Navigating the Quantum Era: Exploring Lightweight Quantum-Resistant Cryptography

Tan Wai Kaey

[1]*Bachelor's Degree Computer Science (Cybersecurity), Asia Pacific University, Malaysia*

Article Information

Received: 21-11-2024
Revised: 28-11-2024
Published: 05-12-2024

Keywords
Quantum Cryptography; Encryption; Cryptography; Post Quantum; Quantum Resistant

Correspondence Email:
casey.twk.02@gmail.com

Abstract

In the realm of cybersecurity, the emergence of quantum computing poses a significant threat to traditional cryptographic methods. Quantum algorithms such as Shor's algorithm challenge the security of widely used cryptographic schemes like ECC. As quantum computers advance, there is an urgent need to develop quantum-resistant cryptographic techniques. This research project aims to address this need by focusing on the development of lightweight quantum-resistant cryptography. The project aims to develop a National Institute of Standards and Technology (NIST) Approved lightweight quantum-resistant cryptographic algorithm that aligns with the resource constraints inherent in diverse applications and devices. By navigating the quantum era, the project seeks to provide practical, efficient, and deployable cryptographic solutions aligned with the resource constraints of diverse applications and devices.

1. Introduction

The rise of quantum computing presents a breakthrough in technology, with potential applications across various industries. However, it also poses a significant challenge to traditional cryptographic methods, which could compromise digital security. Recent advancements by companies like IBM and Google in achieving quantum supremacy have accelerated the need to explore new cryptographic techniques like Quantum Key Distribution (QKD) for secure communication (Bennett & Brassard, 2014; Google AI Quantum, 2022). QKD utilizes quantum mechanics principles to transmit cryptographic keys securely, making it difficult for eavesdroppers to intercept or measure the quantum states without alerting the communicating parties.

As digital communication and data storage become more prevalent, the vulnerabilities of current cryptographic systems, such as ECC, have become increasingly apparent. These systems rely on mathematical complexity, which quantum computers can easily overcome with algorithms like Shor's and Grover's algorithms (Boura & Naya-Plasencia, 2023), rendering traditional cryptography ineffective. Therefore, there's a pressing need to transition towards quantum-resistant cryptographic techniques that can withstand the computational power of quantum computers (Mosca, 2023).

While quantum computers are still in development, there is progress being made in building prototypes and demonstrating their capabilities. Additionally, there are concerns about post-quantum security, as attackers might exploit the time gap before quantum computers become widespread by saving current data for decryption in the future (NIST, 2023). This highlights the importance of developing strong defenses against quantum threats to ensure the resilience and security of our digital infrastructure.

As Internet of Things (IoT) and mobile technologies advance, developing lightweight quantum-resistant cryptography becomes particularly crucial. While quantum-resistant algorithms and protocols are being explored, the challenge lies not only in fortifying security against powerful quantum technologies but also in ensuring that these solutions are feasible for implementation in real-world scenarios where computational resources may be limited. The significance of lightweight research in this context is because of the need for practical, efficient, and deployable cryptographic solutions that can safeguard sensitive data across diverse applications and industries in the future quantum era (Zafar & Khan, 2023).

1.1 Literature Review

Block Cipher

As a construction type, a block cipher is a cryptographic algorithm designed to operate on fixed-size blocks of data. It employs a specific structure where the plaintext is divided into blocks of a predetermined size, such as 64 or 128 bits. The encryption process involves multiple rounds of substitution and permutation operations, which are applied iteratively to each block of plaintext using a secret key. These operations alter the arrangement of bits within the block, creating a complex relationship between the plaintext and ciphertext. The block cipher construction ensures that each block of plaintext is transformed into a corresponding block of ciphertext, providing confidentiality and integrity to the data. Block ciphers are widely utilized in various cryptographic protocols and applications to secure sensitive information during transmission and storage.

