Criptografia Pós-Quântica para IoT: Avaliação de Algoritmos em Dispositivos Embarcados

Gustavo G. Teixeira[1], **Bruno G. Batista**[1], **Otávio S. M. Gomes**[2]

[1] Instituto de Matemática e Computação
[2] Instituto de Engenharia de Sistemas e Tecnologia da Informação
Universidade Federal de Itajubá (UNIFEI)
Itajubá, Brasil

{otavio.gomes, brunoguazzelli}@unifei.edu.br, ggimenezt@gmail.com

Abstract. *This paper evaluates the feasibility of employing the Module Lattice-based Key Encapsulation Mechanism (ML-KEM), the post-quantum key exchange standard defined by the National Institute of Standards and Technology (NIST) in 2024, in embedded devices. ML-KEM is compared with classical alternatives, such as RSA and ECDH, using measurements of execution time, energy consumption, and memory usage on an ESP32-based platform. Additionally, a security architecture for secure message exchange in post-quantum IoT scenarios is proposed. Experimental results indicate that ML-KEM achieves faster and more energy-efficient key exchange operations than classical algorithms with equivalent security levels, at the cost of increased stack memory usage. The proposed architecture was implemented and validated on an ESP32-based IoT setup, demonstrating response times compatible with practical applications and confirming the feasibility of ML-KEM in embedded systems with intermediate resource constraints.*

Resumo. *Este artigo avalia a viabilidade do emprego do Module Lattice-based Key Encapsulation Mechanism (ML-KEM), padrão de troca de chaves pós-quântico definido pelo National Institute of Standards and Technology (NIST) em 2024, em dispositivos embarcados. O ML-KEM é comparado com alternativas clássicas, como RSA e ECDH, por meio de medições de tempo de execução, consumo de energia e uso de memória em uma plataforma baseada no ESP32. Também é proposta uma arquitetura de segurança voltada a cenários IoT pós-quânticos. Os resultados experimentais mostram que o ML-KEM apresenta operações de troca de chaves mais rápidas e energeticamente mais eficientes do que algoritmos clássicos de segurança equivalentes, ao custo de um maior consumo de stack. A arquitetura proposta foi implementada e validada em um ambiente IoT com duas placas ESP32, demonstrando tempos de resposta compatíveis com aplicações práticas e confirmando a viabilidade do uso do ML-KEM em dispositivos com recursos intermediários.*

1. Introdução

À medida que sistemas computacionais se tornam mais automatizados e interconectados, aumenta o volume de dados trafegando em redes de *Internet of Things* (IoT). Esses sistemas fazem uso intensivo de microcontroladores e outros dispositivos embarcados, geralmente projetados com foco em baixo custo e baixo consumo de energia, o que impõe

restrições significativas de processamento e memória. Estimativas recentes indicam que o número de dispositivos IoT conectados deve atingir dezenas de bilhões de unidades na próxima década, reforçando a relevância desse ecossistema na infraestrutura digital contemporânea [Myroshnyk 2024].

Como parte das informações transmitidas em redes IoT pode envolver dados sensíveis, como credenciais, dados pessoais ou informações financeiras, esses sistemas tornam-se alvos naturais de ataques cibernéticos. Nesse contexto, o uso de mecanismos criptográficos adequados é essencial para garantir propriedades como confidencialidade, integridade e autenticação das comunicações [Boeckl et al. 2019].

Paralelamente, os avanços na computação quântica representam uma ameaça direta aos principais algoritmos de criptografia assimétrica atualmente empregados, como o *Rivest–Shamir–Adleman* (RSA) e esquemas baseados em curvas elípticas, incluindo o *Elliptic Curve Diffie–Hellman* (ECDH). A existência de algoritmos quânticos, como o algoritmo de Shor [Shor 1997], torna possível a quebra desses esquemas em tempo significativamente inferior ao requerido por algoritmos clássicos, comprometendo a segurança de protocolos amplamente utilizados. Por esse motivo, organismos de padronização e agências de segurança recomendam que sistemas com requisitos de sigilo de longo prazo iniciem o planejamento de migração para algoritmos resistentes a ataques quânticos [Chen et al. 2016].

