

Kire Jakimoski
Blashko Palitov

DOI: 10.2478/cmc-2026-0012

ZMOGLJIVOSTI KRIPTOGRAFSKEGA ALGORITMA CRYSTALS-DILITHIUM

PERFORMANCE CHARACTERISTICS OF THE CRYPTOGRAPHIC ALGORITHM CRYSTALS-DILITHIUM

Povzetek Crystals-Dilithium je postkvantni algoritem digitalnega podpisa, znan po varnosti, učinkovitosti in odpornosti proti kvantnim napadom. Študija ocenjuje čas izvajanja, uporabo CPU in RAM pri različnih velikostih sporočil, varnostnih ravneh (Dilithium-2, 3, 5) ter uporabi predhodnega zgoščevanja. Rezultati kažejo linearno rast časa s povečanjem velikosti sporočila, zmerne zahteve pri višjih ravneh in znatno pospešitev pri podpisovanju velikih datotek s predhodnim zgoščevanjem. Dilithium-2 je najbolj učinkovit, Dilithium-5 pa zagotavlja močnejšo zaščito z majhnim dodatnim bremenom. Poleg civilne uporabe je primeren tudi za vojaške sisteme, saj omogoča varno komunikacijo, dolgoročno zaupnost podatkov in interoperabilnost v okviru Natovih standardov.

Ključne besede *Postkvantna kriptografija, digitalni podpisi, analiza zmogljivosti, poraba CPU in RAM.*

Abstract CRYSTALS-Dilithium is a post-quantum digital signature algorithm valued for its security, efficiency, and resistance to quantum attacks. This study evaluates its performance in execution time, CPU, and RAM usage across different message sizes, security levels (Dilithium-2, 3, 5), and pre-hashing. The results show linear scaling with message size, modest overhead at higher levels, and major speed gains when pre-hashing large files. Dilithium-2 provides optimal efficiency, while Dilithium-5 delivers stronger protection with practical performance. Beyond civilian use, Dilithium is well-suited for military systems, enabling secure communications, long-term confidentiality of classified data, and NATO interoperability, even in constrained environments such as tactical radios, drones, and field IoT devices.

Key words *Post-quantum cryptography, digital signatures, performance analysis, CPU and memory usage.*

Introduction

With the rapid advancement of quantum computing, traditional cryptographic algorithms based on hard mathematical problems such as integer factorization and the discrete logarithm face serious concerns with regard to their long-term security. The digital signature schemes which currently underpin modern security infrastructures, such as RSA and ECDSA, are expected to become vulnerable once sufficiently powerful quantum systems emerge, primarily due to algorithms such as Shor's. Consequently, there is an urgent need for the development and standardization of post-quantum cryptographic algorithms – schemes capable of ensuring resilience against quantum attacks (Ducas et al., 2018a; Lyubashevsky et al., 2020; Land et al., 2021).

The U.S. National Institute of Standards and Technology (NIST) initiated and conducted a multi-year evaluation process of post-quantum candidates, involving several phases of public scrutiny and testing. In 2022, NIST officially selected CRYSTALS-Dilithium, Falcon, and SPHINCS+ as recommended digital signature schemes (Ducas et al., 2018a; Lyubashevsky et al., 2020; Land et al., 2021). Among them, Dilithium and Falcon are lattice-based constructions, while SPHINCS+ employs a hash-based framework. The formal standardization of Dilithium was later codified in NIST FIPS 204 (2024), which defines its parameters and establishes it as the baseline for post-quantum digital signatures.

CRYSTALS-Dilithium was selected as the main focus of this study because it is recognized as NIST's 'mainline' digital signature standard. What sets it apart is its relatively simple implementation, moderate memory usage, and the fact that it does not depend on specialized arithmetic (Lyubashevsky et al., 2020; Ducas et al., 2018a; Land et al., 2021; Li et al., 2023). On the other hand, Falcon offers shorter signatures but demands high-precision floating-point arithmetic, which makes secure deployment trickier. SPHINCS+, while being conservative and hash-based, faces challenges with much larger signature sizes and slower performance. Recent performance evaluations have confirmed that Dilithium is practical across various platforms, with Demir et al. (2025) pointing out its appealing balance between efficiency and security for industry applications. Meanwhile, ongoing research like that of Liu et al. (2025) emphasizes the need for careful implementation, as side-channel vulnerabilities can arise if proper countermeasures are not taken.

The shift to post-quantum cryptography is not just a technical upgrade; it is a crucial strategy we cannot afford to ignore. With quantum computing threatening the very foundation of classical public-key cryptography, military organizations – often the main targets for sophisticated adversaries – need to proactively embrace quantum-resistant solutions. By placing Dilithium in the context of this larger movement towards cryptographic modernization, this article adds to the vital conversations about how we can protect our national defence systems in the age of quantum technology.

This article aims to take a closer look at CRYSTALS-Dilithium, a post-quantum digital signature scheme, and evaluate how well it fits into military applications. It specifically tackles the key research question: How effectively does CRYSTALS-Dilithium balance security, efficiency, and ease of implementation to function as a reliable digital signature scheme for defence and national security systems in the post-quantum landscape?

To explore this question, the study conducts a hands-on evaluation of CRYSTALS-Dilithium's performance in various operational scenarios. The analysis focuses on three main metrics: the time it takes to sign and verify, how much processing power is used, and the amount of memory consumed. More specifically, it looks at (i) how message size affects computational demands and resource usage, (ii) performance differences between the Dilithium-2, Dilithium-3, and Dilithium-5 parameter sets, and (iii) the impact of pre-hashing, evaluating efficiency when signing pre-computed digests instead of raw messages.

The findings from this evaluation lead to a set of recommendations about which Dilithium parameter sets are best suited for different application contexts. These suggestions weigh security needs against operational performance, pinpointing situations where lighter configurations such as Dilithium-2 may be adequate, as well as cases where the stronger assurances of Dilithium-5 are more appropriate. In doing so, the analysis connects empirical performance results with the larger strategic goal of determining whether Dilithium can serve as a solid foundation for securing military communications, protecting classified information, and ensuring smooth interoperability within post-quantum defence systems.

The experimental evaluation was conducted using the dilithium-py library in a controlled environment. The benchmarks were performed on a contemporary computing platform equipped with an Intel Core i3 @2.5GHz processor, 8GB DDR3 RAM, and running Ubuntu Linux 22.04 (64-bit).

