Performance Comparison of Lattice-Based Post-Quantum Digital Signatures: ML-DSA, FN-DSA, and HAETAE

Boyeon Song

Korea Institute of Science and Technology Information

bysong@kisti.re.kr

Abstract

In this paper, we compare the performance of lattice-based post-quantum digital signature schemes, namely, ML-DSA and FN-DSA from the National Institute of Standards and Technology (NIST) standardization process, and HAETAE from the Korea Post-Quantum Cryptography (KpqC) competition, in terms of key generation, signing, and verification.

Ⅰ. Introduction

PQM4 is a benchmarking and testing framework for post-quantum cryptography (PQC) on ARM Cortex-M4 microcontrollers, a widely used embedded processor in IoT and edge devices [1]. It was developed by the research group at Rheinland-Pfalz University, in collaboration with PQShield and others, to evaluate the performance of cryptographic algorithms, particularly those under standardization in the National Institute of Standards and Technology (NIST) PQC project, on resource-constrained embedded platforms [1]. PQM4 currently provides implementations of post-quantum key-encapsulation mechanisms and digital signature schemes targeting the ARM Cortex-M4 family. Its benchmark results are publicly available in [2].

In this paper, we evaluate the performance of lattice-based post-quantum digital signature schemes, namely ML-DSA and FN-DSA from the NIST PQC process, and HAETAE from the Korea Post-Quantum Cryptography (KpqC) competition, with respect to key generation, signing, and verification based on the PQM4 benchmark results.

Ⅱ. Lattice-Based Post-Quantum Signatures

In the present benchmark results, we compare the performance of lattice-based post-quantum digital signature schemes, including ML-DSA and FN-DSA from the NIST PQC standardization process, and HAETAE from the KpqC competition.

Table 1 presents CPU cycle counts for key generation (KeyGen), signing (Sign), and verification (Verify) operations of ML-DSA, HAETAE, and FN-DSA across different implementations from the PQM4 benchmark results [2]. Four implementation (IMP) types are considered: *clean* (straightforward reference implementation without optimizations), *m4f* (optimized for ARM Cortex-M4 with floating-point), *m4fstack* (stack-optimized variant), and *ref* (portable, unoptimized reference). The security category (C) corresponds to NIST PQC levels (1 to 5). Reported values in the table are mean cycle counts, measured over 1,000 executions for ML-DSA, 100 executions for HAETAE, and 10 executions for FN-DSA_provisional.

Table 1 Speed Comparison of Post-Quantum Signatures [2]

Scheme	C	IMP	KeyGen	Sign	Verify
ML-DSA 44	2	clean	1,874,405	7,925,955	2,063,096
		m4f	1,426,025	3,943,121	1,421,623
		m4f-stack	1,799,062	12,134,284	3,242,333
ML-DSA 65	3	clean	3,205,533	12,359,056	3,377,305
		m4f	2,516,006	6,193,171	2,415,944
		m4f-stack	3,412,622	24,421,526	5,732,397
ML-DSA 87	5	clean	5,341,863	15,579,513	5,610,203
		m4f	4,275,859	7,947,380	4,193,104
		m4f-stack	5,820,537	33,357,899	9,911,514
HAETAE 2	2	m4f	6,743,278	21,993,963	918,459
		ref	9,363,639	31,631,089	1,104,080
HAETAE 3	3	m4f	12,925,388	30,891,994	1,760,745
		ref	20,247,658	41,078,691	1,998,105
HAETAE 5	5	m4f	19,064,310	44,679,058	2,323,830
		ref	18,169,623	63,180,459	2,461,679
FN-DSA provisional 512	1	m4f	67,693,338	22,469,685	396,949
		ref	85,699,591	49,522,949	731,387
FN-DSA provisional 1024	5	m4f	308,608,613	48,321,135	793,856
		ref	274,928,016	107,512,779	1,461,795

In the PQM4 benchmark results, all three signature schemes were implemented using the *m4f* optimization, which consistently outperforms the *clean* or *ref* implementations. Therefore, our comparison focuses on the *m4f* results. Figure 1 presents a bar graph comparing the CPU cycle counts for KeyGen, Sign, and Verify operations across ML-DSA, HAETAE, and FN-DSA_provisional on the *m4f* implementation.

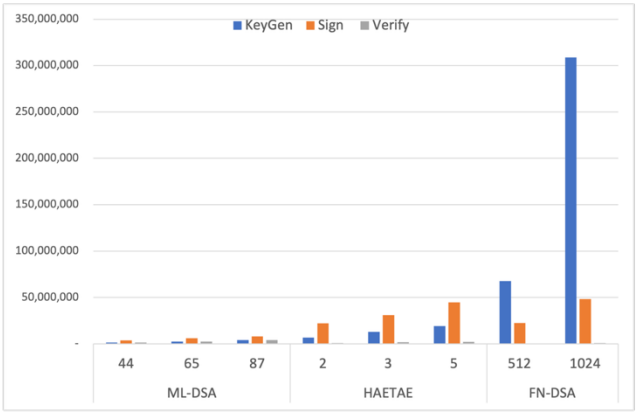

Figure 1 Speed Comparison of ML-DSA, HAETAE & FN-DSA

For clarity, each operation is further separated into individual bar graphs: Figure 2 for KeyGen cycles, Figure 3 for Sign cycles, and Figure 4 for Verify cycles.