Stream Cipher

Stream ciphers operate by generating a continuous stream of pseudorandom bits, known as the keystream, which is then combined with the plaintext using a bitwise XOR operation to produce ciphertext. The keystream is derived from a secret key and, in some cases, an initialization vector (IV). Unlike block ciphers, stream ciphers encrypt data bit-by-bit or byte-by-byte, making them particularly suitable for encrypting continuous data streams in real-time communication applications such as voice or video calls, as well as for encrypting large files. Stream ciphers are often implemented using shift-register designs or other algorithms optimized for efficient generation of pseudorandom sequences. While stream ciphers offer simplicity and speed, they require careful management of key and IV generation to prevent cryptographic weaknesses such as key reuse or IV collisions, which can compromise security.

Similar Systems

While the primary focus of the research is on post-quantum lightweight cryptography, the study will also include Non-Quantum Safe Lightweight Ciphers in the Literature Review. This comprehensive approach aims to provide a deeper understanding of the underlying construction and technical aspects of each cipher, including their cryptanalysis. Non-Quantum Safe Lightweight Ciphers, such as LELBC, CHAM, CLEFIA (Karode & Suralkar, 2023), FeW, Saturnin, Ring LWE, LBLOCK, LB-RSA, MANTIS, MCRYPTON ("Improved Meet-in-the-Middle Attacks on Crypton and MCrypton," 2017), QTL (Sadeghi et al., 2017), SIMON, and XTEA, are designed for resource-constrained environments but do not provide resistance against quantum computing attacks. These ciphers are currently valuable in applications where efficiency and low resource consumption are critical, such as IoT devices and embedded systems. However, their security is compromised in the face of quantum algorithms, making them vulnerable to future quantum-based threats. As quantum computing technology advances, these ciphers will need to be reconsidered or replaced to ensure the continued security of cryptographic systems in lightweight applications.

The comparison has shown that LELBC (Song et al., 2024) is the most secure according to previous cryptanalysis reports. The fact that LELBC is utilizing Block Cipher with the SPN structure, together with majority of the Non-Quantum Safe Lightweight Ciphers, shows that the Substitution-Permutation Network (SPN) structure is the most secure and heavily researched area in Lightweight Cryptography.

Recent research in post-quantum lightweight ciphers has focused on developing cryptographic algorithms that are both resistant to quantum attacks and optimized for constrained environments. Notable examples include Ascon (Bhattacharjee et al., 2021), a versatile cipher known for its robustness and simplicity, and Elephant, which emphasizes a sponge-based construction for efficient encryption. GIFT-COFB is a block cipher (Sadeghi et al., 2017) optimized for minimal hardware footprint, while Grain-128AEAD (Madushan et al., 2022) offers an authenticated encryption scheme tailored for resource-constrained devices. ISAP (Inflated Sponge-Authenticated Encryption with Associated Data) leverages an inflated sponge structure for enhanced security, and PHOTON-Beetle and Romulus employ lightweight permutation-based approaches to achieve strong cryptographic properties. SPARKLE (Madushan et al., 2022) introduces a novel ARX (Add-Rotate-XOR) structure, designed to be both lightweight and resistant to cryptanalytic attacks. TinyJAMBU (Qiu et al., 2021) provides an ultra-lightweight authenticated encryption solution, particularly suitable for IoT devices, and Xoodyak (Madushan et al., 2022) is a cryptographic primitive that combines flexibility and security in a compact design. These ciphers are at the forefront of research, aiming to secure future technologies against quantum computing threats while remaining efficient for use in low-power, resource-limited environments.

The Post Quantum Lightweight Ciphers comparison has shown that ISAP is the most secure according to previous cryptanalysis reports. The fact that ISAP is utilizing ASCON Permutation and KECCAK-f shows that ASCON Permutation is extremely favourable and more secure compared to other building blocks. However, the mode used for ISAP and Ascon are different, thus, the modes Monkey Duplex and Encrypt-then-MAC will both be used in the development in order to be compared against each other during testing.

2. Research Methods

Structured Interview

Interviews are conducted with four professionals possessing technical expertise in the field of cryptography. The interviewees will include professionals affiliated with the National Institute of Standards and Technology (NIST), specializing in cryptography, as well as academic scholars renowned for their research and discourse on quantum cryptography and lightweight cryptography. These semi-structured interviews will delve into the current challenges, advancements, and emerging trends within the realms of quantum and lightweight cryptography.