1 SECURITY DISCUSSION

1.1 Comparative analysis of Dilithium, Falcon, and SPHINCS+

To fully understand the strengths of CRYSTALS-Dilithium, it helps to compare it with the other two digital signature schemes recommended by NIST: Falcon and SPHINCS+. Each of these schemes comes with its own set of trade-offs with regard to security levels, performance, and how easy they are to implement (Digital Signatures for the Future: Dilithium, Falcon and SPHINCS+, 2023; Vidaković & Miličević, 2023).

Table 1:
Comparison
between
Dilithium, Falcon
and SPHINCS+

Scheme	Security Levels (NIST)	Public Key Size (approx.)	Signature Size (approx.)	Performance & Implementation Notes
Dilithium	Levels 2, 3, 5	1.3–2.6 KB	2.4–4.6 KB	Balanced efficiency and security; relatively simple to implement; resistant to side-channel attacks with proper countermeasures
Falcon	Levels 1,5	0.9–1.8 KB	0.7–1.3 KB	Very compact signatures; high efficiency; requires floating-point arithmetic, complicating secure hardware/software implementation
SPHINCS+	Levels 1, 3, 5	32–64 B	8–30 KB	Conservative, hash-based design; stateless; very large signatures and slower performance, but strong long-term security assurances

This comparison highlights why Dilithium was chosen as the primary NIST standard: it offers a practical balance between security, efficiency, and implementability, whereas Falcon is more fragile in deployment and SPHINCS+ is less efficient in constrained environments (NIST, 2024).

1.2 NIST Security Levels and Quantum Threats

The NIST security levels are standardized categories set by the U.S. National Institute of Standards and Technology (NIST) to evaluate the strength of cryptographic algorithms, including those designed for post-quantum security. These levels help us understand just how tough these algorithms are to crack. They act as a universal benchmark, allowing us to compare both classical and post-quantum algorithms in terms of their security equivalence.

Quantum Threats to Classical Cryptography

When it comes to Quantum Threats to Classical Cryptography, we need to consider two key players: RSA and ECC. Both of these cryptographic methods are at risk from Shor’s algorithm, which can efficiently factor large numbers and tackle discrete logarithms in polynomial time. This means that once we have large-scale quantum computers, RSA and elliptic-curve cryptography (ECC) could become insecure (Gitonga, 2025; Freemindtronic, 2024). On the other hand, we have Dilithium, which stands out for its resistance. It is built on module-lattice problems (Module-LWE and Module-SIS), and there is a strong belief that these problems can withstand both classical and quantum attacks. Unlike RSA and ECC, there are no known efficient quantum algorithms which can crack these issues (Bos et al., 2022).

1.3 CRYSTALS-Dilithium in Military Use

The use of CRYSTALS-Dilithium has significant consequences for the military and defence sectors, where both operational efficiency and long-term security are of the utmost importance.

Secure Communications: Military communication systems must remain confidential, even if they are intercepted and stored by adversaries. Dilithium plays a crucial role in ensuring resilience against “harvest now, decrypt later” tactics, where encrypted data is gathered now with a plan to decrypt it later, once quantum computers are up and running (NIST, 2024).

Long-Term Confidentiality of Classified Data: Sensitive archives, such as intelligence reports, satellite images, and strategic planning documents, often need to be safeguarded for many years. Dilithium at NIST Level 5 offers security comparable to AES-256, which meets the confidentiality standards for classified defence information (NIST, 2024; Demir et al., 2025).

Interoperability within NATO Standards: NATO is all about making sure that Allied forces can communicate securely, and a big part of that is cryptographic interoperability. They have set up NATO Interoperability Standards and Profiles (NISP), alongside the NATO Key Management Interoperability Specification (NKMIS), to help coalition partners share information safely (NATO, 2024; NATO, 2025). By adopting NIST-standardized post-quantum cryptography algorithms such as Dilithium, they are ensuring that different platforms and national infrastructures can work together seamlessly.

Deployment in Constrained Systems: Military IoT devices, drones, and tactical radios typically function with limited computing power and memory. Research indicates that Dilithium can be fine-tuned for these types of environments, but it requires careful engineering to handle memory limitations (Bos et al., 2022).

2 RESULTS OBTAINED FROM THE EXPERIMENT

The table below presents the results obtained from the testing (serial processing). It should be noted that the system on which the measurements were taken only supports AVX instructions. Although the execution times appear somewhat slower due to the system's older architecture, they remain relatively short.

Table 2:
Test Results)

File	Variant	Operation	Time (s)	CPU (%)	RAM (MB)
1 MB.zip	Dilithium-2	Key generation	0.0231	11.42	16.64
		Signing	0.0879	22.47	17.21
		Verification	0.0368	15.60	16.67
	Dilithium-3	Key generation	0.0382	15.23	16.67
		Signing	0.1265	23.20	17.55
		Verification	0.0451	17.60	16.66
	Dilithium-5	Key generation	0.0533	18.29	16.66
		Signing	0.1517	22.62	17.93
		Verification	0.0534	19.25	16.67
100 MB.zip	Dilithium-2	Key generation	0.0207	12.85	15.69
		Signing	0.8276	22.47	116.33
		Verification	0.7678	29.23	23.90
	Dilithium-3	Key generation	0.0308	15.15	15.70
		Signing	0.8709	24.25	116.33
		Verification	0.7765	27.68	64.51
	Dilithium-5	Key generation	0.0448	18.32	15.70
		Signing	0.8978	24.94	116.54
		Verification	0.7877	23.19	114.74
1000 MB.zip	Dilithium-2	Key generation	0.0214	10.75	15.76
		Signing	7.6172	29.51	1026.06
		Verification	7.5419	29.29	909.49
	Dilithium-3	Key generation	0.0308	13.51	15.74
		Signing	7.6286	29.49	1020.07
		Verification	7.5481	28.70	968.30
	Dilithium-5	Key generation	0.0462	17.14	15.76
		Signing	7.6722	29.51	1026.08
		Verification	7.5661	28.01	1048.59

2.1 Analysis of the Results (Different File Size)

The size of the message to be signed significantly affects the signing and verification time, as digital signatures process the entire content, usually via hashing. This directly impacts the duration of the cryptographic operations, especially with larger files.

