Custo da Agilidade Criptográfica: Desempenho de Algoritmos Clássicos e Pós-quânticos no OpenSSL

Mariana B. M. Sabino[1], Sabrina O. Sousa[1], Cauan M. Sousa[1], Rafael W. Cavalcanti[1], Matheus V. P. Santos[2], Edmar C. Gurjão[1], Leocarlos B. S. Lima[1]

[1]Unidade Acadêmica de Engenharia Elétrica
Universidade Federal de Campina Grande (UFCG)
58429-900, Campina Grande, PB, Brasil

[2]Centro de Estudos e Sistemas Avançados do Recife (CESAR)
Centro Integrado de Segurança em Sistemas Avançados (CISSA)
Recife, PE, Brasil

{mariana.barros.de.moraes.sabino, sabrina.de.oliveira.sousa}@ccc.ufcg.edu.br,
{cauan.sousa, rafael.cavalcanti}@ee.ufcg.edu.br,
mvps@cesar.org.br, {ecg, leocarlos}@dee.ufcg.edu.br

Abstract. *The quantum threat to asymmetric cryptography drives the adoption of PQC. This work compares ECDSA and ECDH with ML-KEM, ML-DSA, and SLH-DSA in signature and key establishment operations. ML-KEM showed throughput 2.37×, 40.19×, and 72.7× higher than the corresponding classical key establishment operations at the 128, 192, and 256-bit levels, respectively. In signature, ML-DSA-44 was 27.9× slower than ECDSA P-256, while ML-DSA outperformed ECDSA in verification at the 192 and 256-bit levels. SLH-DSA showed the lowest signature throughput, between 1.3 and 2.7 signatures/s. These results support the evaluation of migration to PQC in network infrastructures.*

Resumo. *A ameaça quântica à criptografia assimétrica impulsiona a adoção de PQC. Este trabalho compara ECDSA e ECDH com ML-KEM, ML-DSA e SLH-DSA em operações de assinatura e estabelecimento de chaves. O ML-KEM apresentou throughput 2,37×, 40,19× e 72,7× superior às operações clássicas de estabelecimento de chaves correspondentes nos níveis de 128, 192 e 256 bits, respectivamente. Na assinatura, o ML-DSA-44 foi 27,9× mais lento que o ECDSA P-256, enquanto o ML-DSA superou o ECDSA em verificação nos níveis de 192 e 256 bits. O SLH-DSA apresentou o menor throughput de assinatura, entre 1,3 e 2,7 assinaturas/s. Esses resultados apoiam a avaliação da migração para PQC em infraestruturas de rede.*

1. Introdução

O avanço da computação quântica ameaça algoritmos assimétricos amplamente utilizados, como RSA e ECC (*Elliptic Curve Cryptography*). O algoritmo de Shor [Shor 1994], executado em computadores quânticos de larga escala, compromete diretamente esses sistemas.

Em resposta, o *National Institute of Standards and Technology* (NIST) concluiu em 2024 o processo de padronização de algoritmos de Criptografia Pós-Quântica (PQC), publicando três padrões: FIPS 203 (ML-KEM) [NIST 2024a], FIPS 204 (ML-DSA) [NIST 2024b] e FIPS 205 (SLH-DSA) [NIST 2024c]. Esses algoritmos são baseados em problemas matemáticos considerados resistentes a ataques quânticos conhecidos, como os problemas sobre reticulados (*lattices*) e funções de *hash*.

Embora a segurança teórica desses algoritmos esteja consolidada, aspectos práticos como custo computacional, desempenho e tamanho de parâmetros ainda demandam investigação. Além disso, a adoção de KEMs (*Key Encapsulation Mechanism*) introduz mudanças estruturais no estabelecimento de chaves em relação aos modelos clássicos KAS (*Key Agreement Scheme*).

Este artigo apresenta uma comparação empírica entre algoritmos clássicos e pós-quânticos padronizados pelo NIST, considerando:

1. **Parâmetros estáticos** – tamanhos de chave pública, chave privada e assinatura/texto cifrado por categoria de segurança NIST equivalente;
2. **Desempenho operacional** – *throughput* de assinatura, verificação e das operações utilizadas no estabelecimento de segredos em hardware convencional.