Nesse cenário, a criptografia pós-quântica (*Post-Quantum Cryptography*, PQC) tem como objetivo o desenvolvimento e a padronização de algoritmos seguros mesmo na presença de adversários com acesso a computadores quânticos. Após um processo de avaliação iniciado em 2016, o *National Institute of Standards and Technology* (NIST) publicou, em 2024, os primeiros padrões de criptografia pós-quântica, incluindo o *Module Lattice-based Key Encapsulation Mechanism* (ML-KEM), baseado na família CRYSTALS-Kyber, para encapsulamento de chaves [NIST 2025]. Embora esses algoritmos tenham sido amplamente analisados do ponto de vista criptográfico, sua adoção em dispositivos embarcados com recursos limitados ainda demanda avaliações específicas quanto ao impacto em métricas como tempo de execução, consumo de energia e uso de memória.

Estudos recentes têm demonstrado a viabilidade da criptografia pós-quântica em plataformas embarcadas, especialmente esquemas baseados em reticulados, como o CRYSTALS-Kyber. Trabalhos recentes indicam que esses algoritmos podem superar abordagens clássicas, como o ECDH, em termos de tempo de execução, embora apresentem maior consumo de memória [Mighri et al. 2024, Kannwischer et al. 2019]. No entanto, a maioria desses estudos avalia métricas de forma isolada ou foca em otimizações específicas, carecendo de uma análise conjunta que considere tempo de execução, consumo de energia e uso de memória em cenários IoT. Além disso, parte desses trabalhos foi conduzida antes da padronização final promovida pelo NIST, baseando-se em versões preliminares dos algoritmos, o que reforça a necessidade de reavaliações utilizando os padrões oficialmente estabelecidos.

Diante disso, este artigo avalia a viabilidade do emprego do ML-KEM em dispositivos embarcados, utilizando o ESP32 como plataforma de referência. O desempenho do ML-KEM é comparado com algoritmos clássicos de segurança equivalente, como RSA

e ECDH, por meio de medições de tempo de execução, consumo de energia e uso de memória. Além disso, é proposta e implementada uma arquitetura de segurança voltada à troca de mensagens em um cenário IoT pós-quântico, na qual o ML-KEM é empregado para estabelecer chaves entre dispositivos embarcados.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os conceitos de base e os trabalhos relacionados. A Seção 3 descreve a metodologia experimental e a arquitetura proposta. A Seção 4 discute os resultados obtidos. Por fim, a Seção 5 apresenta as conclusões e trabalhos futuros.

2. Contexto e Trabalhos Relacionados

2.1. Criptografia Pós-Quântica e Padronização

A criptografia pós-quântica (*Post-Quantum Cryptography*, PQC) surge como resposta às ameaças impostas pela computação quântica, propondo algoritmos de chave pública baseados em problemas matemáticos considerados difíceis tanto para computadores clássicos quanto quânticos [Pranjal and Chaturvedi 2024]. Entre as principais famílias de algoritmos PQC destacam-se os esquemas baseados em *lattices*, códigos corretores de erro e funções *hash*.

Após um processo público de avaliação iniciado em 2016, o NIST publicou, em 2024, os primeiros padrões oficiais de criptografia pós-quântica. Dentre eles, a *Federal Information Processing Standard* (FIPS) 203 define o *Module Lattice-based Key-Encapsulation Mechanism* (ML-KEM), mecanismo de encapsulamento de chaves baseado no algoritmo CRYSTALS-Kyber e fundamentado no problema *Module Learning With Errors* (MLWE) [NIST 2024].