This section analyses how increasing input file size influences:

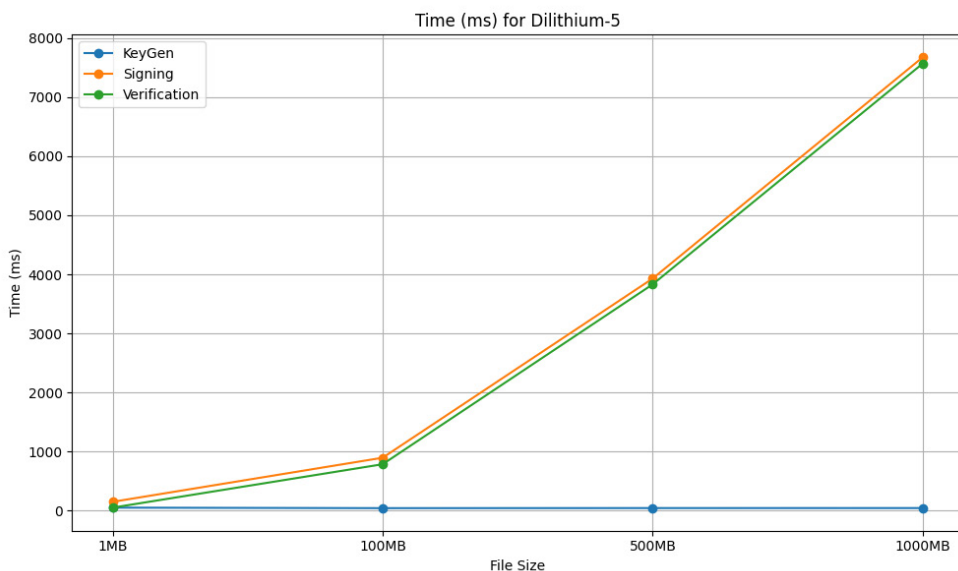
- Key generation, signing, and verification time (Section 2.1.1),
- CPU load (Section 2.1.2),
- RAM consumption (Section 2.1.3).

The analysis covers all three CRYSTALS-Dilithium security levels (D2, D3, D5), with results based on averages from 100 consecutive runs to ensure statistical reliability.

2.1.1 Analysis of the Impact of Message/File Size on Key Pair Generation Time, Signing Time, and Verification Time

For the CRYSTALS-Dilithium variants (Dilithium-2, Dilithium-3, Dilithium-5), the key generation time is very low (<0.05 seconds) and unaffected by file size, since it depends only on cryptographic operations defined by the algorithm's parameters. The differences between 1 MB and 1000 MB are negligible – for example, in Dilithium-2 the average remains ~0.02 seconds in all cases.

Figure 1:
File Size vs.
Execution Time
in CRYSTALS-
Dilithium 5



In contrast, signing and verification times grow significantly with file size. In Dilithium-2, signing increases from ~ 0.088 seconds at 1 MB to ~ 7.62 seconds at 1000 MB ($\approx 86\times$ increase). Verification shows the same trend, rising from ~ 0.037 seconds to ~ 7.54 seconds over the same range. The results clearly indicate an almost linear relationship: a tenfold input increase leads to a nearly tenfold execution increase. For instance, between 100 MB and 1000 MB, signing time grows by $\approx 9.2\times$. This trend occurs because the algorithm must process the entire message – typically via hashing – before generating or verifying a digital signature.

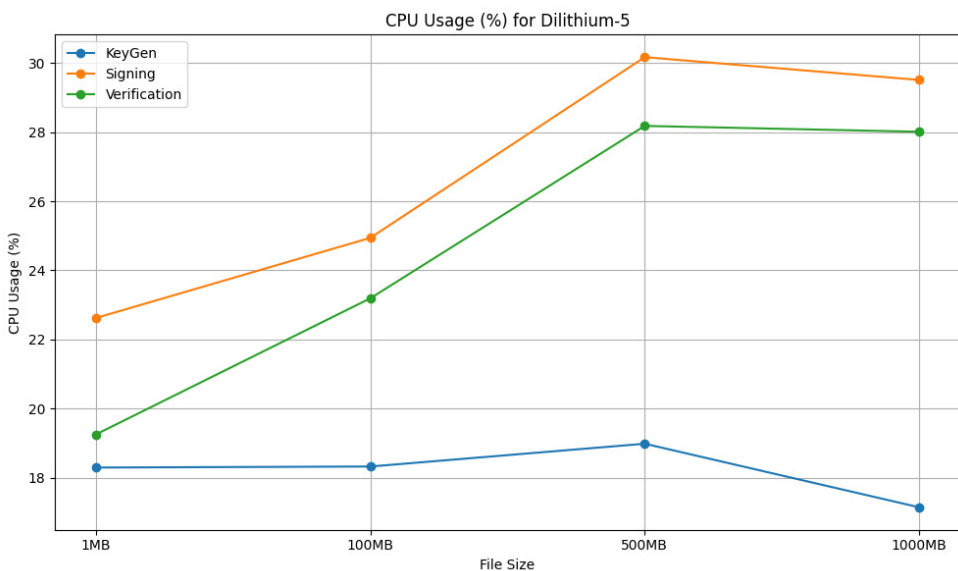
The results show that execution-time differences between Dilithium-2, Dilithium-3, and Dilithium-5 are more pronounced for small files, while they nearly vanish for the largest ones. For example, when signing a 1 MB message, Dilithium-5 is $\sim 72\%$ slower than Dilithium-2 (0.1517s vs. 0.0879s), and verification is $\sim 45\%$ slower (0.0534s vs. 0.0368s). However, at 1000 MB, signing times converge to ~ 7.6 seconds and verification to ~ 7.55 seconds across all variants. This indicates that for large files, message processing (I/O and hashing) dominates execution time, rendering the extra cryptographic cost of higher security levels negligible.

For key generation, higher levels show moderate increases: for instance, at 1 MB, Dilithium-5 requires 0.0533s compared to 0.0231s for Dilithium-2. Yet these differences, in the order of tenths of a second, remain insignificant in practice, especially since key generation is not performed frequently.

2.1.2 Analysis of the Impact of File Size on CPU Load during Key Pair Generation, Signing, and Verification

The CPU load during cryptographic operations increases with file size, particularly in the signing and verification phases, which involve intensive data processing. Table 1 shows the average CPU utilization values (in %) obtained from the same experiments across different file sizes and Dilithium security levels.