As contribuições incluem a análise comparativa por categorias de segurança do NIST (1, 3 e 5), a distinção entre os modelos KAS e KEM conforme o NIST SP 800-227 e a avaliação experimental de duas configurações híbridas clássico–PQC [Alagic et al. 2025].

2. Fundamentação Teórica

A transição para a criptografia pós-quântica introduz mudanças significativas nos fundamentos, modelos operacionais e requisitos da criptografia assimétrica. Nesse contexto, esta seção apresenta os conceitos necessários para compreender os algoritmos e mecanismos analisados, incluindo as categorias de segurança do NIST, as limitações dos sistemas clássicos frente à computação quântica e as diferenças entre mecanismos de acordo de chaves (KAS) e encapsulamento de chaves (KEM).

2.1. Categorias de Segurança NIST

São definidas cinco categorias de segurança para PQC, comparando a quebra de um algoritmo PQC com a dificuldade equivalente de quebrar um algoritmo clássico [NIST 2024]. A Tabela 1 resume esse mapeamento.

Tabela 1. Categorias de segurança do NIST e conjuntos de parâmetros avaliados.

Categoria	Referência de segurança	Conjuntos PQC
1	Busca exaustiva em AES-128	ML-KEM-512, ML-DSA-44, SLH-DSA-128s
2	Colisão em SHA-256	–
3	Busca exaustiva em AES-192	ML-KEM-768, ML-DSA-65, SLH-DSA-192s
4	Colisão em SHA-384	–
5	Busca exaustiva em AES-256	ML-KEM-1024, ML-DSA-87, SLH-DSA-256s

As categorias permitem comparar conjuntos de parâmetros pós-quânticos segundo referências comuns de custo de ataque. Neste trabalho, são considerados

principalmente os conjuntos correspondentes às categorias 1, 3 e 5, por serem os níveis disponibilizados pelos algoritmos padronizados e avaliados experimentalmente.

2.2. Algoritmos Clássicos

Historicamente, o algoritmo RSA dominou a criptografia de chave pública. Contudo, a *Elliptic Curve Cryptography* (ECC) consolidou-se como a sucessora preferencial por oferecer segurança equivalente com chaves menores: cerca de 256 bits em ECC correspondem a 3.072 bits em RSA para aproximadamente 128 bits de segurança.

Entretanto, tanto RSA quanto ECC tornam-se vulneráveis diante da computação quântica. O algoritmo de Shor [Shor 1994] resolve problemas de fatoração e logaritmo discreto em tempo polinomial, comprometendo diretamente a segurança dos sistemas que utilizam criptografia clássica.

2.3. KAS versus KEM

ECDH é um esquema de acordo de chaves no qual ambas as partes contribuem com valores públicos para derivar um segredo compartilhado. No ML-KEM, o destinatário gera um par de chaves, o encapsulador produz um texto cifrado e um segredo compartilhado, e o destinatário recupera esse segredo por decapsulamento. Assim, embora ambos sejam utilizados no estabelecimento de chaves, possuem fluxos e operações distintos, exigindo adaptações na codificação das mensagens e na integração com protocolos como TLS e SSH [Alagic et al. 2025].

3. Trabalhos Relacionados

O projeto *Open Quantum Safe* (OQS) fornece bibliotecas e integrações experimentais para algoritmos PQC no ecossistema OpenSSL, incluindo configurações híbridas [Open Quantum Safe Project 2025]. Em plataformas restritas, o *framework* pqm4 disponibiliza implementações e *benchmarks* em microcontroladores ARM Cortex-M4 [Kannwischer et al. 2019].

Este trabalho complementa essas iniciativas ao comparar, em um único ambiente OpenSSL, algoritmos clássicos e pós-quânticos em níveis aproximadamente equivalentes de segurança, considerando desempenho, tamanho dos parâmetros, diferenças entre KAS e KEM e configurações híbridas.

4. Metodologia

Esta seção apresenta a abordagem metodológica utilizada para avaliar empiricamente algoritmos criptográficos clássicos e padrões pós-quânticos do NIST. Os experimentos têm como objetivo quantificar o custo computacional da agilidade criptográfica em um ecossistema de uso geral amplamente adotado na indústria. Para isso, são detalhados o ambiente experimental, os algoritmos selecionados e as métricas de desempenho utilizadas na coleta e na análise dos resultados.

