213309
LinearComplexity	987	986	0.115387	0.794391

6 QRE-ChaCha 性能测试分析

为了深入分析 QRE-ChaCha 算法的性能表现, 本文采用软件加密同样大小的文件所用的时间作为评估标准, 测试环境为: AMD Ryzen 7 5700U with Radeon Graphics 处理器、主频 1.80 GHz,64 位 Windows 10 22H2 企业版操作系统, 基于 x64 环境,16GB 内存, 使用 C 语言进行软件层面的算法加解密.

本文在进行性能测试时, 依旧使用 8 轮版本的 QRE-ChaCha 与 8 轮版本的 ChaCha 进行加解密对比测试, 同时加入 20 轮版本的 ChaCha 进行参照对比. 每种算法都使用随机生成的种子密钥对 10MB、20MB、30MB、40MB 和 50MB 大小的相同文件分别进行 5 次加密测试, 并取加密时间的平均值, 作为最终的性能测试的结果, 结果如表5所示.

根据性能测试的结果来看, QRE-ChaCha8 的加密时间几乎和 ChaCha8 一致. 本文所进行的性能测试, 并没有将量子随机数生成器产生量子随机数的时间考虑在内, 只是针对算法本身的结构进行测试分析. 所以, 可以说明 QRE-ChaCha 的加解密性能并没有随着量子随机数的引入而降低, 在提升一定的安全性的同时性能依旧能保持算法本身的高效性.

7 总结

本文通过引入量子随机数来增强 ChaCha 算法的安全性, 提出了一种创新性的密码算法——量子随机数增强的 ChaCha (QRE-ChaCha) 流密码算法. 本文所提出的 QRE-ChaCha 算法, 使用量子随机数通过异或组件作用于初始状态矩阵中的初始常量, 从而增强了种子密钥的随机性, 同时对于奇数轮输出的中间状态矩阵, 使用量子随机数通过异或组件对状态矩阵中的起始 128bit 数据进行增强, 结合算法轮变换操作, 使得量子随机数的真随机性扩散至整个状态矩阵空间, 从而增强算法的安全性.

在安全性分析方面, 本文采用了基于 SAT 自动化搜索方法对 QRE-ChaCha 算法进行了差分和线性密码分析测试. 通过与原始 ChaCha 算法的对比, QRE-ChaCha 在抗差分攻击和抗线性攻击的能力上显

表 **4** 10000 组 QRE-ChaCha8 密钥流国密随机性测试结果

Table 4 10,000 sets of QRE-ChaCha8 keystream state secret randomness test results

国密随机性测试	≥ 显著性水平样本数		分布均匀性	
	QRE-ChaCha8	**ChaCha8**	**QRE-ChaCha8**	**ChaCha8**
单比特频数检测	9884	9892	0.862398	0.438383
块内频数检测 $m = 10000$	9902	9898	0.969009	0.588514
扑克检测 $m = 4$	9889	9892	0.469806	0.636911
扑克检测 $m = 8$	9910	9899	0.362434	0.601351
重叠子序列检测 $m = 3$ P1	9890	9897	0.978538	0.76875
重叠子序列检测 $m = 3$ P2	9901	9897	0.211848	0.127393
重叠子序列检测 $m = 5$ P1	9918	9906	0.61007	0.279706
重叠子序列检测 $m = 5$ P2	9915	9913	0.90688	0.225644
游程总数检测	9915	9907	0.113239	0.401375
游程分布检测	9900	9888	0.399442	0.645657
块内最大 1 游程检测 $m = 10000$	9900	9873	0.386748	0.044797
块内最大 0 游程检测 $m = 10000$	9902	9898	0.65086	0.140054
二元推导检测 $k = 3$	9905	9897	0.699313	0.090826
二元推导检测 $k = 7$	9889	9907	0.669151	0.633579
自相关检测 $d = 1$	9915	9907	0.073281	0.390374
自相关检测 $d = 2$	9898	9895	0.187378	0.714252
自相关检测 $d = 8$	9902	9896	0.128354	0.762307
自相关检测 $d = 16$	9893	9898	0.846168	0.456314
矩阵秩检测	9902	9884	0.008056	0.040318
累加和前向检测	9885	9897	0.39437	0.135487
累加和后向检测	9889	9896	0.447116	0.640035
近似熵检测 $m = 2$	9890	9897	0.981469	0.802608
近似熵检测 $m = 5$	9915	9902	0.216485	0.981258
线性复杂度检测 $m = 500$	9882	9878	0.526907	0.878465
线性复杂度检测 $m = 1000$	9887	9896	0.155238	0.586861
Maurer 通用统计检测 $L = 7$ $Q = 1280$	9892	9874	0.621922	0.017151
离散傅里叶检测 $m = 500$	9892	9890	0.294959	0.331564