Figure 2:
File Size vs.
CPU Utilization
in CRYSTALS-
Dilithium 5



Key generation consumes relatively few resources (~10–19%), with no clear dependency on file size, since the process relies only on the algorithmic parameters. Small variations are likely due to measurement noise or system activity (e.g. OS interruptions background processes).

During signing, the CPU load is notably higher, approaching the full use of a single core. For smaller files (1 MB, 100 MB), average utilization is ~22–24% (Dilithium-2/3 at ~22–23%, Dilithium-5 slightly higher at ~22–25%). For larger files (500 MB, 1000 MB), utilization rises to ~29–30% for all variants, indicating continuous workload on one core. On a quad-core system, ~25% corresponds to one fully loaded core; ~30% suggests hyper-threading or occasional task switching between cores.

A similar trend is seen in verification: ~15–19% at 1 MB, increasing to ~28–29% at 1000 MB. For 100 MB, Dilithium-5 shows lower utilization (23.19%) than Dilithium-2 (29.23%), but this difference diminishes at higher sizes, converging near ~29%. These minor deviations may stem from implementation differences, internal optimizations, or measurement artifacts (e.g. thread management). Overall, the results confirm that file size drives higher CPU load, stabilizing near the saturation point of a single core.

The analysis indicates that cryptographic key pair generation consumes a relatively small portion of CPU resources and is effectively independent of input file size, as this phase relies on algorithmic parameters and internally generated values rather than message processing.

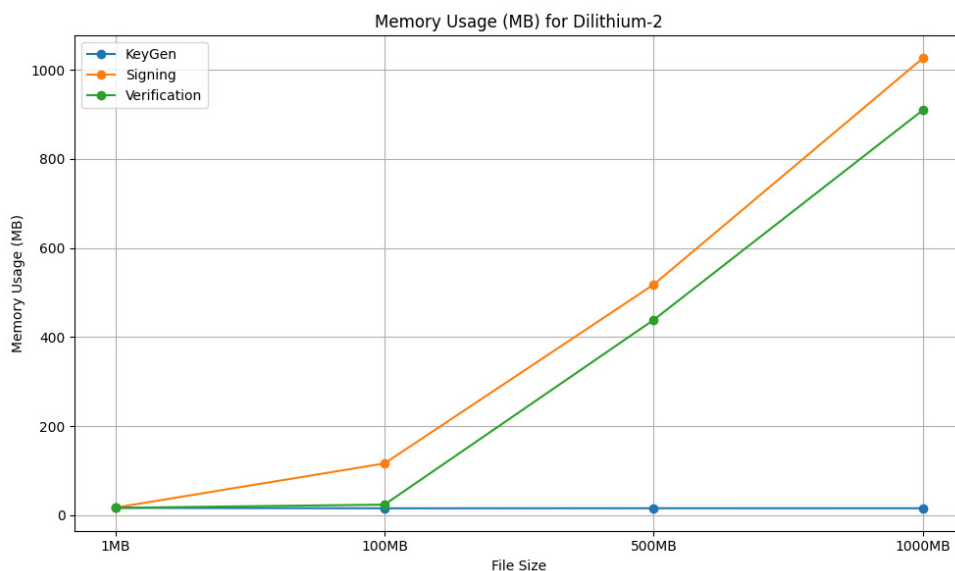
In contrast, signing and verification operations, which require reading and hashing variable-size data, progressively increase CPU load with file size. This increase is, however, limited; CPU utilization stabilizes at a saturation point due to architectural constraints, specifically the processing capacity of a single core.

For large files, all three CRYSTALS-Dilithium variants (Dilithium-2, 3, and 5) reach similar maximum CPU utilization (~30%), indicating that the bottleneck is identical across variants: the single-core processing speed for the given data volume, independent of cryptographic parameters.

2.1.3 Analysis of the Impact of File Size on Memory (RAM – Random Access Memory) during Key Pair Generation, Signing, and Verification

Memory (RAM) usage in cryptographic operations also increases with input file size, especially during signing and verification, as larger files require bigger buffers and more data in active memory.

Figure 3:
File Size vs.
Memory (RAM)
Utilization in
CRYSTALS-
Dilithium 2



During key generation, memory usage is very low and roughly constant (~15–16 MB), independent of file size, since the process involves fixed structures (small vectors and matrices) defined by algorithm parameters, rather than the message itself. For example, with a 1 MB file, memory usage is ~16.6 MB, while with 100 MB it is ~15.7 MB, variations attributable to OS memory allocation, caching, or fragmentation, not functional changes in the algorithm.

In signing, memory usage grows almost linearly with file size. For a 1 MB file, ~17–18 MB is needed, while for 100 MB it rises to ~116 MB, reflecting the loading of the entire file plus buffers and cryptographic structures (~16 MB overhead). For larger files:

- 500 MB: Dilithium-2 ~518 MB, Dilithium-5 ~531 MB
- 1000 MB: all variants ~1.02 GB RAM

This indicates that the implementation loads the full file into memory before or during signing, with additional space for buffers, cryptographic structures, and internal algorithm functions. The increase roughly matches the 1:1 ratio of file size plus a fixed base overhead.

During verification, memory usage also grows with file size, with differences between the security levels.

- For a 100 MB file: Dilithium-2: ~23.9 MB, Dilithium-3: ~64.5 MB, Dilithium-5: ~114.7 MB

These differences suggest not all variants load the full file into memory, or manage internal buffers and caching differently. Dilithium-2 may process the message chunk-wise, while Dilithium-5 likely pre-allocates more memory or loads the entire file upfront.

For larger files, the numbers approach full file load scenarios:

- 500 MB: Dilithium-2 ~438.6 MB, Dilithium-3 ~489.4 MB, Dilithium-5 ~541.2 MB
- 1000 MB: Dilithium-2 ~909 MB, Dilithium-3 ~968 MB, Dilithium-5 ~1049 MB

These values are approximately equal to the file size, with a small, consistent overhead increasing with security level, likely due to the larger public keys, signatures, and internal parameters in Dilithium-5.

In conclusion, verification, like signing, requires memory roughly equal to message size, with possible implementation optimizations at lower security levels (e.g. Dilithium-2 using smaller working buffers or streamed processing). In practice, signing or verifying very large files is memory-intensive and requires sufficient RAM – at least equal to the file size, plus additional space for cryptographic logic.

This trend indicates that in practical applications with large documents, memory usage optimization strategies become necessary. A recommended approach is, for example, to sign a precomputed hash of the document instead of the entire file, an approach which will be further discussed in this section.