4.1. Configuração Experimental

A configuração experimental foi realizada em um computador equipado com processador AMD Ryzen 5 PRO 3400G (4 núcleos, 8 *threads*, frequência base de 3,7

GHz e suporte às extensões AVX2), 16 GB de memória RAM DDR4 e armazenamento em disco rígido de 1 TB.

Os benchmarks foram executados utilizando o OpenSSL 3.5.5 no sistema operacional Ubuntu 22.04.5 LTS (Jammy Jellyfish), kernel Linux 6.8.0-124-generic, em arquitetura x86-64. O OpenSSL foi compilado com o compilador GCC utilizando as opções `-fPIC -pthread -m64 -Wa,-noexecstack -Wall -O3`, além das definições `OPENSSL_USE_NODELETE`, `OPENSSL_PIC`, `OPENSSL_BUILDING_OPENSSL` e `DNDEBUG`, habilitando o uso das instruções vetoriais AVX2 disponíveis no processador.

Os algoritmos padronizados ML-KEM, ML-DSA e SLH-DSA foram avaliados utilizando exclusivamente o *OpenSSL Default Provider* (versão 3.5.5). As configurações híbridas X25519MLKEM768 e SecP256r1MLKEM768 foram avaliadas separadamente por meio de scripts automatizados utilizando comandos da interface de linha de comando do OpenSSL, uma vez que essas combinações não são avaliadas diretamente pelo utilitário `openssl speed`.

Os algoritmos suportados pelo `openssl speed` foram avaliados diretamente por esse utilitário. Para algoritmos e mecanismos não suportados diretamente, foram implementados benchmarks específicos utilizando os comandos `openssl genpkey`, `openssl pkeyutl` e `openssl kdf`, executando repetidamente as operações criptográficas correspondentes e calculando o throughput em operações por segundo.

Cada configuração foi executada em 30 rodadas independentes, utilizando um único processo, com duração aproximada de 10 segundos por rodada. Não foi realizada etapa específica de aquecimento (*warm-up*). Para cada métrica foram calculadas a média aritmética e o desvio-padrão amostral, sendo as barras de erro apresentadas nas figuras correspondentes iguais a um desvio-padrão. Durante todos os experimentos, o sistema permaneceu sem aplicações concorrentes intensivas, utilizando o governador de frequência da CPU `schedutil`.

4.2. Algoritmos e Métricas Avaliadas

Foram avaliados algoritmos clássicos e pós-quânticos para assinatura digital e estabelecimento de chaves. Para assinatura digital, foram considerados ECDSA com as curvas P-256, P-384 e P-521, além de ML-DSA-44, ML-DSA-65, ML-DSA-87 e das variantes SLH-DSA-SHA2-128s, SLH-DSA-SHA2-192s e SLH-DSA-SHA2-256s, denominadas abreviadamente SLH-DSA-128s, SLH-DSA-192s e SLH-DSA-256s nas figuras e na discussão dos resultados.

Como referências clássicas para o estabelecimento de segredos nos níveis aproximados de 128, 192 e 256 bits, foram utilizados, respectivamente, X25519, secp384r1 e secp521r1. Os mecanismos pós-quânticos avaliados foram ML-KEM-512, ML-KEM-768 e ML-KEM-1024. O OpenSSL 3.5 disponibiliza os grupos híbridos X25519MLKEM768, SecP256r1MLKEM768 e SecP384r1MLKEM1024; entretanto, neste trabalho foram avaliadas experimentalmente apenas as configurações X25519MLKEM768 e SecP256r1MLKEM768.

As métricas foram selecionadas conforme a natureza de cada mecanismo. Para os algoritmos de assinatura, foram medidos os *throughputs* de geração (*Sign*) e verificação (*Verify*) de assinaturas. Para os mecanismos clássicos de acordo de

chaves, foram medidas a geração do par de chaves (*Keygen*) e a derivação do segredo compartilhado. Para ML-KEM, foram medidos *Keygen*, encapsulamento (*Encaps*) e decapsulamento (*Decaps*). Para as configurações híbridas, foi medido o *throughput* do procedimento combinado de estabelecimento do segredo.

Como mecanismos KAS não realizam encapsulamento nem decapsulamento, as operações de ECDH são apresentadas separadamente das métricas de ML-KEM, evitando atribuir aos mecanismos clássicos operações próprias de KEMs.