O ML-KEM foi projetado para substituir esquemas clássicos de troca de chaves, como RSA e ECDH, oferecendo níveis equivalentes de segurança sem depender de problemas matemáticos vulneráveis a algoritmos quânticos. Apesar de sua robustez criptográfica, a adoção do ML-KEM em plataformas com recursos restritos pode impor desafios relacionados ao poder de processamento, uso de *stack* e custo energético, que precisam ser avaliados experimentalmente.

2.2. Segurança em Dispositivos Embarcados e IoT

Dispositivos embarcados são amplamente empregados em aplicações da *Internet of Things* (IoT), caracterizadas por restrições severas de processamento, memória e consumo de energia. Essas limitações tornam a implementação de mecanismos criptográficos avançados um desafio, especialmente no contexto da criptografia pós-quântica, cujos algoritmos, em geral, demandam estruturas de dados maiores e operações matemáticas mais complexas do que seus equivalentes clássicos [Kleidermacher and Kleidermacher 2012].

Uma prática comum em sistemas embarcados é o uso de *pre-shared keys* (PSKs), que eliminam a necessidade de protocolos de troca de chaves em tempo de execução. Embora essa abordagem reduza o custo computacional, ela apresenta limitações importantes, como dificuldades de escalabilidade, gerenciamento de chaves e comprometimento total da comunicação em caso de vazamento [Barker 2020]. Nesse contexto, mecanismos dinâmicos de estabelecimento de chaves, como os baseados em encapsulamento de chaves (KEMs), tornam-se alternativas mais adequadas, especialmente em cenários pós-quânticos.

2.3. Trabalhos Relacionados

O desempenho do CRYSTALS-Kyber é avaliado em diferentes plataformas, incluindo o ESP32, evidenciando superioridade em relação ao ECDH em termos de tempo de execução, embora com maior consumo de memória [Mighri et al. 2024]. Resultados semelhantes também são reportados na literatura, como no *framework* pqm4 para avaliação de KEMs e algoritmos de assinatura PQC em microcontroladores ARM Cortex-M [Kannwischer et al. 2019].

Além disso, demonstra-se que o uso de aceleradores criptográficos e paralelismo em plataformas ESP32 permite reduzir significativamente o tempo de execução do Kyber [Segatz and Hafiz 2025]. De forma complementar, outros trabalhos reforçam o potencial do ML-KEM como substituto pós-quântico de algoritmos clássicos em sistemas com recursos moderados, destacando a necessidade de avaliações que considerem consumo energético e integração em arquiteturas completas de comunicação segura [Nagy et al. 2025].

Apesar desses avanços, a literatura ainda carece de uma avaliação integrada que considere simultaneamente tempo de execução, consumo de energia e uso de memória em plataformas embarcadas, bem como da validação de arquiteturas completas de comunicação segura em cenários IoT. Além disso, parte dos estudos existentes baseia-se em versões preliminares dos algoritmos, anteriores à padronização final promovida pelo NIST.

Neste contexto, este trabalho realiza uma avaliação conjunta dessas métricas em uma plataforma ESP32 sob condições controladas, propõe e valida uma arquitetura completa de comunicação segura pós-quântica e avalia o algoritmo ML-KEM com base no padrão oficial do NIST, já em sua versão final padronizada.

3. Metodologia Experimental e Arquitetura

3.1. Escopo do Experimento

O experimento tem como objetivo comparar o desempenho do mecanismo de encapsulamento de chaves pós-quântico ML-KEM com algoritmos clássicos de estabelecimento de chaves, especificamente RSA e ECDH, em uma plataforma embarcada baseada no ESP32. A comparação considera níveis de segurança equivalentes de 128, 192 e 256 bits, conforme definido pelo NIST, e avalia as métricas de tempo de execução, consumo de energia e uso de memória.

Adicionalmente, é proposta e implementada uma arquitetura de troca segura de mensagens em cenário pós-quântico, com o objetivo de demonstrar a viabilidade prática do uso do ML-KEM em aplicações IoT reais.