2.2 Analysis of the Results (Different Key Size)

The CRYSTALS-Dilithium algorithm is defined through three parameter sets: Dilithium-2, Dilithium-3, and Dilithium-5, designed to provide varying levels of cryptographic security, roughly equivalent to 128-bit, 192-bit, and 256-bit classical security, respectively. This section analyses how the choice of parameter set, that is, key size, signature size, and associated algorithmic configurations, affects the following performance aspects:

- Execution time for key generation, signing, and verification (Section 2.2.1)
- CPU load during these operations (Section 2.2.2)
- Memory (RAM) consumption in all three phases (Section 2.2.3)

The analysis uses results from the same measurement series presented in the previous sections, but focuses on comparing parameter levels (Dilithium-2, 3, and 5) under identical conditions, i.e. a fixed file size. This approach enables a better understanding

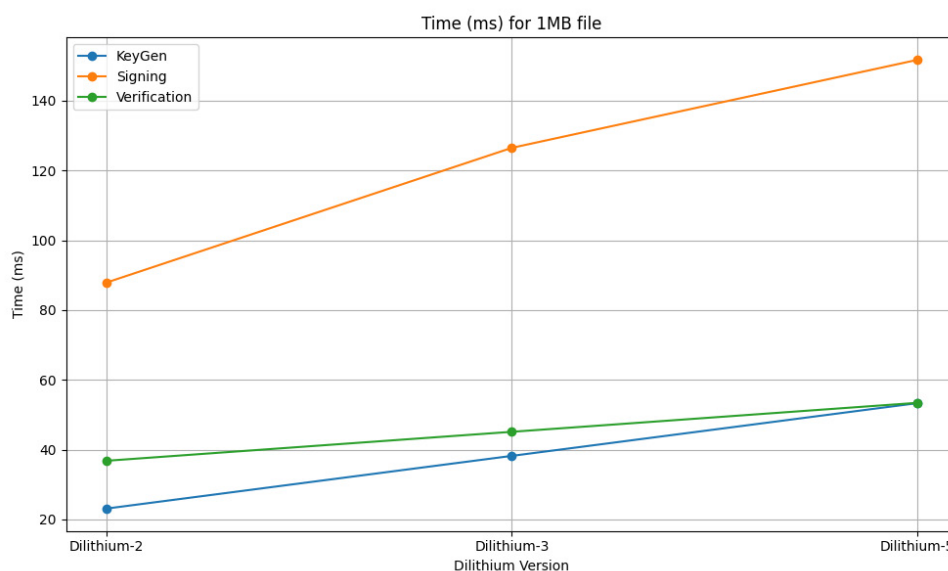
of the technical trade-offs between security and performance, and provides guidance for parameter selection depending on security requirements and system resources.

2.2.1 Analysis of the Impact of CRYSTALS-Dilithium Key Size (2, 3, and 5) on Key Pair Generation Time, Signing and Verification Time

As shown in Table 2 execution time differences between CRYSTALS-Dilithium parameter sets (Dilithium-2, 3, 5) are most noticeable for small messages, where cryptographic computation dominates.

For a 1 MB file:

- Signing: Dilithium-2 ~0.088 s, Dilithium-3 ~0.127 s, Dilithium-5 ~0.152 s (~70% slower than Dilithium-2);
- Verification: Dilithium-2 ~0.037 s, Dilithium-5 ~0.053 s (~45% slower), with Dilithium-3 between the two;
- Key generation: Dilithium-2 0.0231 s, Dilithium-5 0.0533 s (~2× slower).



These differences result from larger matrices, polynomials, and vectors in higher-security variants, increasing arithmetic complexity in key generation, signing, and verification.

With larger files, relative differences decrease:

- 100 MB signing: Dilithium-5 0.8978 s (~8.5% slower than Dilithium-2 0.8276s)

- 500 MB signing: 3.9298 s vs. 3.8791 s ($\sim 1.3\%$ difference)
- 1000 MB signing: 7.67 s vs. 7.62 s ($< 1\%$ difference)
- Verification: difference between D5 and D2 drops from $\sim 2.6\%$ (100 MB) to $< 1\%$ (500 MB) and $\sim 0.3\%$ (1000 MB)

This confirms that for large messages, total time is dominated by hashing and I/O rather than cryptographic complexity. For small messages, parameter set choice significantly affects performance, due to the higher fixed cryptographic overhead in Dilithium-5.

Key generation differences follow the same trend: Dilithium-3 is $\sim 1.5\times$ slower than Dilithium-2, and Dilithium-5 $\sim 2\times$ slower. For a 1000 MB file, the key generation times are 0.021 s (D2), 0.031 s (D3), and 0.046 s (D5). Absolute times remain below 0.05 s, keeping key generation fast and efficient across all security levels.

2.2.2 Analysis of the impact of CRYSTALS-Dilithium key size (2, 3, and 5) on CPU load during key generation, signing, and verification

The differences in CPU load between Dilithium-2, 3, and 5 are most noticeable in short-term operations, similar to execution time. During key generation, the larger algorithm (Dilithium-5) uses more CPU power (up to $\sim 17\text{--}18\%$ on average) compared to Dilithium-2 ($\sim 10\text{--}12\%$) because it carries out more operations. The same applies to the verification of small messages: for 1 MB, Dilithium-5 uses $\sim 19\%$ CPU, compared to $\sim 15.6\%$ for Dilithium-2. This indicates that the more complex algorithm consumes more CPU cycles for processing, so even over short periods it registers a higher load.