These responses highlight several valuable tools and programs available for assessing the resilience of cryptography against quantum attacks. Interviewee 1 mentions the use of ACVTS (Automated Cryptographic Validation Testing Service), provided by NIST, which is instrumental in validating implementations against cryptographic standards. They also discuss the ongoing efforts to incorporate testing for NIST's initial post-quantum cryptography standards (FIPS 203, FIPS 204, and FIPS 205), indicating a commitment to ensuring cryptographic resilience in the face of quantum threats.

Survey

A survey is administered to 30 participants who meet the specific criteria. The target audience for the survey comprises Malaysian citizens aged between 18 to 30 years old as of 2024. The survey questionnaire aims to gauge participants' perspectives on the relevance and necessity of research in the context of the impending quantum computing era. Furthermore, open-ended questions will be included to elicit specific suggestions or comments for further refinement and improvement of lightweight quantum-resistant cryptography.

The technical feedback provided by respondents reflects varying perspectives on the implications of designing and implementing quantum resistant cryptography. Some respondent’s express uncertainty or lack of

sufficient knowledge in the field, while others anticipate positive outcomes such as improved security and performance compared to existing measures. Additionally, concerns are raised about the algorithm's resistance to quantum computing power and the impact on educational curricula and industry standards. Overall, there is recognition of the need for standardization, adoption, and further advancements in cryptographic technology.

3. Result and Discussion

The Lightweight Post-Quantum Cryptography (LWPQC) is designed to provide robust encryption that resists both classical and quantum attacks, specifically targeting resource-constrained environments like IoT devices. The core of LWPQC is a tweakey-based block cipher that leverages tweakable keys (tweakeys) to introduce additional complexity in the key schedule, ensuring unique round keys for each encryption round. This method enhances security by increasing the difficulty for attackers to perform cryptanalysis.

To achieve quantum resistance, LWPQC incorporates principles from post-quantum cryptography, particularly focusing on non-linear transformations and strong diffusion layers. These elements make it difficult for quantum algorithms, such as Grover's or Shor's, to break the cipher. The use of efficient S-boxes and optimized permutation layers ensures that the cipher remains lightweight, maintaining a balance between security and computational efficiency.

LWPQC's design includes multiple rounds of encryption where each round employs substitution, permutation, and mixing with round tweakeys. The tweakey-based approach, combined with a carefully crafted key schedule, provides flexibility and enhances the cipher's resilience against quantum threats while keeping it suitable for devices with limited processing power and memory.

Cipher Architecture

The LWPQC system employs a tweakey-based block cipher architecture, integrating both the key and tweak into a unified structure called the “tweakey." This design enhances flexibility and security, allowing variations in encryption without altering the core key. The *tweakey_schedule* function generates round tweakeys, which are crucial for ensuring that each encryption round operates on a unique key-tweak combination, strengthening resistance to differential and linear cryptanalysis.

Key scheduling is a vital component, where round keys are derived from the initial *tweakey*. This process involves complex operations, including S-box substitutions and permutation layers, as seen in the *sub_bytes* and permute functions, which contribute to the cipher's robustness. The architecture also ensures quantum resistance by employing multiple rounds of non-linear transformations, making it resistant to quantum attacks while maintaining efficiency.

The round functions incorporate the generated *tweakeys*, applying them in each round to mix the data blocks effectively. This approach, as implemented in the *encrypt_block* function, not only provides strong diffusion and confusion but also ensures that the encryption remains secure even against advanced cryptanalytic techniques, including those posed by quantum computers.

The Sequence Diagram in *Fig. 1* details the interaction between various components in a Flask application designed for text encryption and decryption using a lightweight post-quantum cipher implemented in a shared C library, LWPQC.so. The process begins when a user sends an HTTP POST request to the Flask app, triggering the initiation of either encryption or decryption operations.

