International IT Journal of Research (IITJR), ISSN: 3007-6706
Volume 3, Issue 3, July-September, 2025
Available online at: https://itjournal.org/index.php

Post-Quantum Lattice-Based Cryptography: Hardware Acceleration and Latency Optimization for Edge Devices

Dr. Lukas Bergmann

Senior Research Scientist, Institute for IT Security and Cryptography, Technical University of Munich, Germany

ABSTRACT

The rapid advancement of quantum computing presents significant challenges to conventional cryptographic systems, particularly those based on integer factorization and discrete logarithm problems. Quantum algorithms, such as Shor's algorithm, have the potential to compromise widely deployed public-key cryptographic techniques, creating an urgent need for quantum-resistant security mechanisms. Post-quantum cryptography (PQC) has emerged as a promising solution, with lattice-based cryptographic algorithms receiving considerable attention due to their strong mathematical foundations, computational efficiency, and resistance to known quantum attacks. However, deploying these algorithms on resource-constrained edge devices introduces challenges related to computational complexity, memory consumption, energy efficiency, and communication latency.

This research explores hardware acceleration and latency optimization techniques for implementing lattice-based cryptographic algorithms in edge computing environments. The proposed framework investigates optimized implementations of lattice-based key encapsulation and digital signature schemes, particularly Module-Lattice-Based Key Encapsulation Mechanism (ML-KEM) and Module-Lattice-Based Digital Signature Algorithm (ML-DSA). It examines hardware acceleration techniques using Field-Programmable Gate Arrays (FPGAs), Application-Specific Integrated Circuits (ASICs), and specialized processor architectures to improve cryptographic execution speed while minimizing resource utilization. Furthermore, optimization strategies involving Number Theoretic Transform (NTT), parallel processing, memory-efficient polynomial arithmetic, instruction-level optimization, and hardware-software co-design are considered to reduce computational overhead and overall cryptographic latency.

The study proposes a performance evaluation framework based on execution time, encryption and decryption latency, throughput, energy consumption, memory utilization, hardware resource requirements, and security effectiveness. Particular emphasis is placed on balancing cryptographic security with the limited processing capabilities of Internet of Things (IoT) devices, wireless sensor networks, and industrial edge computing platforms. The anticipated findings suggest that hardware-assisted lattice-based cryptographic implementations can significantly improve processing efficiency compared with conventional software-only approaches, although actual performance gains depend on hardware architecture, algorithm parameters, and implementation techniques.

The research highlights the importance of developing scalable, energy-efficient, and low-latency post-quantum cryptographic solutions for securing next-generation edge infrastructures. The proposed approach contributes to the advancement of quantum-resistant security architectures by providing an optimization-oriented framework for integrating lattice-based cryptography into resource-constrained computing environments.

Keywords: Post-Quantum Cryptography (PQC), Lattice-Based Cryptography, Hardware Acceleration, Latency Optimization, Edge Computing

INTRODUCTION

The rapid evolution of quantum computing is transforming the landscape of modern information technology while introducing significant security challenges for existing cryptographic systems. Traditional public-key cryptographic algorithms, including Rivest–Shamir–Adleman (RSA), Elliptic Curve Cryptography (ECC), and Diffie–Hellman key exchange, rely on mathematical problems that are computationally difficult for classical computers. However, the development of powerful quantum computers presents a potential threat to these cryptographic mechanisms. Quantum algorithms, particularly Shor's algorithm, can efficiently solve integer factorization and discrete logarithm problems, potentially compromising the confidentiality, integrity, and authenticity of sensitive digital information. Consequently, the development of quantum-resistant cryptographic techniques has become an important research priority for securing future communication networks and computing infrastructures.

Post-Quantum Cryptography (PQC) represents a new generation of cryptographic approaches designed to withstand attacks from both classical and quantum computing systems. Among the different PQC techniques, lattice-based cryptography has emerged as one of the most promising approaches because of its strong mathematical foundations, computational efficiency, and flexibility. It is based on hard mathematical problems involving high-dimensional lattices, including the Shortest Vector Problem (SVP), Learning with Errors (LWE), and Module Learning with Errors (Module-LWE). These mathematical problems are believed to remain computationally difficult even for large-scale quantum computers. Furthermore, lattice-based cryptographic algorithms support multiple security applications, including key encapsulation, digital signatures, secure authentication, and encrypted communication.