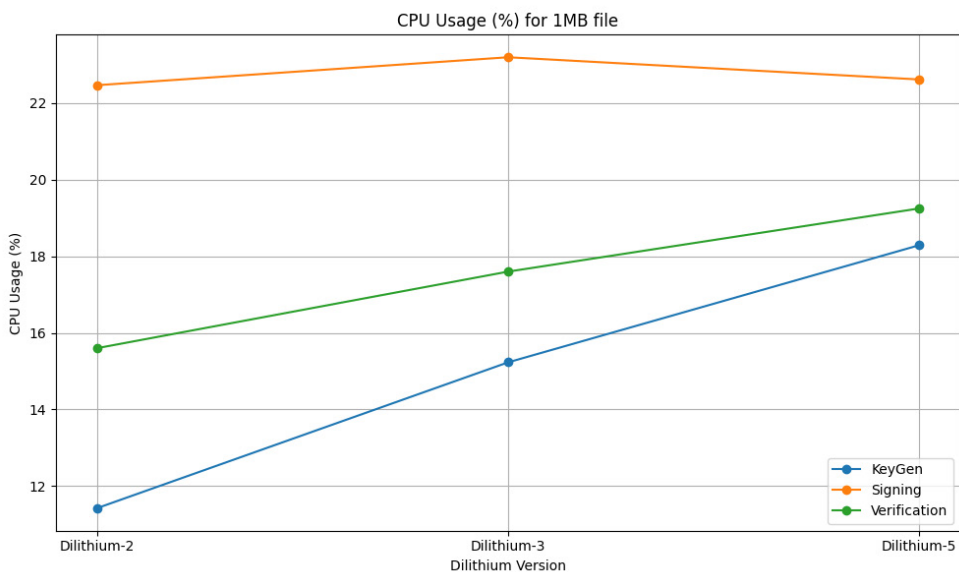
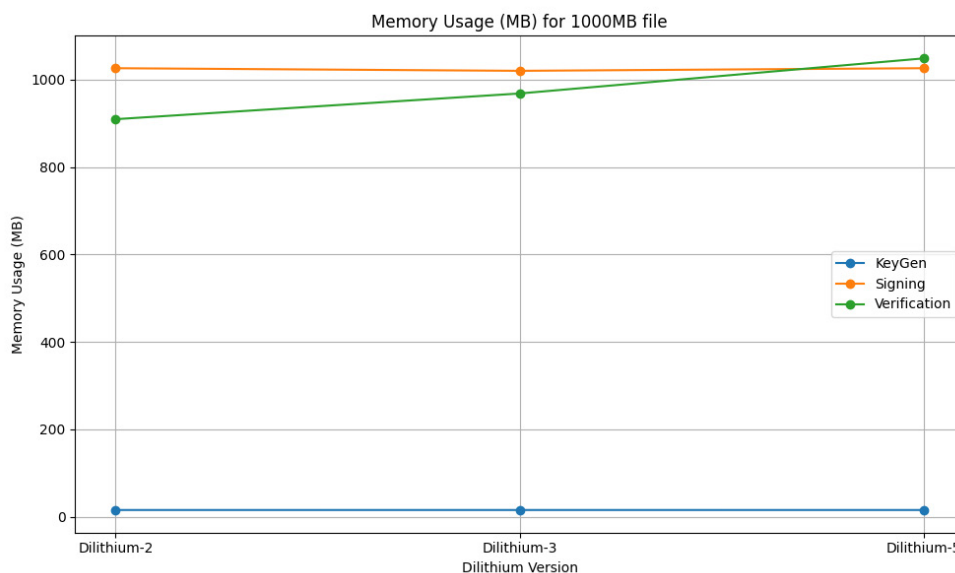


Figure 5:
Security Level
(CRYSTALS-
Dilithium 2, 3, 5)
vs. CPU Utilization
for 1 MB File

However, for longer tasks, such as signing or verifying large files, the average CPU load levels out. As seen in Table 2 for 500 MB and 1000 MB, all variants use ~28–30% CPU, with no significant difference. For example, signing a 1000 MB file uses ~29.5% CPU across all three levels. This means that all the variants ultimately limit CPU usage to a similar degree (as discussed, roughly one core at maximum). The differences observed with small inputs disappear here because the execution is long, and the CPU is continuously engaged; whether executing Dilithium-2 or 5, it always has enough work (from data hashing or the signing logic itself) to remain near 100% utilization of one core. Therefore, for long operations, the choice of algorithm does not affect the average CPU load, only the operation duration; but as we have seen, the duration is almost identical when data processing dominates.

2.2.3 Analysis of the impact of the CRYSTALS-Dilithium key size (2, 3, and 5) on memory usage during key generation, signing, and verification

Figure 6:
Security Level
(CRYSTALS-
Dilithium 2, 3,
5) vs. Memory
Utilization for 1
MB File



The key size, i.e. the choice of parameter set, also affects the algorithm's memory requirements, although its impact is significantly smaller than that of file size. As shown in Table 2 during key generation, all variants use roughly the same memory, around 15–16 MB, with no notable differences between Dilithium-2, 3, and 5, since even the largest structures, such as secret keys, occupy only a few kilobytes. Memory usage here is more influenced by the runtime environment, allocation methods, and system configuration than by the algorithm itself.

During signing, higher security levels consume slightly more memory, which becomes more noticeable for larger files. For example, when signing a 500 MB file, Dilithium-5 used ~531 MB versus ~518 MB for Dilithium-2, a difference of ~2.5%. For smaller files, such as 100 MB, the difference is minimal: 116.5 MB for

Dilithium-5 versus 116.3 MB for Dilithium-2. This indicates that the extra memory required by larger cryptographic structures, such as public keys and signatures, represents a relatively constant overhead which becomes proportionally smaller for larger inputs.

During verification, differences in memory usage across parameter sets are more pronounced, particularly for medium-sized files. For a 100 MB message, Dilithium-5 requires ~114.7 MB RAM, while Dilithium-2 uses only ~23.9 MB, likely due to different buffer management strategies in the implementation. For larger files, the trend continues: for 1000 MB, Dilithium-5 uses ~1048 MB, Dilithium-2 ~909 MB (~15% difference), likely caused by larger internal structures at higher security levels, such as expanded matrix vectors. Nevertheless, all the variants generally use memory approximately equal to the file size, meaning that for memory-sensitive applications, choosing Dilithium-2 over Dilithium-5 will not significantly reduce the requirements for the same large files. The differences become more relevant with smaller files: for 1 MB messages, all three sets use ~16–18 MB RAM, showing negligible deviations. Overall, the memory overhead of Dilithium-5 appears mainly for large data, with a worst-case increase of up to ~100 MB compared to Dilithium-2.