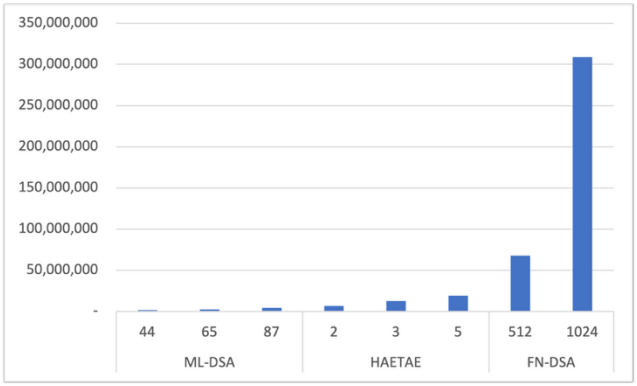

Figure 2 KeyGen Cycles of ML-DSA, HAETAE & FN-DSA

Figure 2 compares the KeyGen performance of the three schemes across different parameter sizes. ML-DSA achieves the lowest costs, with cycle counts ranging from approximately 1.4 million at parameter size 44 to 4.3 million at 87. HAETAE shows slightly higher values, increasing from 6.7 million at level 2 to 19.1 million at 5, yet still remaining within a few tens of millions. In contrast, FN-DSA incurs a dramatic increase, requiring 67.7 million cycles at size 512 and exceeding 300 million at 1024. Overall, ML-DSA is the most efficient for KeyGen, HAETAE offers moderate performance, and FN-DSA is most costly.

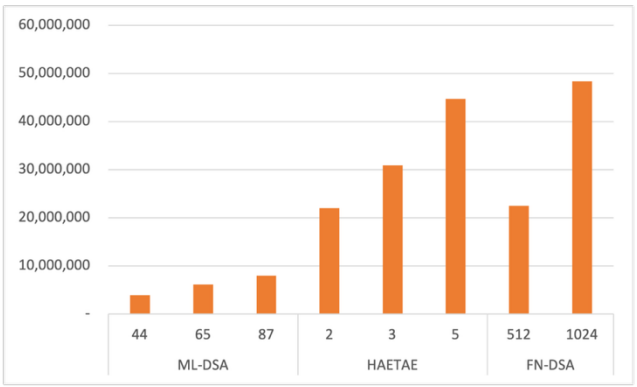

Figure 3 Sign Cycles of ML-DSA, HAETAE & FN-DSA

Figure 3 presents the Sign performance of the three schemes. ML-DSA again demonstrates the lowest Sign costs, increasing gradually from approximately 3.9 million cycles at size 44 to 7.9 million at 87. HAETAE requires higher costs, ranging from 22.0 million at level 2 to 44.7 million at 5, while FN-DSA shows comparable or larger values, with 22.5 million cycles at size 512 and 48.3 million at 1024. In summary, ML-DSA is the most efficient for Sign operation.

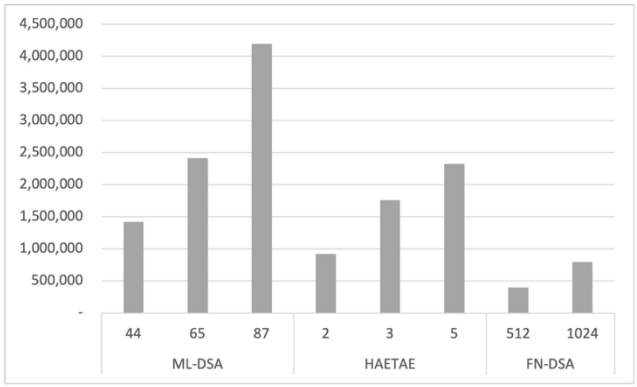

Figure 4 Verify Cycles of ML-DSA, HAETAE & FN-DSA

Figure 4 illustrates the Verify performance across the three schemes. FN-DSA achieves the lowest cost, requiring only 0.4 million cycles at size 512 and 0.8 million at 1024. HAETAE provides moderate performance, ranging from 0.9 million at level 2 to 2.3 million at level 5. In contrast, ML-DSA exhibits the highest verification cost, requiring 1.4 million cycles at size 44 to about 4.2 million at 87. Overall, FN-DSA shows the highest efficiency in verification.

Ⅲ. Conclusion

ML-DSA demonstrates efficient key generation and signing performance; however, its verification becomes increasingly costly at higher security levels compared with the other two schemes. HAETAE provides relatively faster verification than ML-DSA, though it is slower in both key generation and signing. In contrast, FN-DSA exhibits extremely high key generation costs, but offers very low verification overhead, making it unsuitable for frequent key creation on resource-constrained devices.

The optimized *m4f* implementations of all three schemes outperforms their *ref* or *clean* counterparts, highlighting the trade-offs between speed, resource usage, and suitability for embedded environments.

ACKNOWLEDGMENT

This research was supported by Korea Institute of Science and Technology Information (KISTI) (No. K25L5M2C2-01), and by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2025-02263666).

REFERENCES

[1] M. J. Kannwischer, M. Krausz, R. Petri and S. Yang, “pqm4: Benchmarking NIST Additional Post-Quantum Signature Schemes on Microcontrollers,” Cryptology ePrint Archive, Paper 2024/112, 2024.

[2] mupq/pqm4, https://github.com/mupq/pqm4