The standardization of lattice-based cryptographic algorithms has further accelerated their adoption in modern cybersecurity architectures. The National Institute of Standards and Technology (NIST) standardized ML-KEM (formerly CRYSTALS-Kyber) for secure key establishment and ML-DSA (formerly CRYSTALS-Dilithium) for digital signatures in 2024. These algorithms provide practical mechanisms for implementing quantum-resistant security in cloud computing, distributed systems, Internet of Things (IoT) networks, and edge computing environments. Nevertheless, their deployment in resource-constrained devices introduces several technical challenges that require further investigation.

Edge computing has become an essential component of modern digital infrastructure because it enables data processing closer to the source of data generation rather than depending entirely on centralized cloud servers. Applications such as smart manufacturing, autonomous vehicles, healthcare monitoring, smart cities, industrial automation, and wireless sensor networks require fast communication, low response times, and reliable security mechanisms. Edge devices generally operate with limited computational resources, restricted memory capacity, and strict energy requirements. Therefore, implementing computationally intensive post-quantum cryptographic algorithms on these devices can increase processing delays, energy consumption, and system overhead.

PROPOSED MODELS AND METHODOLOGIES

The proposed research focuses on developing a hardware-accelerated and latency-optimized framework for implementing post-quantum lattice-based cryptographic algorithms on resource-constrained edge devices. The methodology combines standardized lattice-based cryptographic mechanisms, optimized mathematical operations, parallel hardware processing, and energy-efficient computing architectures. The primary objective is to improve cryptographic execution speed, reduce computational overhead, and maintain the security requirements of post-quantum communication systems.

The proposed framework primarily considers Module-Lattice-Based Key Encapsulation Mechanism (ML-KEM) and Module-Lattice-Based Digital Signature Algorithm (ML-DSA). These algorithms provide standardized quantum-resistant cryptographic functionality and are suitable candidates for evaluating secure communication in edge computing environments.

2.1 Proposed System Architecture

The proposed architecture consists of five major layers:

1. **Edge Device Layer:** Includes IoT sensors, embedded processors, industrial controllers, and low-power computing devices.
2. **Cryptographic Algorithm Layer:** Implements ML-KEM and ML-DSA for secure key establishment and digital authentication.
3. **Optimization Layer:** Performs polynomial arithmetic optimization, NTT acceleration, modular multiplication, and efficient memory management.
4. **Hardware Acceleration Layer:** Uses FPGA, ASIC, and specialized processor architectures for parallel cryptographic computations.
5. **Security and Performance Evaluation Layer:** Measures cryptographic latency, throughput, power consumption, memory utilization, and security effectiveness.

ML-KEM / ML-DSA Post-Quantum Cryptography IP - Post Quantum IP | Secure-IC
semiiphub.com

The architecture is designed to distribute computational tasks between software and hardware components. Computationally intensive operations are assigned to dedicated acceleration units, while algorithm control, parameter management, and communication functions remain within the embedded processor.

2.2 Selection of Lattice-Based Cryptographic Algorithms

The proposed methodology uses two standardized lattice-based cryptographic algorithms.

A. ML-KEM (Module-Lattice-Based Key Encapsulation Mechanism)
ML-KEM is designed to establish shared secret keys between communicating devices over potentially insecure networks. It is based on the Module-LWE problem and supports secure key establishment in quantum-resistant communication environments.

The major operations include:
- Key generation.
- Encapsulation.
- Decapsulation.
- Polynomial multiplication.
- Modular arithmetic.
- Noise sampling and encoding.

ML-KEM is particularly relevant for secure communication between IoT devices, edge gateways, and distributed industrial systems.

B. ML-DSA (Module-Lattice-Based Digital Signature Algorithm)
ML-DSA provides digital signature functionality for authentication, integrity verification, and message validation. It is based on structured lattice problems and is designed to resist known classical and quantum attacks.

Its major operations include:
- Public and private key generation.
- Digital signature generation.
- Signature verification.
- Polynomial vector operations.
- Hash-based processing.
- Rejection sampling.

The performance of both algorithms will be evaluated to identify the computational operations that benefit most from hardware acceleration.