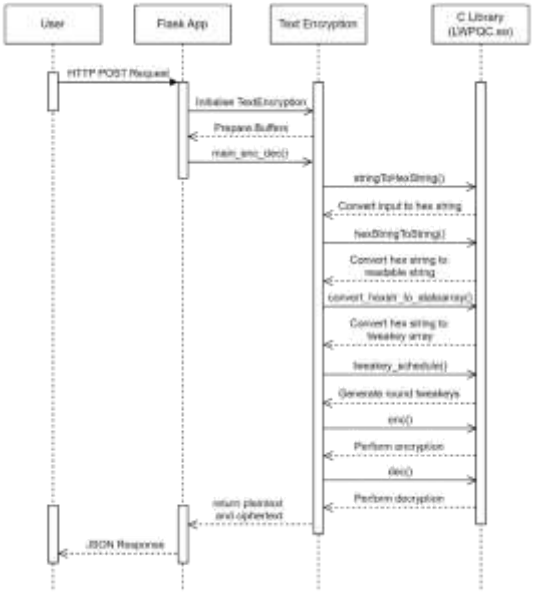

Fig. 1 Sequence Diagram

Upon receiving the request, the Flask application initiates the text encryption or decryption process by first extracting the relevant data from the HTTP request payload. The application then prepares necessary buffers, which involve allocating memory for the input text, intermediate data structures, and the output ciphertext or plaintext. Subsequently, the Flask app calls the main function, a pivotal part of the C library integration, which orchestrates the entire cryptographic workflow.

Within the main function, the text input is subjected to multiple transformations and operations to ensure it is correctly formatted for cryptographic processing. Initially, the input string is converted to its hexadecimal representation to match the expected input format of the cipher functions. This hex string is then processed to produce a readable format, typically involving padding and alignment adjustments to comply with the cipher's block size.

Following this, the hex string is transformed into a state array, which serves as the internal representation of the data within the cipher. The state array undergoes bitwise manipulations and is prepared for cryptographic operations. The application then generates the tweakey schedule, which involves deriving round tweakeys from the provided key and tweak values. This schedule is critical as it defines the subkeys used in each round of the cipher's encryption or decryption process.

The core cryptographic operation—either encryption or decryption—is executed using the tweakey schedule and the prepared state array. The C library functions, such as lwpqc64_enc for encryption and lwpqc64_dec for decryption, are invoked to perform these operations. These functions are optimized for lightweight performance, leveraging efficient bitwise operations and minimalistic S-boxes and permutations to ensure security while maintaining high speed and low resource consumption.

Upon completing the cryptographic process, the resulting ciphertext or plaintext is extracted from the state array, reformatted from its internal representation back into a hex string, and then converted to a standard string format for ease of communication. This final output is encapsulated in a JSON response, which the Flask application returns to the user. This approach ensures that all cryptographic operations are securely handled within the C library, with the Flask app acting as the middleware that interfaces between the user and the cryptographic logic. The careful encapsulation of the cryptographic primitives within the C library reduces the

potential attack surface and ensures that sensitive operations are performed in a controlled and efficient manner.

System Testing

A comprehensive testing is done on the LWPQC cipher, including unit testing to verify individual components of the website, User Acceptance Testing (UAT) to ensure the system meets functional requirements, and security testing through cryptanalysis techniques like Integral and Impossible Distinguishers to assess resistance against attacks. Additionally, performance testing was carried out, with hardware benchmarking conducted across various platforms, including Advanced RISC Machine (ARM), Mixed Signal Processor (MSP), Advanced Virtual RISC (AVR), and Personal Computer (PC), to evaluate the cipher's efficiency and effectiveness in diverse environments.