3.2. Plataforma Experimental

Os experimentos foram conduzidos em uma placa de desenvolvimento ESP32, escolhida por sua ampla adoção em aplicações IoT, baixo custo e suporte a bibliotecas criptográficas consolidadas. As implementações dos algoritmos clássicos (RSA e ECDH) utilizam a biblioteca mbedTLS, integrada ao framework ESP-IDF. Para o ML-KEM, foi empregada a implementação de referência da biblioteca PQClean, alinhada com a padronização do NIST.

O desenvolvimento foi realizado com o framework ESP-IDF v5.3, utilizando o sistema operacional FreeRTOS. Para garantir consistência entre execuções, a frequência da CPU foi fixada em 240 MHz, recursos de economia de energia foram desativados e todas as medições foram realizadas com a mesma configuração de *firmware* e fonte de alimentação.

3.3. Metodologia de Avaliação

A comparação entre os algoritmos foi baseada no conceito de *security strength*, conforme definido pelo NIST. As variantes do ML-KEM foram associadas a níveis de segurança equivalentes aos dos algoritmos clássicos RSA e ECDH, permitindo uma comparação justa entre esquemas criptográficos de naturezas distintas.

Para cada algoritmo, foram medidas separadamente as operações relevantes para o estabelecimento de chaves. No caso do ML-KEM e do RSA, foram avaliadas as etapas de geração de chaves, encapsulamento/criptografia e desencapsulamento/decriptação. Para o ECDH, o tempo total do protocolo foi obtido a partir da geração de chaves efêmeras, troca de chaves públicas e cálculo do segredo compartilhado.

As medições de tempo foram realizadas por meio de temporizadores de alta resolução do ESP-IDF e contadores de ciclos da CPU. O consumo de energia foi medido externamente com o auxílio de um sensor de corrente e tensão, permitindo estimar a energia adicional consumida durante a execução dos algoritmos em relação a um estado de repouso. O uso de memória foi avaliado a partir do consumo de *heap* e do pico de utilização de *stack* da tarefa responsável pela execução dos testes.

Cada experimento foi repetido 1000 vezes, após uma fase inicial de aquecimento, e os resultados foram tratados estatisticamente por meio da mediana e do intervalo interquartil, de modo a reduzir o impacto de valores atípicos. Para garantir reprodutibilidade, foram utilizadas fontes determinísticas de aleatoriedade tanto para o ML-KEM quanto para os algoritmos clássicos.

3.4. Arquitetura de Segurança Pós-Quântica

Com base nos requisitos de segurança, nas restrições impostas por dispositivos embarcados e na literatura recente sobre criptografia pós-quântica, foi definida uma arquitetura de troca segura de mensagens voltada a um cenário IoT, ilustrada na Figura 1.

Após o encapsulamento de chaves, o segredo compartilhado é processado por uma função de derivação de chaves baseada em *hash* (HKDF), gerando chaves independentes para confidencialidade e integridade. A confidencialidade das mensagens é garantida por meio do algoritmo AES-128 no modo CTR, enquanto a integridade e autenticidade são asseguradas por um código de autenticação de mensagem baseado em HMAC-SHA-256, conforme ilustrado no fluxo da Figura 1.

Essa combinação permite a construção de um canal seguro *quantum-safe*, mantendo compatibilidade com as limitações de dispositivos embarcados e concentrando a segurança criptográfica no mecanismo de encapsulamento pós-quântico.