2.3 Hardware Acceleration Model
The hardware acceleration model is designed to improve the computational efficiency of lattice-based cryptographic operations. Three hardware platforms are considered.

A. FPGA-Based Acceleration
Field-Programmable Gate Arrays provide reconfigurable hardware resources that can be customized for cryptographic computations. The proposed FPGA architecture includes dedicated arithmetic units for polynomial multiplication, modular reduction, and NTT operations.
Key implementation techniques include:
- Parallel polynomial multiplication.
- Pipelined arithmetic operations.
- Dedicated modular arithmetic units.
- Optimized memory access.
- Hardware-level instruction scheduling.

FPGA acceleration allows researchers to experiment with different hardware configurations while evaluating performance improvements.

B. ASIC-Based Acceleration
Application-Specific Integrated Circuits are considered for developing specialized cryptographic processing units. Unlike general-purpose processors, ASICs can be optimized for specific mathematical operations used in lattice-based cryptography.

The proposed ASIC-oriented architecture focuses on:
- Dedicated NTT processing units.
- Low-power modular multiplication circuits.
- Optimized polynomial arithmetic.
- Reduced data-transfer overhead.
- Efficient memory hierarchy.

ASIC implementations may offer improved energy efficiency and processing speed, although their development requires greater design effort and manufacturing costs.

C. Embedded Processor Optimization
Embedded processors are used to evaluate software-based implementations and hardware-assisted processing. Compiler optimization, efficient instruction scheduling, and processor-level parallelism are investigated to improve execution speed on resource-constrained devices.

2.4 Number Theoretic Transform (NTT) Optimization
The Number Theoretic Transform is a fundamental mathematical operation in lattice-based cryptographic algorithms. It enables efficient polynomial multiplication by transforming polynomial coefficients into a suitable computational domain.
Traditional polynomial multiplication can require substantial computational resources, particularly when polynomial degrees are large. NTT reduces this computational complexity and provides opportunities for parallel processing.
The proposed methodology investigates:

1. Parallel NTT computation.
2. Optimized modular multiplication.
3. Efficient twiddle-factor management.
4. Pipelined butterfly operations.
5. Memory-access optimization.
6. Reduced data movement between processing units.

Accelerating Post-Quantum Cryptography: A High-Efficiency NTT for ML-KEM on RISC-V

The optimization strategy aims to minimize the execution time of polynomial operations while maintaining mathematical correctness and cryptographic security.

2.5 Parallel Processing and Pipelining Model
Parallel processing is introduced to execute independent cryptographic operations simultaneously. The proposed model distributes polynomial arithmetic tasks across multiple processing units to improve computational throughput.
The methodology includes:

- Parallel execution of polynomial operations.
- Multi-stage pipelining for arithmetic computations.
- Concurrent memory access.
- Hardware resource scheduling.
- Reduction of idle processing cycles.

Pipelining divides complex operations into smaller execution stages. While one operation is being processed in a particular stage, another operation can enter the preceding stage. This improves hardware utilization and can increase overall throughput.

RESULTS & ANALYSIS

The results and analysis section evaluates the proposed hardware-accelerated lattice-based cryptographic framework for improving computational performance and reducing latency in resource-constrained edge devices. The evaluation focuses on standardized post-quantum cryptographic algorithms, particularly Module-Lattice-Based Key Encapsulation Mechanism (ML-KEM) and Module-Lattice-Based Digital Signature Algorithm (ML-DSA).

The proposed framework considers four implementation approaches: conventional software-based execution, optimized CPU implementation, FPGA-based hardware acceleration, and hardware–software co-design. These approaches are compared using cryptographic execution time, latency, throughput, energy consumption, memory utilization, and hardware resource requirements.

Since this research presents a proposed experimental framework rather than measured implementation results, the following analysis describes expected outcomes and illustrative comparisons. The numerical values are hypothetical and should not be interpreted as experimentally verified findings.

3.1 Computational Performance Analysis
Computational performance is one of the most important parameters for evaluating post-quantum cryptographic algorithms on edge devices. Lattice-based cryptographic operations involve polynomial multiplication, modular arithmetic, sampling, and transformation operations, which may introduce computational overhead.

The proposed optimization framework is expected to improve computational performance through parallel processing, pipelining, and hardware-assisted arithmetic operations.