5. Resultados e Discussão

Esta seção apresenta os resultados experimentais em três frentes: análise de parâmetros estáticos (tamanhos de chaves e assinaturas), avaliação de desempenho das operações criptográficas (assinatura digital e estabelecimento de chaves via KAS e KEM) e, por fim, análise complementar de funções de *hash* e avaliação de abordagens híbridas na transição para PQC.

5.1. Parâmetros Estáticos

A Tabela 2 apresenta uma comparação dos tamanhos, em bytes (B), dos principais parâmetros criptográficos de algoritmos clássicos e pós-quânticos com níveis de segurança aproximadamente equivalentes (Categoria 1 do NIST). Os valores referentes aos algoritmos pós-quânticos seguem as especificações dos padrões FIPS 203 e FIPS 204 [NIST 2024a, NIST 2024b]. A comparação evidencia o aumento no tamanho de chaves, assinaturas e textos cifrados decorrente da adoção de algoritmos pós-quânticos.

Tabela 2. Comparação simplificada de parâmetros criptográficos na Categoria 1 de segurança do NIST.

Algoritmo	Chave Pública	Chave Privada	Saída
ECDSA P-256	64 B	32 B	64 B
ML-DSA-44	1312 B	2560 B	2420 B
X25519	32 B	32 B	32 B
ML-KEM-512	800 B	1632 B	768 B

Observa-se que os algoritmos pós-quânticos apresentam chaves públicas, chaves privadas e assinaturas ou textos cifrados significativamente maiores que seus equivalentes clássicos. Esse aumento pode impactar diretamente protocolos de rede e mecanismos de transporte, elevando o volume de dados transmitidos e afetando cenários com MTU reduzida, dispositivos com recursos limitados ou restrições de largura de banda.

5.2. Desempenho de Assinaturas

A Figura 1 apresenta o *throughput* das operações de assinatura e verificação digital, em operações por segundo, para os algoritmos clássicos e pós-quânticos avaliados.

Os resultados mostram que o ML-DSA apresentou desempenho competitivo nas operações de verificação de assinaturas, superando os algoritmos baseados em

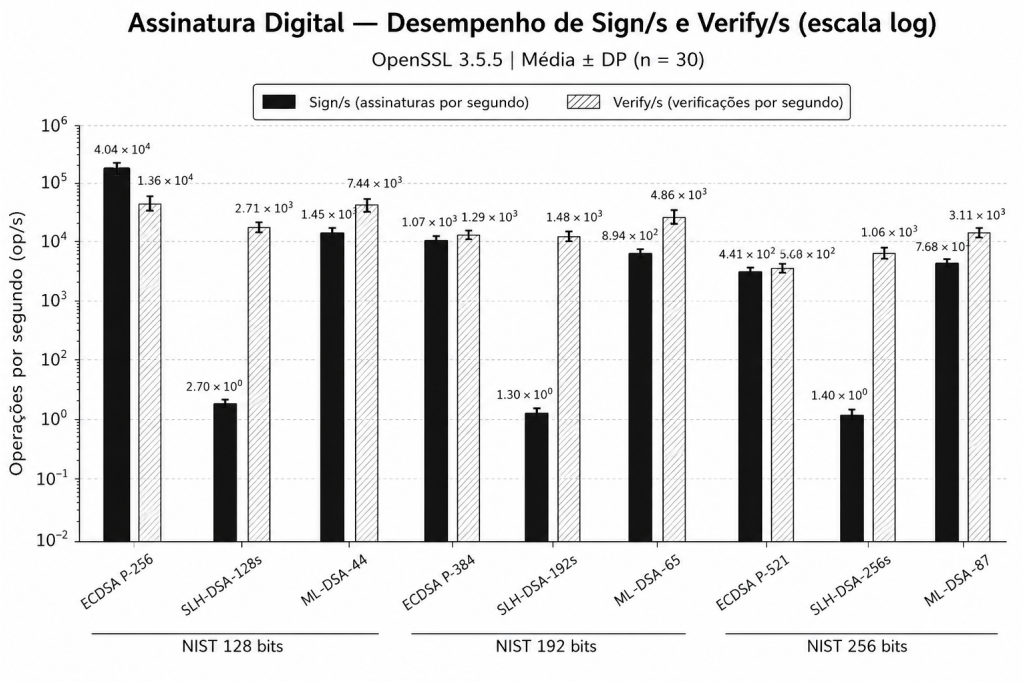

Figura 1. Desempenho das operações de assinatura e verificação para algoritmos clássicos e pós-quânticos.