3.5. Validação em Cenário IoT

A arquitetura proposta foi validada em um cenário IoT composto por dois dispositivos ESP32 comunicando-se por meio de uma interface serial. Um dos dispositivos atua

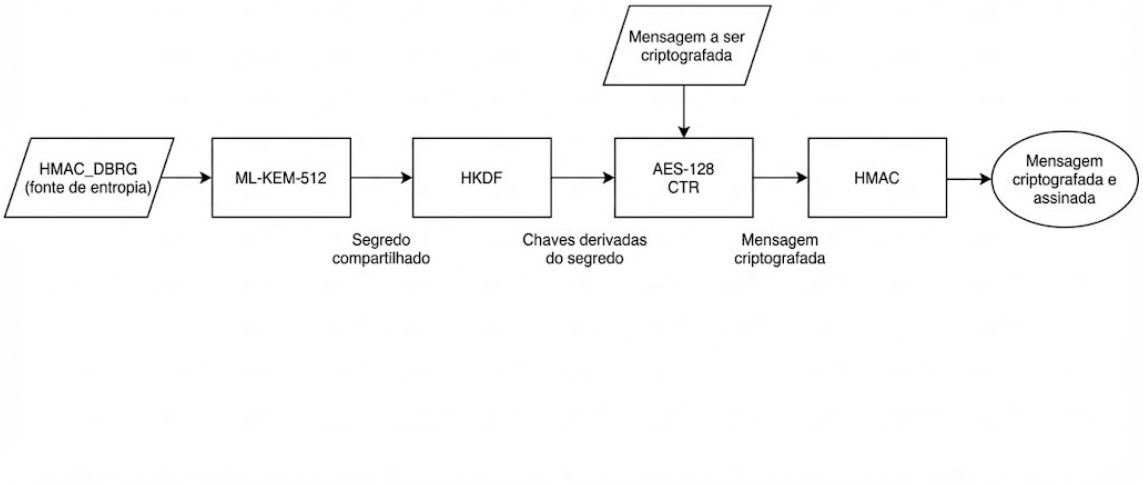

Figura 1. Arquitetura de segurança *quantum-safe* proposta para troca de mensagens entre dispositivos embarcados.

como emissor, lendo dados de um sensor NFC e tratando essas informações como mensagens sensíveis. Os dados são criptografados e autenticados segundo a arquitetura proposta e transmitidos ao segundo dispositivo, que realiza a verificação de integridade e a decriptação da mensagem.

A implementação utiliza um protocolo leve de enquadramento de mensagens, adequado a dispositivos com recursos limitados, e reutiliza as mesmas bibliotecas criptográficas avaliadas nos *benchmarks*. Esse cenário permitiu avaliar, na prática, o impacto do uso do ML-KEM no fluxo completo de comunicação, validando tempos de resposta e consumo de recursos compatíveis com aplicações IoT reais.

Visando à reprodutibilidade dos experimentos, o código-fonte utilizado para a implementação dos algoritmos, bem como os scripts de teste e de pós-processamento de dados, está disponível em um repositório público em:

`https://github.com/ggtxz/MLKEM-on-ESP32`.

4. Resultados e Discussão

4.1. Tempo de Execução

Os resultados de tempo de execução evidenciam uma vantagem expressiva do ML-KEM em relação aos algoritmos clássicos avaliados. Considerando níveis equivalentes de segurança, o ML-KEM apresentou desempenho aproximadamente 36 vezes superior ao ECDH e até 768 vezes superior ao RSA, confirmando sua adequação a plataformas embarcadas. Esse comportamento explica, em parte, a inviabilidade prática do RSA em microcontroladores, especialmente para tamanhos de chave elevados.

No caso do RSA-15360, a geração de chaves não pôde ser concluída no ESP32 devido a limitações de recursos, motivo pelo qual essa configuração foi excluída das

Tabela 1. Distribuição percentual do tempo de execução por fase.

Algoritmo	Keygen (%)	Enc./PubKey (%)	Dec./Shared (%)
ECDH P-256	50.26	0.02	49.72
ML-KEM-512	27.07	32.58	40.35
ML-KEM-768	27.91	32.85	39.24
RSA-3072	93.42	0.23	6.35

medições experimentais. Esse resultado reforça as dificuldades de adoção do RSA em cenários com restrições severas de processamento.