The major observations anticipated include:

- Software-only implementations may experience higher computational overhead on low-power embedded processors.
- Optimized CPU implementations may reduce execution time through efficient instruction scheduling and memory management.
- FPGA acceleration may improve the execution speed of computationally intensive polynomial operations.
- Hardware–software co-design may provide a balanced solution by distributing computational workloads efficiently.

3.2 Cryptographic Latency Analysis

Latency is a critical performance parameter in edge computing applications because cryptographic operations must be completed within acceptable response times.

The proposed framework focuses on reducing latency in ML-KEM key generation, encapsulation, decapsulation, and ML-DSA signature generation and verification.
The primary sources of latency include:

1. Polynomial multiplication and transformation.
2. Modular arithmetic operations.
3. Memory access and data transfer.
4. Cryptographic sampling and hashing.
5. Communication between software and hardware components.

Hardware acceleration is expected to reduce the processing time of arithmetic-intensive operations. However, communication overhead between the embedded processor and hardware accelerator must also be considered.
The overall latency improvement will depend on the proportion of computation that can be accelerated and the efficiency of the hardware architecture.

3.3 Comparative Latency Analysis

The following table presents an illustrative comparison of the four proposed implementation approaches.

ILLUSTRATIVE DATA ONLY — NOT MEASURED EXPERIMENTAL RESULTS

Implementation model	Relative latency index	Performance interpretation
Software-only CPU	100	Baseline
Optimized CPU	80	Moderate improvement
FPGA acceleration	55	Significant improvement
Hardware–software co-design	45	Best illustrative result

3.4 Analysis of ML-KEM Performance

ML-KEM is evaluated based on three major cryptographic operations:

- Key generation.
- Encapsulation.
- Decapsulation.

Key generation involves polynomial sampling and arithmetic operations. Encapsulation and decapsulation require additional polynomial computations, encoding, decoding, and cryptographic processing.

The proposed methodology is expected to improve ML-KEM performance through optimized NTT operations, modular multiplication, and efficient memory access.

The expected performance benefits include:

- Reduced polynomial multiplication overhead.
- Faster key establishment.
- Improved processing efficiency.
- Reduced computational burden on embedded processors.
- Better suitability for secure edge-to-edge communication.

3.5 Analysis of ML-DSA Performance

ML-DSA is investigated to evaluate the effectiveness of hardware acceleration in digital signature operations.
Signature generation and verification involve polynomial vector operations, hashing, sampling, and arithmetic computations.

The proposed acceleration model focuses on optimizing computationally intensive operations while maintaining the correctness and security properties of the standardized algorithm.

The anticipated improvements include reduced signature generation time, faster verification, improved throughput, and more efficient utilization of hardware resources.

However, the actual performance improvement may differ between signature generation and verification because their computational workloads are not identical.

CONCLUSION

Post-Quantum Lattice-Based Cryptography represents a promising approach to addressing the emerging cybersecurity challenges associated with the advancement of quantum computing. Traditional cryptographic mechanisms based on integer factorization and discrete logarithm problems may become vulnerable to powerful quantum computing attacks, creating an urgent need for quantum-resistant security solutions. In this context, lattice-based cryptographic algorithms, particularly Module-Lattice-Based Key Encapsulation Mechanism (ML-KEM) and Module-Lattice-Based Digital Signature Algorithm (ML-DSA), provide standardized mechanisms for establishing secure communication and ensuring digital authentication.

This research explores hardware acceleration and latency optimization techniques for implementing lattice-based cryptographic algorithms in resource-constrained edge computing environments. The proposed framework integrates FPGA-based acceleration, ASIC-oriented architectures, optimized Number Theoretic Transform (NTT), parallel processing, pipelining, efficient modular arithmetic, and hardware–software co-design. These techniques aim to reduce computational overhead, improve cryptographic execution speed, and optimize hardware resource utilization.

The comparative analysis highlights the potential advantages of hardware-assisted implementations over conventional software-based approaches. In particular, FPGA acceleration offers flexibility and parallel processing capabilities, while ASIC architectures provide opportunities for specialized and energy-efficient cryptographic operations. Hardware–software co-design presents a balanced implementation strategy by distributing computationally intensive operations across dedicated hardware units and embedded processors. However, the effectiveness of these approaches depends on hardware architecture, algorithm parameters, memory organization, and implementation complexity.