curvas elípticas nos níveis de segurança de 192 e 256 bits. No nível de 192 bits, o ML-DSA-65 atingiu aproximadamente $4,86 \times 10^3$ verificações por segundo, superando o ECDSA P-384, que alcançou aproximadamente $1,29 \times 10^3$ verificações por segundo. No nível de 256 bits, o ML-DSA-87 realizou aproximadamente $3,11 \times 10^3$ verificações por segundo, enquanto o ECDSA P-521 processou cerca de $5,60 \times 10^2$ verificações por segundo. No nível de 128 bits, entretanto, o ECDSA P-256 permaneceu superior, atingindo aproximadamente $1,36 \times 10^4$ verificações por segundo, frente às $7,44 \times 10^3$ verificações por segundo obtidas pelo ML-DSA-44.

Nas operações de assinatura, o ECDSA manteve vantagem expressiva na categoria de 128 bits, alcançando aproximadamente $4,04 \times 10^4$ assinaturas por segundo, contra $1,45 \times 10^3$ assinaturas por segundo do ML-DSA-44, correspondendo a uma razão aproximada de 28 vezes. Essa diferença diminui nos níveis de segurança mais elevados, embora o ECDSA continue apresentando maior *throughput* nas operações de assinatura.

O SLH-DSA apresentou o menor desempenho entre os algoritmos avaliados nas operações de assinatura. No nível de 128 bits, o SLH-DSA-128s realizou aproximadamente $2,70 \times 10^0$ assinaturas por segundo e $2,71 \times 10^3$ verificações por segundo. Nos níveis de 192 e 256 bits, o algoritmo atingiu aproximadamente $1,30 \times 10^0$ e $1,40 \times 10^0$ assinaturas por segundo, respectivamente, enquanto as operações de verificação alcançaram cerca de $1,48 \times 10^3$ e $1,06 \times 10^3$ verificações por segundo. Esses resultados evidenciam o elevado custo computacional das operações de assinatura em esquemas baseados estritamente em funções de hash, embora o desempenho de verificação permaneça significativamente superior ao de assinatura.

5.3. Desempenho de Estabelecimento de Chaves

A Figura 2 apresenta as operações empregadas no estabelecimento de segredos: geração de chaves e derivação para ECDH, e geração de chaves, encapsulamento e decapsulamento para ML-KEM.

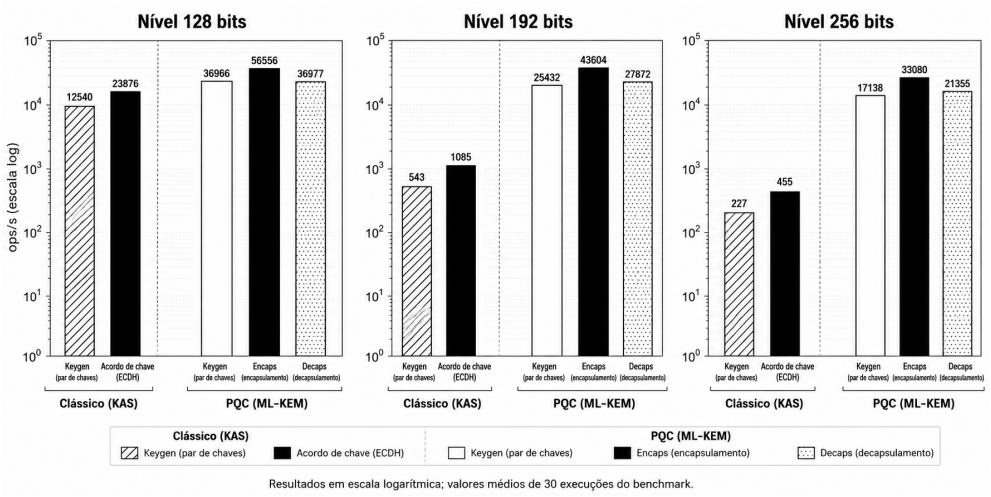

Figura 2. *Throughput* das operações utilizadas no estabelecimento de segredos em mecanismos clássicos e pós-quânticos.