表 **5** QRE-ChaCha8、ChaCha8 与 ChaCha20 加密测试时间对比结果
Table 5 QRE-ChaCha8, ChaCha8 and ChaCha20 encryption test time comparison results

文件大小	加密时间 (s)		
	QRE-ChaCha8	**ChaCha8**	**ChaCha20**
10MB	0.1037854	0.1051830	0.2025330
20MB	0.2096104	0.2115156	0.4061916
30MB	0.3118018	0.3147998	0.6116970
40MB	0.4168038	0.4228406	0.8162400
50MB	0.5273160	0.5308238	1.0211580

示出了显著提升. 具体来说, QRE-ChaCha 算法在 2 轮和 3 轮差分路线的平均概率上界分别为 2^{-4} 和 2^{-25}, 而 ChaCha 算法对应的概率上界分别为 2^{-2} 和 2^{-12}. 此外, QRE-ChaCha 算法在 20 轮时的概率上界为 2^{-154}, 远小于 ChaCha 算法的 2^{-74}. 为了进一步验证 QRE-ChaCha 算法的安全性, 本文还采用了 NIST 统计测试套件和国密规范对算法生成的密钥流进行了随机性检测. 测试结果表明, QRE-ChaCha 算法生成的密钥流具有良好的随机性, 通过了包括单比特频数检测、块内频数检测、游程总数检测等在内的多项统计测试, 证明了其在密钥流随机性方面的优越性.

在性能评估方面, 本文通过测量 QRE-ChaCha 算法的加密速度, 评估了其加解密性能. 测试结果表明, QRE-ChaCha 算法能够 ChaCha 算法的保持加解密性能不降低. 具体来说, QRE-ChaCha8 版本的加密时间与 ChaCha8 版本几乎一致, 表明量子随机数的引入并未对算法的性能产生负面影响. 这一点在 10MB 至 50MB 不同大小文件的加密测试中得到了验证.

本文提出的 QRE-ChaCha 算法在继承了 ChaCha 算法高效性的同时, 通过量子随机数的引入, 有效提升了算法的安全性. 这不仅体现在抗差分攻击和抗线性攻击的能力提升上, 也体现在密钥流的高随机性上. 性能测试结果也证实了 QRE-ChaCha 算法在实际应用中的可行性, 在提升安全性的同时, 依然保持了良好的加解密速度. 此外, QRE-ChaCha 算法作为一种的随机数扩张算法, 为量子随机数在更广泛领域的应用提供了新的可能性.