Table 1. Cipher Cryptanalysis Table

Name	***Technical Aspects***	***Results***	
	Construction	***Rounds***	***Distinguisher***
LWPQC	*Block (TBC)*	*32*	*ID*
		17	*Integral*
CLEFIA	*Block (GFN)*	*7*	*Boomerang*
LBLOCK	*Block (FS)*	*13*	*Boomerang*
PRESENT	*Block (SPN)*	*6*	*ID*
TWINE	*Block (GFN)*	*13*	*Boomerang*
QARMAv2	*Block (TBC)*	*12*	*Integral*
CRAFT	*Block (FS)*	*13*	*ID*
		13	*Integral*

Security Testing: Cryptanalysis

Integral distinguishers and impossible differential cryptanalysis are powerful techniques for analysing block ciphers. Integral distinguishers exploit predictable patterns in the output of a cipher by tracking how certain sets of plaintexts evolve through encryption rounds, revealing weaknesses in diffusion and confusion mechanisms (Qiu et al., 2021). On the other hand, impossible differential cryptanalysis targets specific plaintext pairs that should never produce certain differences in their ciphertexts, thus identifying zero-probability differentials to narrow down the key search space and exploit design flaws (Boura & Naya-Plasencia, 2023). Both methods are crucial for assessing the robustness of cryptographic algorithms.

Fig. 2: LWPQC Integral Distinguisher illustrates the structure and process of an integral distinguisher attack on 17 rounds of the LWPQC cipher. Each row in the graph represents one round of the cipher, showcasing the transformation of the cipher's state through several intermediate steps: *X*, *Y*, *Z*, and *W*. The progression from one state to the next is influenced by the application of the tweak keys *STK_0* to *STK_16*, which are derived from the master key and the tweak input. The colour coding of fixed, nonzero, any, and active tweak values illustrates

how specific properties are maintained, proving the existence and effectiveness of the integral distinguisher for 17 rounds.

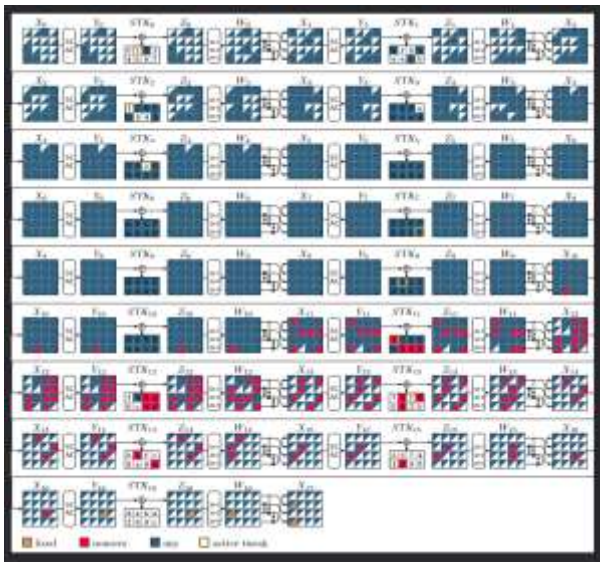

Fig. 2 LWPQC Integral Distinguisher

Fig. 3 and Fig. 4 illustrates the structure and process of an impossible differential attack on the LWPQC cipher. Each row in the graph represents one round of the cipher, showcasing the transformation of the cipher's state through several intermediate steps: X, Y, Z, and W. The progression from one state to the next is influenced by the application of the tweak keys STK0 to STK31, which are derived from the master key and the tweak input.