3 ANALYSIS OF THE IMPACT OF THE CRYSTALS-DILITHIUM KEY SIZE (2, 3, AND 5) ON KEY-PAIR GENERATION TIME, SIGNING AND VERIFICATION TIME, CPU LOAD, AND MEMORY USAGE WHEN USING A PRECOMPUTED HASH OF THE MESSAGE/FILE

Table 3:
Test results
(precomputed
hash of 500MB
file size)

Algorithm	Key generation (s)	Signing (hash) (s)	Verification (hash) (s)	Total time (s)
Dilithium-2	0.0207	0.1091	0.0209	7.4836
Dilithium-3	0.0296	0.1251	0.0274	7.6476
Dilithium-5	0.0431	0.1741	0.0395	8.8096

Focusing on the signing and verification of a precomputed hash (Table 3 shows substantial improvements compared to signing the entire file. For instance, while signing a 500 MB file directly takes ~3.9 seconds and verification ~3.8 seconds (Table 2, signing a 32-byte SHA-256 hash reduces these times to approximately 0.11 s (Dilithium-2), 0.125 s (Dilithium-3), and 0.174 s (Dilithium-5), with verification completed in only 0.02–0.04 s.

This efficiency arises because the hash-then-sign approach operates on a fixed, small data size, independent of the original file. Consequently, the CPU load drops to ~18–20% for signing, compared to 22–30% when signing the full file, and under 10% for verification. Memory usage also decreases, requiring only ~14–15 MB RAM instead of hundreds of megabytes, since the full file does not need to be loaded for cryptographic operations.

Considering the full workflow – key generation, signing, and verification, averaged over 20 iterations – the total time is ~7.5 s (Dilithium-2), 7.65 s (Dilithium-3), and 8.81 s (Dilithium-5), approximately matching the sum of direct signing and verification times (~7.69 s for Dilithium-2). While the absolute difference between Dilithium-5 and Dilithium-2 exceeds 1 second, it may become relevant in large-scale or real-time signing contexts.

The hash-then-sign approach offers clear advantages in resource-constrained environments, such as HSMs (Hardware Security Module), where only the small fixed-size hash (32 bytes) is transmitted, and streaming processing can handle large files without full in-memory loading. Security is maintained, as a valid hash signature implies a valid signature on the original message.

Performance measurements confirm substantial gains: CPU usage drops to ~18% for signing and ~5% for verification, while memory use remains at ~15 MB. This enables the secure signing of large documents even on devices with limited resources, provided the hash is precomputed.

4 PRACTICAL IMPLICATIONS AND RECOMMENDATIONS ON CPU LOAD

CPU utilization is a key factor when choosing a variant, depending on device type, context, and security-efficiency priorities:

- Resource-constrained edge devices: For IoT, embedded systems, or smart sensors primarily carrying out signature verification, Dilithium-2 is optimal due to its low CPU load (~4.3%) and reduced latency, enhancing battery life and responsiveness.
- High-throughput servers: For servers handling thousands of verification requests, Dilithium-3 (~6.4% CPU) offers a balance of AES-192 equivalent security and efficiency. Dilithium-5 (~8.6%) is suitable for critical applications (military, governmental) requiring maximum security.
- Signing with delayed verification: When signing dominates (e.g. archiving, blockchain, timestamping) and verification is infrequent, Dilithium-5 is recommended despite its higher CPU usage (~20% during signing), ensuring long-term security.
- Batch or pre-hashing optimization: Since CPU load is dominated by message hashing, partitioned or pre-computed hashing (SHAKE256) using additional cores or GPU can relieve the main processor, improving throughput and reducing latency.

5 PRACTICAL IMPLICATIONS AND RECOMMENDATIONS ON MEMORY USAGE

The implementation of CRYSTALS-Dilithium also requires careful assessment of memory, especially for resource-constrained devices or large-scale parallel processing:

- Memory-limited devices (IoT, embedded systems):
 - Dilithium-2 is optimal for devices with <32 MB RAM.
 - Dilithium-3 and Dilithium-5 require more memory, which is feasible on devices with 64–128 MB RAM, such as advanced embedded or edge platforms.
- Server and cloud environments:
 - Dilithium-3 balances AES-192 equivalent security with moderate memory use (~16.5 MB per process).
 - Dilithium-5 adds only ~0.4 MB RAM per parallel process, negligible in modern server setups, offering AES-256 level security.
- Massive parallel processing:
 - Per-task memory usage is ~15–17 MB, enabling up to ~100 concurrent processes under 2 GB RAM, feasible for high-throughput server or cloud infrastructures.

6 RECOMMENDATIONS FOR OPTIMAL IMPLEMENTATION AND PERFORMANCE

- Utilize optimized libraries with vectorized instructions (AVX2/AVX-512), which can accelerate signing by 3–5 times according to reference benchmarks.
- Sign the hash of the message for large documents, reducing memory consumption and splitting the process into two lighter phases.
- Employ parallel hashing where possible, or parallel signing of multiple independent messages to leverage multiple cores—but with caution, as large files may saturate memory resources.
- Follow the authors' recommendations, such as using a hybrid signing model (combining Dilithium with a classical algorithm) when additional security is required during the transition period.
- By applying these measures, the transition to post-quantum digital signatures such as CRYSTALS-Dilithium can be achieved without significant performance compromise, while substantially enhancing the long-term security of digital communications and data.

Conclusion This study offers a thorough performance analysis of the post-quantum digital signature algorithm CRYSTALS-Dilithium, looking closely at its efficiency, scalability, and how well it can be deployed in real-world applications.

Key findings:

- **Efficiency and scalability:** Dilithium truly excels when it comes to performance. Key generation and signing operations take just milliseconds, and verification is even faster! In addition, its ability to scale linearly with message size means you can rely on consistent performance across a wide range of applications.
- **Security levels:** The three parameter sets – Dilithium-2, Dilithium-3, and Dilithium-5 – provide increasing levels of cryptographic strength. Although the higher levels come with slight performance trade-offs, they still work well for real-world applications, enabling you to customize security based on your operational needs.
- **Large data handling:** Pre-hashing can really boost performance when dealing with large messages, cutting down the signing time by as much as five times for data sets that are in the hundreds of megabytes.
- **Hardware utilization:** Dilithium is highly optimized for parallelization and hardware acceleration. You can expect almost linear speed improvements on multi-core systems, and there is even more potential for boosts with vectorized instructions and specialized accelerators.

Strengths and Limitations:

Dilithium derives its strength from its NIST standardization (FIPS 204), robust security built on lattice-based cryptography, and its ability to maintain efficiency across various platforms. These features position it as a top contender for post-quantum adoption. That said, there are still some limitations to overcome: the signature sizes tend to be larger compared to traditional schemes, and secure implementation needs to be mindful of side-channel resistance and memory limitations, especially in environments with limited resources.