参考文献

[1] BERNSTEIN D J, LANGE T. Post-quantum cryptography[J]. Nature, 2017, 549(7671): 188-194. [DOI: 10.1038/nature23461]

[2] BERNSTEIN D J, HENINGER N, LOU P, et al. Post-quantum RSA[C]. In: Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings 8. Springer Cham, 2017: 311-329. [DOI: 10.1007/978-3-319-59879-6_18]

[3] JAO D, DE FEO L. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies[C]. In: Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4. Springer Berlin Heidelberg, 2011: 19-34. [DOI: 10.1007/978-3-642-25405-5_2]

[4] MICCIANCIO D, REGEV O. Lattice-based cryptography[M]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[5] JI Z, QIAO Y, SONG F, et al. General linear group action on tensors: A candidate for post-quantum cryptography[C]. In: Theory of Cryptography Conference. Springer Cham, 2019: 251-281. [DOI: 10.1007/978-3-030-36030-6_11]

[6] BENNETT C H, BRASSARD G. Quantum cryptography: Public key distribution and coin tossing[J]. Theoretical computer science, 2014, 560: 7-11. [DOI: 10.1016/j.tcs.2014.05.025]

[7] PIRANDOLA S, ANDERSEN U L, BANCHI L, et al. Advances in quantum cryptography[J]. Advances in optics and photonics, 2020, 12(4): 1012-1236. [DOI: 10.1364/aop.361502]

[8] PORTMANN C, RENNER R. Security in quantum cryptography[J]. Reviews of Modern Physics, 2022, 94(2): 025008. [DOI: 10.1103/revmodphys.94.025008]

[9] LI C L, ZHANG K Y, ZHANG X, et al. Device-independent quantum randomness–enhanced zero-knowledge proof[J]. Proceedings of the National Academy of Sciences, 2023, 120(45): e2205463120. [DOI: 10.1073/pnas.2205463120]

[10] LO H K, CHAU H F. Unconditional security of quantum key distribution over arbitrarily long distances[J]. science,