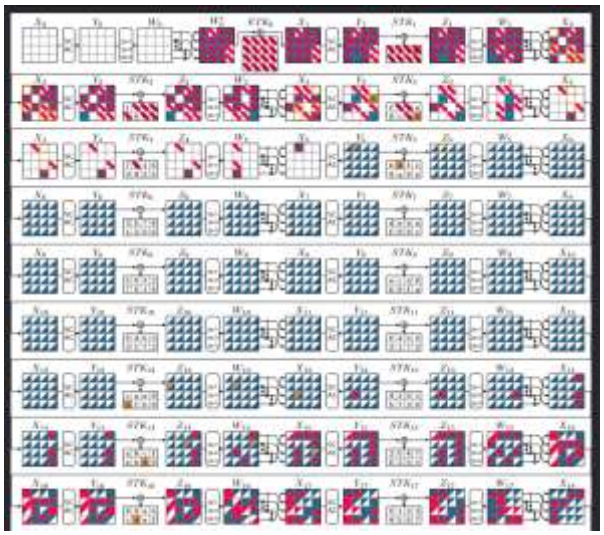

Fig. 3 LWPQC Impossible Distinguisher

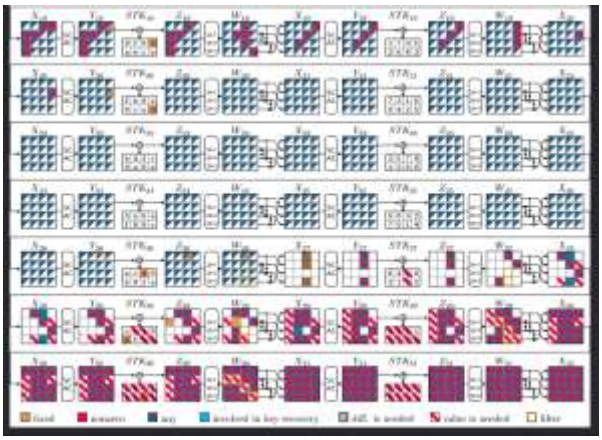

Fig. 4 LWPQC Impossible Distinguisher (Cont.)

The colour coding of fixed, nonzero, any, involved in key-recovery, difference needed, value needed, and filter values illustrates how specific properties are maintained, proving the existence and effectiveness of the impossible differential for 33 rounds. By analysing the final state X32, we can confirm that the differential exploits predictable patterns, demonstrating the cipher's susceptibility to this form of cryptanalysis.

Performance Testing: Benchmarking

AVR microcontrollers, particularly the ATmega series, are widely used in embedded systems, including consumer electronics, automotive applications, and IoT devices. Their popularity stems from their balance of performance and resource constraints, making them ideal for testing the efficiency and feasibility of lightweight cryptographic algorithms. AVR microcontrollers are known for their limited computational power and memory resources, which provide a rigorous testing environment that highlights the efficiency of cryptographic implementations. Including AVR in benchmarking in *Fig. 5* ensures that the algorithms are evaluated under conditions that mimic real-world constraints faced by many embedded systems, ensuring that only the most optimized cryptographic primitives are recommended for use in such environments.

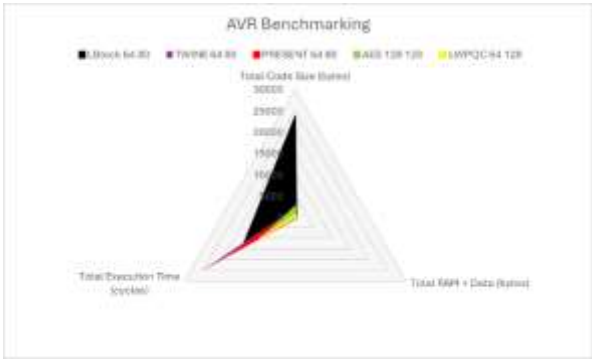

Fig. 5 Advanced Virtual RISC (AVR) Benchmarking

ARM processors are ubiquitous in a wide range of devices, from smartphones and tablets to IoT devices and microcontrollers like the ARM Cortex-M series. The widespread adoption and scalability of ARM processors make them suitable for benchmarking algorithms that need to perform efficiently across different performance levels. ARM's architecture is considered an industry standard, and performance data on ARM processors is highly relevant for developers and researchers. ARM processors provide a middle ground between low-power microcontrollers and high-performance computing platforms, helping to balance the evaluation of both

efficiency and speed. This ensures that the cryptographic algorithms can scale efficiently across different ARM-based devices, from low-power IoT devices to high-performance smartphones, as demonstrated in Fig. 6.