REFERENCES

1. Ajtai, M. (1996). Generating hard instances of lattice problems. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (pp. 99–108). ACM. https://doi.org/10.1145/237814.237838
2. Alkim, E., Ducas, L., Pöppelmann, T., & Schwabe, P. (2016). Post-quantum key exchange—A new hope. In *25th USENIX Security Symposium* (pp. 327–343). USENIX Association.
3. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., Seiler, G., & Stehlé, D. (2020). *CRYSTALS-Kyber algorithm specifications and supporting documentation*. National Institute of Standards and Technology, Post-Quantum Cryptography Standardization Project.
4. Bos, J. W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., & Stehlé, D. (2018). CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)* (pp. 353–367). IEEE. https://doi.org/10.1109/EuroSP.2018.00032
5. Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). *Report on post-quantum cryptography* (NISTIR 8105). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8105
6. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., & Stehlé, D. (2018). CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018*(1), 238–268. https://doi.org/10.13154/tches.v2018.i1.238-268
7. Güneysu, T., Lyubashevsky, V., & Pöppelmann, T. (2012). Practical lattice-based cryptography: A signature scheme for embedded systems. In *Cryptographic Hardware and Embedded Systems—CHES 2012* (pp. 530–547). Springer. https://doi.org/10.1007/978-3-642-33027-8_31
8. Lyubashevsky, V., Peikert, C., & Regev, O. (2010). On ideal lattices and learning with errors over rings. In *Advances in Cryptology—EUROCRYPT 2010* (pp. 1–23). Springer. https://doi.org/10.1007/978-3-642-13190-5_1
9. Micciancio, D., & Regev, O. (2009). Lattice-based cryptography. In D. J. Bernstein, J. Buchmann, & E. Dahmen (Eds.), *Post-quantum cryptography* (pp. 147–191). Springer. https://doi.org/10.1007/978-3-540-88702-7_5
10. National Institute of Standards and Technology. (2024a). *Module-lattice-based key-encapsulation mechanism standard* (FIPS PUB 203). U.S. Department of Commerce. https://doi.org/10.6028/NIST.FIPS.203
11. National Institute of Standards and Technology. (2024b). *Module-lattice-based digital signature standard* (FIPS PUB 204). U.S. Department of Commerce. https://doi.org/10.6028/NIST.FIPS.204

12. National Institute of Standards and Technology. (2024c). *Stateless hash-based digital signature standard* (FIPS PUB 205). U.S. Department of Commerce. https://doi.org/10.6028/NIST.FIPS.205
13. Peikert, C. (2016). A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science, 10*(4), 283–424. https://doi.org/10.1561/0400000074
14. Pöppelmann, T., & Güneysu, T. (2014). Area optimization of lightweight lattice-based encryption on reconfigurable hardware. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 2796–2799). IEEE.
15. Regev, O. (2005). On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing* (pp. 84–93). ACM. https://doi.org/10.1145/1060590.1060603
16. Roy, S. S., Basso, A., Guo, J., & Verbauwhede, I. (2014). Compact ring-LWE cryptoprocessor. In *Cryptographic Hardware and Embedded Systems—CHES 2014* (pp. 371–391). Springer. https://doi.org/10.1007/978-3-662-44709-3_21
17. Oder, T., Schneider, T., Pöppelmann, T., & Güneysu, T. (2018). Practical CCA2-secure and masked ring-LWE implementation. *IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018*(1), 142–174. https://doi.org/10.13154/tches.v2018.i1.142-174
18. Bernstein, D. J., Buchmann, J., & Dahmen, E. (Eds.). (2009). *Post-quantum cryptography*. Springer. https://doi.org/10.1007/978-3-540-88702-7
19. Kannwischer, M. J., Rijneveld, J., Schwabe, P., & Stoffelen, K. (2020). PQM4: Post-quantum crypto library for the ARM Cortex-M4. In *Progress in Cryptology—AFRICACRYPT 2020* (pp. 15–34). Springer.
20. Banerjee, A., et al. (2026). *Post-quantum cryptography for engineers* (RFC 9958). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc9958.html