1999, 283(5410): 2050-2056. [DOI: 10.1126/science.283.5410.2050]
[11] LO H K, MA X, CHEN K. Decoy state quantum key distribution[J]. Physical review letters, 2005, 94(23): 230504. [DOI: 10.1103/physrevlett.94.230504]
[12] ACÍN A, BRUNNER N, GISIN N, et al. Device-independent security of quantum cryptography against collective attacks[J]. Physical Review Letters, 2007, 98(23): 230501. [DOI: 10.1103/physrevlett.98.230501]
[13] LO H K, CURTY M, QI B. Measurement-device-independent quantum key distribution[J]. Physical review letters, 2012, 108(13): 130503. [DOI: 10.1103/physrevlett.108.130503]
[14] PROCTER G. A Security Analysis of the Composition of ChaCha20 and Poly1305[J/OL]. IACR Cryptology ePrint Archive, 2014: 2014/613. https://eprint.iacr.org/2014/613.pdf
[15] LANGLEY A, CHANG W, MAVROGIANNOPOULOS N, et al. ChaCha20-Poly1305 cipher suites for transport layer security (TLS)[R]. IETF RFC 7095. 2016. https://www.rfc-editor.org/rfc/rfc7905
[16] BERNSTEIN D J. ChaCha, a variant of Salsa20[J/OL]. Workshop record of SASC. 2008, 8(1): 3-5. https://cr.yp.to/chacha/chacha-20080120.pdf
[17] KEBANDE V R. Extended-Chacha20 Stream Cipher With Enhanced Quarter Round Function[J]. IEEE Access, 2023. [DOI: 10.1109/access.2023.3324612]
[18] JAIN D K, MOHAN P, LAKSHMANNA K, et al. Enhanced data privacy in cyber-physical system using improved Chacha20 algorithm[J]. 2022. [DOI: 10.21203/rs.3.rs-1558846/v1]
[19] MAHDI M S, HASSAN N F, ABDUL-MAJEED G H. An improved chacha algorithm for securing data on IoT devices[J]. SN Applied Sciences, 2021, 3(4): 429. [DOI: 10.1007/s42452-021-04425-7]
[20] MANNALATHA V, MISHRA S, PATHAK A. A comprehensive review of quantum random number generators: Concepts, classification and the origin of randomness[J]. Quantum Information Processing, 2023, 22(12): 439. [DOI: 10.1007/s11128-023-04175-y]
[21] MA X, YUAN X, CAO Z, et al. Quantum random number generation[J]. npj Quantum Information, 2016, 2(1): 1-9. [DOI: 10.1038/npjqi.2016.21]
[22] HERRERO-COLLANTES M, GARCIA-ESCARTIN J C. Quantum random number generators[J]. Reviews of Modern Physics, 2017, 89(1): 015004. [DOI: 10.1103/revmodphys.89.015004]
[23] IAVICH M, KUCHUKHIDZE T, OKHRIMENKO T, et al. Novel quantum random number generator for cryptographical applications[C]. In: 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T). IEEE, 2020: 727-732. [DOI: 10.1109/picst51311.2020.9467951]
[24] IAVICH M, KUCHUKHIDZE T, IASHVILI G, et al. Hybrid quantum random number generator for cryptographic algorithms[J]. Radioelectronic and Computer Systems, 2021 (4): 103-118. [DOI: 10.32620/reks.2021.4.09]
[25] STIPČEVIĆ M. Quantum random number generators and their applications in cryptography[C]. In: SPIE Defense, Security, and Sensing. 2012: 1-1. [DOI: doi.org/10.1117/12.919920]
[26] HUANG L, ZHOU H, FENG K, et al. Quantum random number cloud platform[J]. npj Quantum Information, 2021, 7(1): 1-7. [DOI: 10.1038/s41534-021-00442-x]
[27] KUANG R, LOU D, HE A, et al. Pseudo quantum random number generator with quantum permutation pad[C]. In: 2021 IEEE international conference on quantum computing and engineering (QCE). IEEE, 2021: 359-364. [DOI: 10.1109/qce52317.2021.00053]
[28] IAVICH M, KUCHUKHIDZE T, GNATYUK S, et al. Novel certification method for quantum random number generators[J]. International Journal of Computer Network and Information Security, 2021, 13(3): 28-38. [DOI: 10.5815/ijcnis.2021.03.03]
[29] DRAHI D, WALK N, HOBAN M J, et al. Certified quantum random numbers from untrusted light[J]. Physical Review X, 2020, 10(4): 041048. [DOI: 10.1103/physrevx.10.041048]
[30] NAJM Z, JAP D, JUNGK B, et al. On comparing side-channel properties of AES and ChaCha20 on microcontrollers[C]. In: 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). IEEE, 2018: 552-555. [DOI: 10.1109/apccas.2018.8605653]
[31] KUMAR S V D, PATRANABIS S, BREIER J, et al. A practical fault attack on arx-like ciphers with a case study on chacha20[C]. In: 2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC). IEEE, 2017: 33-40. [DOI: doi.org/10.1109/fdtc.2017.14]
[32] TORDSSON P. Partitioning oracle attacks against variants of AES-GCM and ChaCha20-Poly1305[J/OL]. Linnaeus University, Department of Mathematics. 2021. https://lnu.diva-portal.org/smash/get/diva2:1562903/FULLTEXT01.pdf
[33] DEGABRIELE J P, GOVINDEN J, GÜNTHER F, et al. The security of ChaCha20-Poly1305 in the multi-user setting[C]. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. 2021: 1981-2003. [DOI: 10.1145/3460120.3484814]

[34] BARBERO S, BELLINI E, MAKARIM R H. Rotational analysis of ChaCha permutation[J]. Advances in Mathematics of Communications, 2023, 17(6): 1422-1439. [DOI: 10.3934/amc.2021057]

[35] CENTELLAS CLAROS L S, BLANCO COCA L, SANDOVAL ALCOCER J P. Comparative study of the symmetric cryptography algorithms AES, 3DES and ChaCha20[J/OL]. Acta Nova, 2022, 10(3): 283-302. ISSN 1683-0789. http://www.scielo.org.bo/pdf/ran/v10n3/1683-0789-ran-10-03-283.pdf