O encapsulamento de ML-KEM-512 atingiu 56.556 operações/s, frente a 23.876 operações/s da derivação baseada em X25519. Para os níveis de 192 e 256 bits, ML-KEM alcançou 43.604 e 33.080 operações/s, enquanto as derivações ECDH com secp384r1 (P-384) e secp521r1 (P-521) atingiram 1.085 e 455 operações/s, respectivamente. Essas comparações caracterizam primitivas distintas e não representam diretamente o custo de procedimentos completos.

Embora ML-KEM tenha apresentado elevado *throughput*, seus parâmetros maiores aumentam o volume de dados transportado. Sobre UDP, isso pode provocar fragmentação ou exigir transporte alternativo; sobre TCP, pode demandar mais segmentos e, dependendo das condições da rede, elevar a latência. Esses efeitos não foram medidos neste trabalho.

5.4. Desempenho de Funções de Hash

Como análise complementar, foram avaliadas SHA-256 e SHA-512, utilizadas por construções criptográficas como SLH-DSA. A Figura 3 apresenta o *throughput* médio em função do tamanho da entrada.

O *throughput* cresceu com o tamanho da entrada, e SHA-256 superou SHA-512 em todas as condições avaliadas. Esses resultados contextualizam o custo das primitivas subjacentes, mas não permitem inferir isoladamente o desempenho de SLH-DSA, que também depende da quantidade e da organização das chamadas de *hash*.

Em conjunto, os resultados indicam que o custo computacional de esquemas baseados em hash está relacionado tanto ao desempenho das primitivas criptográficas subjacentes quanto à estrutura do algoritmo. Dessa forma, o SLH-DSA mostra-se mais

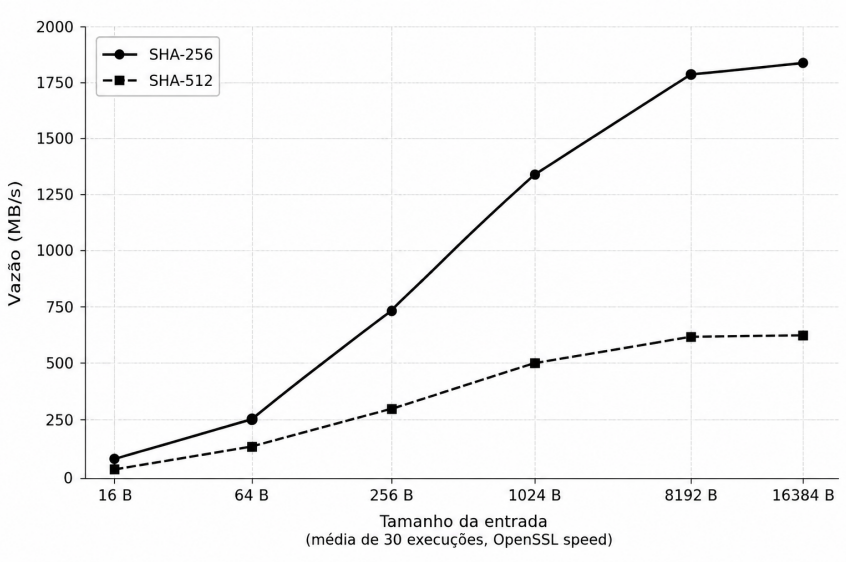

Figura 3. *Throughput* de SHA-256 e SHA-512 por tamanho da mensagem.

adequado a cenários em que a frequência de geração de assinaturas é relativamente baixa, enquanto sua fundamentação em funções de *hash* pode torná-lo atraente em aplicações que valorizem diversidade de premissas criptográficas.

5.5. Desempenho em Modo Híbrido Clássico e Pós-quântico

A transição para PQC pode empregar esquemas híbridos, nos quais mecanismos clássicos e pós-quânticos são combinados por meio de uma função de derivação de chaves [Alagic et al. 2025]. Essa abordagem busca preservar a segurança enquanto ao menos um dos componentes permanecer resistente.

O OpenSSL 3.5 disponibiliza os grupos híbridos X25519MLKEM768, SecP256r1MLKEM768 e SecP384r1MLKEM1024. Neste trabalho, foram avaliados experimentalmente X25519MLKEM768 e SecP256r1MLKEM768; SecP384r1MLKEM1024 não foi incluído nos experimentos.