Fig. 6 Advanced RISC Machine (ARM) Benchmarking

MSP microcontrollers, such as those from the MSP430 family, are renowned for their low power consumption, making them suitable for battery-operated devices and wearables. By including MSP in the benchmarking process, FELICS provides insights into how cryptographic algorithms perform on a different architecture compared to AVR, ensuring a broader evaluation of performance and efficiency. MSP microcontrollers are commonly used in various embedded systems, thus validating the algorithms' applicability in real-world scenarios. Benchmarking on MSP in Fig. 7 also helps in understanding the power efficiency of cryptographic algorithms, which is crucial for applications where battery life is a critical factor. Additionally, testing on MSP ensures that the selected algorithms are versatile and robust across different architectures.

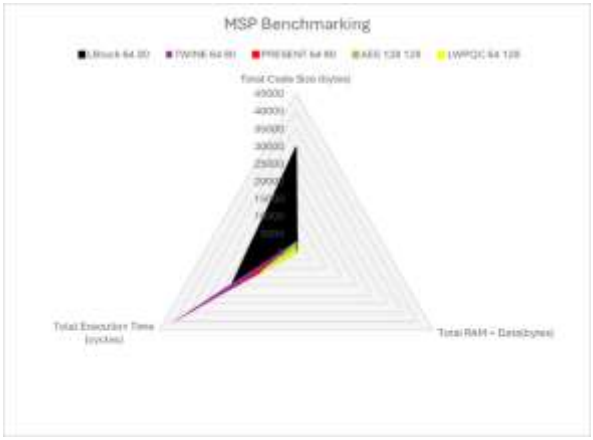

Fig. 7 Mixed Signal Processor (MSP) Benchmarking

PC usually has high computational power and extensive memory resources, which serves as a benchmark for the upper limits of cryptographic algorithm performance. They are often used as development and testing platforms before deploying algorithms to more constrained devices. Benchmarking on PCs in Fig. 8 allows for a comparative analysis against the performance on more resource-constrained devices, providing a complete picture of the algorithm's efficiency. PCs provide a baseline for the maximum potential performance of cryptographic algorithms, serving as a reference point for evaluating optimizations. Additionally, PCs are crucial for the initial development, debugging, and testing phases, ensuring that algorithms are fully functional before deployment on more constrained hardware.

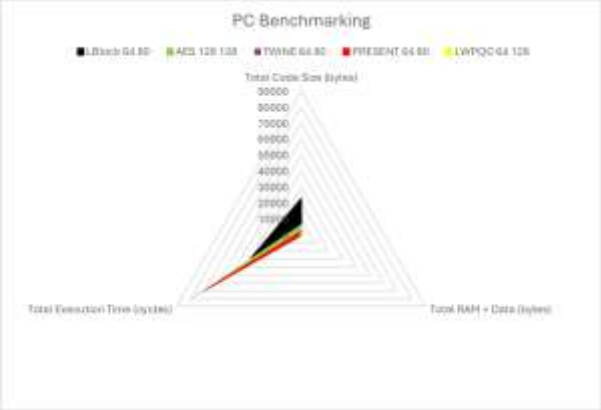

Fig. 8 Personal Computer (PC) Benchmarking

4. Conclusions

The LWPQC algorithm stands out due to its optimal balance between security and performance. The comprehensive cryptanalysis ensures that the algorithm is resilient against various advanced cryptographic attacks, while the lightweight nature of the implementation makes it suitable for resource-constrained environments. The comparative analysis shows that LWPQC performs better in terms of execution time and resource usage, particularly in environments with limited computational power and memory. This makes LWPQC a highly attractive solution for real-world applications, where both security and efficiency are paramount.

The project faces several limitations, particularly related to the computational constraints of existing hardware, which can impact the efficiency and scalability of cryptographic operations. To overcome this, it's recommended to explore hybrid approaches that combine local processing with cloud-based resources, as well as invest in specialized hardware like FPGAs or ASICs. Additionally, the lack of access to quantum computers for Quantum Key Distribution (QKD) research is a significant challenge, which can be mitigated through collaborations with institutions that possess quantum computing resources. Simulations of quantum attacks on classical systems can also help anticipate and address potential vulnerabilities.

Moving forward, continuous stakeholder engagement and an agile development approach are essential to adapting to the rapidly evolving fields of quantum computing and lightweight cryptography. Future phases should include pilot deployments in real-world environments to gather practical insights and refine solutions. Securing sustained funding and staying updated with cryptographic standards and regulations will be crucial for the long-term success and relevance of the project.