Further research should focus on:

- **Deployment in real-world systems,** including TLS, VPNs, blockchain, and software signing.
- **Testing military communication systems.** It is essential to focus on long-term confidentiality, ensuring interoperability with NATO standards, and maintaining resilience in challenging environments.

- Hardware-level optimizations, such as FPGA and GPU acceleration, which significantly ramp up scalability for high-stakes defence and national security applications.

Overall, CRYSTALS-Dilithium is already laying down a solid and secure groundwork for post-quantum digital signatures. With ongoing tweaks and real-world testing, it is set to become a key player in ensuring secure communication for both civilian and military applications.

References

1. Chen, Y. N., Gao, Y., Liu, Y., Zhou, Y., 2023, December. Low latency implementations of CRYSTALS-Dilithium for cloud applications. In *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pp 679–686. IEEE.
2. Bos, J. W., Renes, J., Sprenkels, A., 2022. Dilithium for memory constrained devices. *IACR Cryptology ePrint Archive*, Report 2022/1234.
3. Demir, A., Kaya, M., Özdemir, S., 2025. Performance analysis and industry deployment of post-quantum cryptography algorithms. *Journal of Cryptographic Engineering* (forthcoming).
4. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D., 2018a. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp 238–268.
5. Ducas, L., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D., 2018b. CRYSTALS-Dilithium: Digital signatures from module lattices. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2018(1), pp 238–268.
6. Freemindtronic, 2024. *Quantum threats to encryption: RSA, AES & ECC defense*. Freemindtronic SAS.
7. Gitonga, C. K., 2025. The impact of quantum computing on cryptographic systems. *Journal of Information Security Research*, 14(2), pp 45–59.
8. Land, G., Sasdrich, P., Güneysu, T., 2021, November. A hard crystal – implementing Dilithium on reconfigurable hardware. In *International Conference on Smart Card Research and Advanced Applications*, pp 210–230. Cham: Springer International Publishing.
9. Li, X., Lu, J., Liu, D., Li, A., Yang, S., Huang, T., 2023. A high speed post-quantum crypto-processor for CRYSTALS-Dilithium. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 71(1), pp 435–439.
10. Liu, H., Zhang, Y., Chen, W., 2025. Efficient side-channel attack on Dilithium. *IACR Cryptology ePrint Archive*, Report 2025/567.
11. Lyubashevsky, V., Ducas, L., Kiltz, E., Lepoint, T., Schwabe, P., Seiler, G., Stehlé, D., Bai, S., 2020. CRYSTALS-Dilithium. *Algorithm Specifications and Supporting Documentation*, 2.
12. National Institute of Standards and Technology (NIST), 2024. *FIPS 204: Module-Lattice-Based Digital Signature Standard*. U.S. Department of Commerce.
13. NATO, 2024. *NATO Interoperability Standards and Profiles (NISP)*. NATO Communications and Information Agency.

14. NATO, 2025. *NATO Cryptographic Key Management Conformance Tool*. NATO Standardization Office.
15. Türe, N. D., Cenk, M., 2024, June. Efficient batch post-quantum signatures with CRYSTALS-Dilithium. In *International Workshop on the Arithmetic of Finite Fields*, pp 237–257. Cham: Springer Nature Switzerland.
16. Vidaković, M., Miličević, K., 2023. Performance and applicability of post-quantum digital signature algorithms in resource-constrained environments. *International Journal of Information Security*, 22(4), pp 321–338.
17. Zhou, Z., He, D., Liu, Z., Luo, M., Choo, K. K. R., 2021. A software/hardware co-design of CRYSTALS-Dilithium signature scheme. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 14(2), pp 1–21.

Email: kire.jakimoski@ugd.edu.mk

Email: palitov_b@yahoo.com

Dr. Kire Jakimoski je profesor in strokovnjak na področjih telekomunikacij, računalniških omrežij in kibernetske varnosti. Njegova kariera traja že več kot dve desetletji, v katerih združuje akademsko raziskovanje s praktičnimi aplikacijami v javnem in zasebnem sektorju. Bil je na številnih vodstvenih položajih v akademskem svetu, med drugim je bil dekan Fakultete za informatiko, in je bil priznan za svoje prispevke tako na področju poučevanja kot raziskovanja. Je avtor in soavtor več kot 55 publikacij, vključno z raziskovalnimi članki, knjigami in zborniki konferenc.

Kire Jakimoski, PhD is a professor and expert in the fields of telecommunications, computer networks, and cybersecurity. His career spans over two decades, blending academic research with practical applications in both the public and private sectors. He has held numerous leadership roles in academia, including Dean of the Faculty of Informatics, and has been recognized for his contributions to both teaching and research. He has authored and co-authored over 55 publications, including research papers, books, and conference proceedings.

Email: kire.jakimoski@ugd.edu.mk

* Prispevki, objavljeni v Sodobnih vojaških izzivih, niso uradno stališče Slovenske vojske niti organov, iz katerih so avtorji prispevkov.

* Articles published in the Contemporary Military Challenges do not reflect the official viewpoint of the Slovenian Armed Forces nor the bodies in which the authors of articles are employed.

Stotnik Blashko Palitov je podiplomski študent na Vojaški akademiji General Mihailo Apostolski v Skopju, kjer se ukvarja z raziskovanjem postkvantnih algoritmov Crystals-Dilithium in Crystals-Kyber. Služi kot častnik v Armadi Republike Severne Makedonije, in sicer kot načelnik S-3 v bataljonu za zveze. Prej je služil kot častnik za informacijske sisteme in informacijsko varnost. Njegova strokovna zanimanja vključujejo kibernetško varnost, digitalno forenziko in kriptografijo.

Captain Blashko Palitov is a postgraduate student at the »General Mihailo Apostolski« Military Academy – Skopje, conducting research focused on the post-quantum algorithms CRYSTALS-Dilithium and CRYSTALS-Kyber. He serves as an officer in the Army of the Republic of Macedonia, currently holding the position of Chief of S-3 in the Signal Battalion. Previously, he served as an officer for information systems and information security. His professional interests include cybersecurity, digital forensics, and cryptography.

Email: palitov_b@yahoo.com

* Prispevki, objavljeni v Sodobnih vojaških izzivih, niso uradno stališče Slovenske vojske niti organov, iz katerih so avtorji prispevkov.

* Articles published in the Contemporary Military Challenges do not reflect the official viewpoint of the Slovenian Armed Forces nor the bodies in which the authors of articles are employed.