A Figura 4 apresenta duas análises complementares. À esquerda, compara o *throughput* das operações empregadas no estabelecimento de segredo para mecanismos clássicos e para o ML-KEM em níveis aproximadamente equivalentes de segurança (128, 192 e 256 bits). Para os mecanismos clássicos, foram utilizadas as operações correspondentes ao estabelecimento do segredo compartilhado, enquanto, para o ML-KEM, foram consideradas as operações de encapsulamento. À direita, são apresentados os resultados de duas configurações híbridas representativas, combinando um mecanismo clássico e um mecanismo pós-quântico. Os resultados mostram que o encapsulamento do ML-KEM apresentou maior throughput que as operações clássicas utilizadas como referência para o estabelecimento do segredo compartilhado. No nível de 128 bits, o ML-KEM-512 atingiu aproximadamente 56.556 operações por segundo, enquanto a operação correspondente baseada em X25519 alcançou cerca de 23.876 operações por segundo, correspondendo a uma razão aproximada de 2,37. Para os níveis de 192 e 256 bits, as razões observadas foram de aproximadamente 40,19 e 72,70, respectivamente. Essas comparações devem ser interpretadas com cautela,

pois os mecanismos clássicos de acordo de chaves e os KEMs possuem modelos operacionais distintos e executam operações diferentes durante o estabelecimento do segredo compartilhado.

A configuração híbrida X25519+ML-KEM-768 alcançou aproximadamente 10.113 operações por segundo, enquanto secp256r1+ML-KEM-768 atingiu cerca de 9.075 operações por segundo. Em comparação com a operação clássica baseada em X25519 (23.876 operações por segundo), a configuração híbrida apresentou redução de aproximadamente 57,6% no throughput. Apesar desse custo adicional, as configurações híbridas mantiveram uma elevada taxa de operações no ambiente experimental avaliado. A repercussão desse custo sobre servidores reais, dispositivos embarcados e protocolos completos depende também do tamanho das mensagens, da capacidade do equipamento e das condições da rede, não podendo ser determinada apenas pelos *benchmarks* das primitivas.

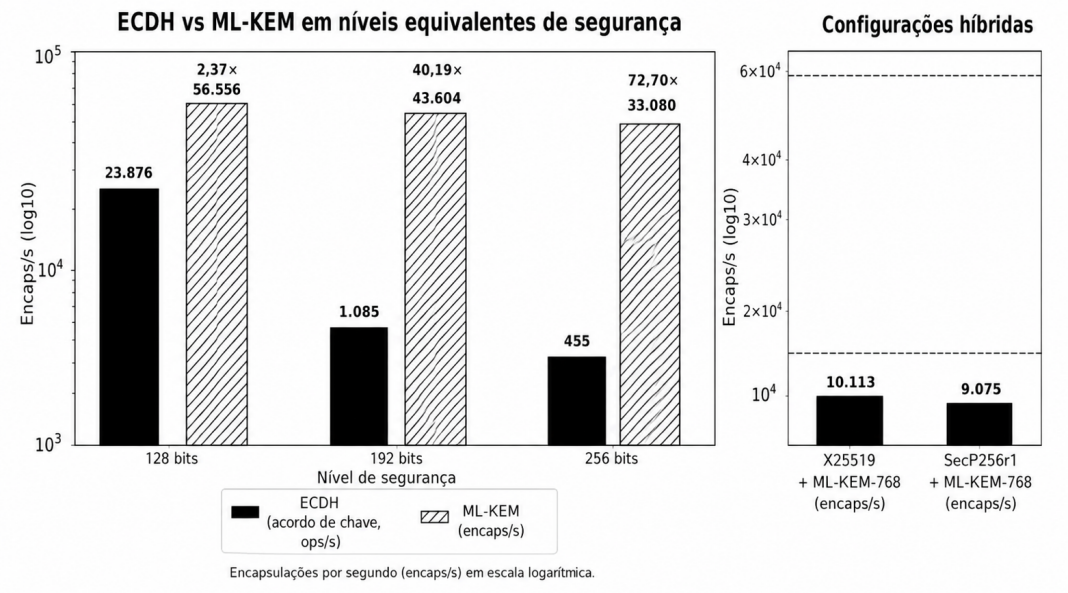

Figura 4. *Throughput* das operações de estabelecimento de segredo para mecanismos clássicos, ML-KEM e configurações híbridas clássico–PQC.