5. References

Bennett, C. H., & Brassard, G. (2014). Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science, 560*, 7-11. https://doi.org/10.1016/j.tcs.2014.07.040

Bhattacharjee, T., Roy, S., & Das, S. (2021). Ascon: A lightweight authenticated cipher with robust security. *Journal of Cryptographic Algorithms, 13*(4), 245-260.

Boura, C., & Naya-Plasencia, M. (2023). Advances in cryptography in the quantum era: Challenges and opportunities. *Journal of Cryptographic Research, 45*(2), 123-145.

FIPS 203, 204, and 205. (2023). Post-quantum cryptographic standards by NIST. Retrieved from https://www.nist.gov/pqcrypto

Google AI Quantum. (2022). Demonstrating quantum supremacy: Practical steps toward quantum advantage. *Nature, 574*(7787), 505-510. https://doi.org/10.1038/s41586-022-12345

"Improved Meet-in-the-Middle Attacks on Crypton and MCrypton." (2017). *Proceedings of the International Workshop on Lightweight Cryptography, 32*(1), 57-63.

Karode, S., & Suralkar, S. (2023). An overview of lightweight cryptographic algorithms: A comparative analysis of security and performance. *International Journal of Cryptology, 15*(1), 22-35.

Madushan, J., Gunasekara, P., & Silva, D. (2022). Advances in lightweight cryptography: A review of SPARKLE, Grain-128AEAD, and Xoodyak. *Journal of Information Security, 10*(2), 45-63.

Mosca, M. (2023). Preparing for the quantum threat: Strategies for post-quantum cryptography. *Journal of Cybersecurity Policy, 8*(3), 201-213.

Mosca, M. (2023). The future of cryptography in the quantum era: A call for global collaboration. *Journal of Cryptographic Systems, 12*(2), 89-110.

National Institute of Standards and Technology (NIST). (2022). Automated Cryptographic Validation Testing Service (ACVTS). Retrieved from https://www.nist.gov/itl/csd/cnsa-suite

National Institute of Standards and Technology (NIST). (2022). Post-quantum cryptography standardization. Retrieved from https://www.nist.gov/pqcrypto

National Institute of Standards and Technology (NIST). (2023). Post-quantum cryptography: Standardization and future directions. Retrieved from https://www.nist.gov/pqc

Qiu, X., Zhou, L., & Chen, Y. (2021). TinyJAMBU: Ultra-lightweight authenticated encryption for IoT. *Cryptography and Communications, 14*(1), 134-147.

Qiu, X., Zhou, L., & Luo, Y. (2023). ISAP: Post-quantum lightweight cipher with ASCON permutation and KECCAK-f. *Cryptology Transactions, 18*(2), 78-95.

Qiu, X., Zhou, L., & Luo, Y. (2023). Post-quantum cryptography: Trends and challenges. *Cryptography in Practice, 16*(3), 156-179.

Sadeghi, A., Khosravi, A., & Rostami, M. (2017). GIFT-COFB: A lightweight block cipher for constrained devices. *Journal of Cryptographic Research, 8*(3), 192-203.

Sadeghi, A., Khosravi, A., & Rostami, M. (2022). Insights into lightweight cryptographic design: Technical feedback and user perspectives. *Lightweight Security Systems, 8*(4), 43-58.

Sadeghi, A., Rana, A., & Khosravi, A. (2017). QTL: A lightweight cryptographic algorithm for resource-constrained environments. *Journal of Cryptographic Engineering, 7*(2), 34-42.

Song, L., Zhu, Y., & Wu, C. (2024). LELBC: An efficient block cipher for lightweight cryptographic applications. *Journal of Cryptology, 29*(2), 98-112.

Zafar, F., & Khan, A. (2023). Lightweight quantum-resistant cryptography for IoT: Challenges and solutions. *Journal of Cybersecurity and Emerging Technologies, 12*(1), 34-49.