[36] AUMASSON J P, FISCHER S, KHAZAEI S, et al. New features of Latin dances: analysis of Salsa, ChaCha, and Rumba[C]. In: Fast Software Encryption: 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers 15. Springer Berlin Heidelberg, 2008: 470-488. [DOI: 10.1007/978-3-540-71039-4_30]

[37] SHI Z, ZHANG B, FENG D, et al. Improved key recovery attacks on reduced-round Salsa20 and ChaCha[C]. In: International Conference on Information Security and Cryptology. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012: 337-351. [DOI: 10.1007/978-3-642-37682-5_24]

[38] CHOUDHURI A R, MAITRA S. Differential Cryptanalysis of Salsa and ChaCha–An Evaluation with a Hybrid Model[J/OL]. IACR Cryptology ePrint Archive, 2016: 2016/377. https://eprint.iacr.org/2016/377.pdf

[39] DEEPTHI K K C, SINGH K. Cryptanalysis of Salsa and ChaCha: revisited[C]. In: International Conference on Mobile Networks and Management. Springer Cham, 2017: 324-338. [DOI: 10.1007/978-3-319-90775-8_26]

[40] MIYASHITA S, ITO R, MIYAJI A. PNB-focused differential cryptanalysis of ChaCha stream cipher[C]. In: Australasian Conference on Information Security and Privacy. Springer Cham, 2022: 46-66. [DOI: 10.1007/978-3-031-22301-3_3]

[41] BELLINI E, GERAULT D, GRADOS J, et al. Boosting differential-linear cryptanalysis of ChaCha7 with MILP[J]. IACR Transactions on Symmetric Cryptology, 2023. [DOI: 10.46586/tosc.v2023.i2.189-223]

[42] GHAFOORI N, MIYAJI A. Higher-Order Differential-Linear Cryptanalysis of ChaCha Stream Cipher[J]. IEEE Access, 2024. [DOI: 10.1109/access.2024.3356868]

[43] MAOLOOD A T, GBASHI E K, MAHMOOD E S. Novel lightweight video encryption method based on ChaCha20 stream cipher and hybrid chaotic map[J]. International Journal of Electrical & Computer Engineering (2088-8708), 2022, 12(5). [DOI: 10.11591/ijece.v12i5.pp4988-5000]

[44] FU K, WANG M, GUO Y, et al. MILP-based automatic search algorithms for differential and linear trails for speck[C]. In: Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers 23. Springer Berlin Heidelberg, 2016: 268-288. [DOI: 10.1007/978-3-662-52993-5_14]

[45] ETH ZÜRICH. Quantum RNG[EB/OL]. 2024. http://qrng.ethz.ch/live/

[46] BASSHAM L E, RUKHIN A L, SOTO J, et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications[J/OL]. National Institute of Standards and Technology, 2010. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf

[47] Trisia. randomness, Version 1.5.0[EB/OL]. 2023. https://github.com/Trisia/randomness

[48] 密码行业标准化技术委员会. 随机性检测规范: GM/T 0005-2021[S]. 中国标准出版社. 2021.

作者信息

刘超 (2002–)，浙江金华人，本科生在读．主要研究领域为量子密码．
liu.chao@hdu.edu.cn

贾晨浩 (2000–)，山东聊城人，硕士生在读．主要研究领域为对称密码的分析与设计．
222270059@hdu.edu.cn

崔婷婷 (1990–)，山东青岛人，副教授．主要研究领域为对称密码的设计和分析．
cuitingting@hdu.edu.cn

赵帅 (1992–)，河南永城人，讲师．主要研究领域为量子信息物理学和量子密码．
zhaoshuai@hdu.edu.cn

胡耿然 (1989–)，浙江丽水人，副教授．主要研究领域为格上密码学、区块链技术及其应用．
grhu@hdu.edu.cn