6. Ameaças à Validade

Este estudo utiliza *benchmarks* sintéticos executados com o OpenSSL em um único ambiente computacional. Portanto, os resultados não devem ser interpretados como medições diretas de desempenho em protocolos completos, como TLS ou DNSSEC, mas como estimativas do custo computacional das primitivas criptográficas avaliadas.

Além disso, fatores como versão da biblioteca criptográfica, opções de compilação, suporte a instruções vetoriais (AVX2/AVX-512), frequência dinâmica da CPU e carga do sistema podem influenciar os resultados obtidos. Para reduzir esses efeitos, os testes foram executados em ambiente controlado, com repetição das medições e registro da configuração de hardware e software utilizada.

A comparação entre mecanismos clássicos de acordo de chaves, como ECDH,

e mecanismos pós-quânticos baseados em KEM, como ML-KEM, também deve ser interpretada com cautela. Embora ambos desempenhem funções equivalentes no estabelecimento de segredos compartilhados, seus modelos operacionais e fluxos de protocolo são estruturalmente distintos.

7. Conclusão

Este artigo apresentou uma comparação sistemática entre os algoritmos criptográficos clássicos e os padrões PQC do NIST (FIPS 203, 204 e 205) em duas dimensões: parâmetros estáticos (tamanhos de chave e assinatura) e desempenho das operações criptográficas em hardware convencional.

Os principais resultados obtidos foram: (i) o ML-KEM apresentou *throughput* superior às operações clássicas de referência utilizadas para estabelecimento de segredos compartilhados nos três níveis de segurança avaliados, embora essas comparações devam ser interpretadas com cautela devido às diferenças conceituais entre mecanismos KAS e KEM; (ii) o ML-DSA apresentou menor desempenho na geração de assinaturas em relação ao ECDSA, especialmente na Categoria 1, enquanto apresentou desempenho superior nas operações de verificação nas Categorias 3 e 5; (iii) o SLH-DSA apresentou *throughput* bastante reduzido para geração de assinaturas, mostrando-se mais adequado para aplicações em que assinaturas são produzidas com baixa frequência; (iv) as configurações híbridas X25519MLKEM768 e SecP256r1MLKEM768 apresentaram menor *throughput* que as operações isoladas utilizadas como referência, mas mantiveram taxas elevadas no ambiente avaliado; e (v) os algoritmos pós-quânticos apresentaram parâmetros significativamente maiores que seus equivalentes clássicos, como observado nas comparações entre ML-DSA-44 e ECDSA P-256 e entre ML-KEM-512 e X25519, indicando que parte importante do custo da migração para PQC decorre do aumento do volume de dados transportados e da integração dessas primitivas aos protocolos de comunicação.

Trabalhos futuros incluem avaliar o impacto dos algoritmos pós-quânticos na latência do handshake TLS 1.3, na validação DNSSEC e no consumo energético em plataformas ARM Cortex-A e RISC-V.

Agradecimentos

Este trabalho foi desenvolvido no âmbito do projeto “Avaliação da Implementação e do Impacto de Algoritmos de Criptografia Pós-Quântica nos Protocolos DNSSEC e TLS”, com apoio financeiro do LACNIC por meio do Programa FRIDA.

Referências

Alagic, G., Barker, E., Chen, L., Moody, D., Robinson, A., Silberg, H., and Waller, N. (2025). Recommendations for key-encapsulation mechanisms. Technical Report NIST SP 800-227, National Institute of Standards and Technology (NIST).

Kannwischer, M. J., Rijneveld, J., Schwabe, P., and Stoffelen, K. (2019). pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4. In *Progress in Cryptology – AFRICACRYPT 2019*, pages 176–195. Springer.

NIST (2024a). FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. Technical report, National Institute of Standards and Technology.

NIST (2024b). FIPS 204: Module-Lattice-Based Digital Signature Standard. Technical report, National Institute of Standards and Technology.

NIST (2024c). FIPS 205: Stateless Hash-Based Digital Signature Standard. Technical report, National Institute of Standards and Technology.

NIST (2024). Post-quantum cryptography standardization. Technical report, National Institute of Standards and Technology, Gaithersburg, MD.

Open Quantum Safe Project (2025). Open Quantum Safe. `https://openquantumsafe.org`. Acesso em: 09 maio 2026.

Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134.