A Figura 2 apresenta a comparação dos tempos médios de execução, levando em consideração o tempo total necessário para o estabelecimento de chaves. Observa-se que o ML-KEM mantém desempenho consistente mesmo com o aumento do nível de segurança, comportamento diretamente relacionado à estrutura do problema de *Module Learning With Errors*, que evita operações custosas como exponenciações modulares ou aritmética em curvas elípticas.

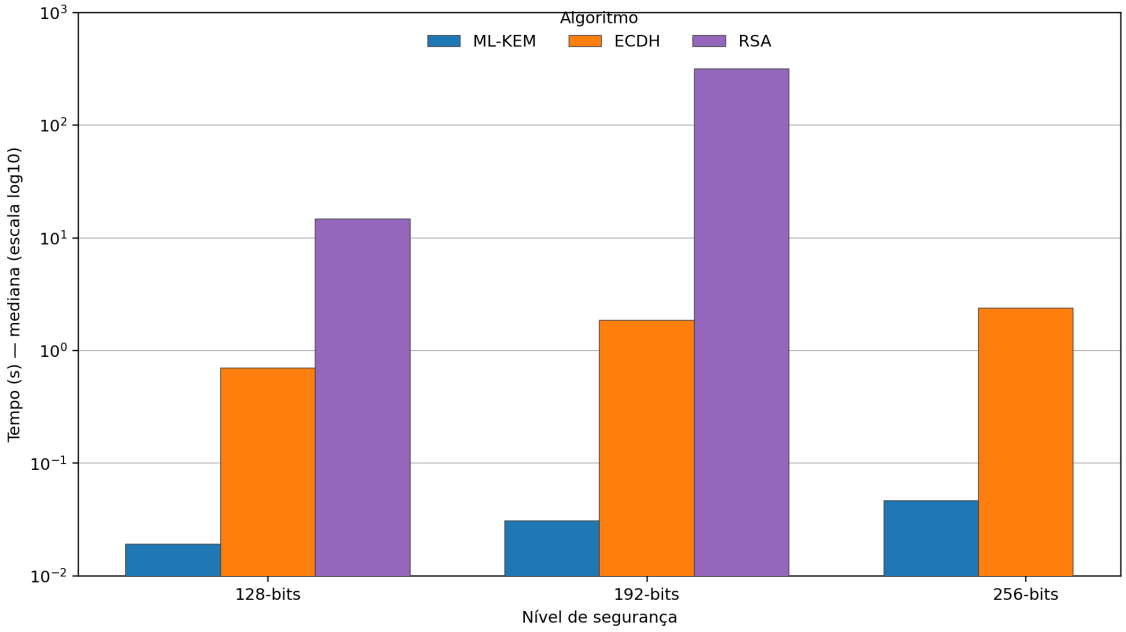

Figura 2. Comparação dos tempos de execução dos algoritmos avaliados.

A Tabela 1 detalha a distribuição percentual do tempo por fase. No ML-KEM, as etapas de geração de chaves, encapsulamento e desencapsulamento apresentam tempos relativamente equilibrados, sem a presença de gargalos dominantes. Em contraste, no ECDH, as fases de geração de chaves e cálculo do segredo compartilhado concentram praticamente todo o custo computacional, enquanto no RSA a geração de chaves responde por cerca de 90% do tempo total, devido à geração de grandes primos e testes de primalidade.

4.2. Consumo de Energia

O comportamento observado nas medições de energia acompanha de forma consistente os resultados de tempo de execução. Como ilustrado na Figura 3, algoritmos com maior tempo médio de execução apresentaram, consequentemente, maior consumo energético total.

O ML-KEM destacou-se novamente por apresentar consumo de energia significativamente inferior ao ECDH e, principalmente, ao RSA. Nos experimentos realizados, as diferenças de potência média instantânea entre os algoritmos foram pequenas quando comparadas às diferenças de tempo, indicando que o tempo de execução é o principal fator determinante do consumo energético total. Esse resultado é particularmente relevante para aplicações IoT alimentadas por bateria, nas quais trocas frequentes de chaves podem impactar diretamente a autonomia do sistema.

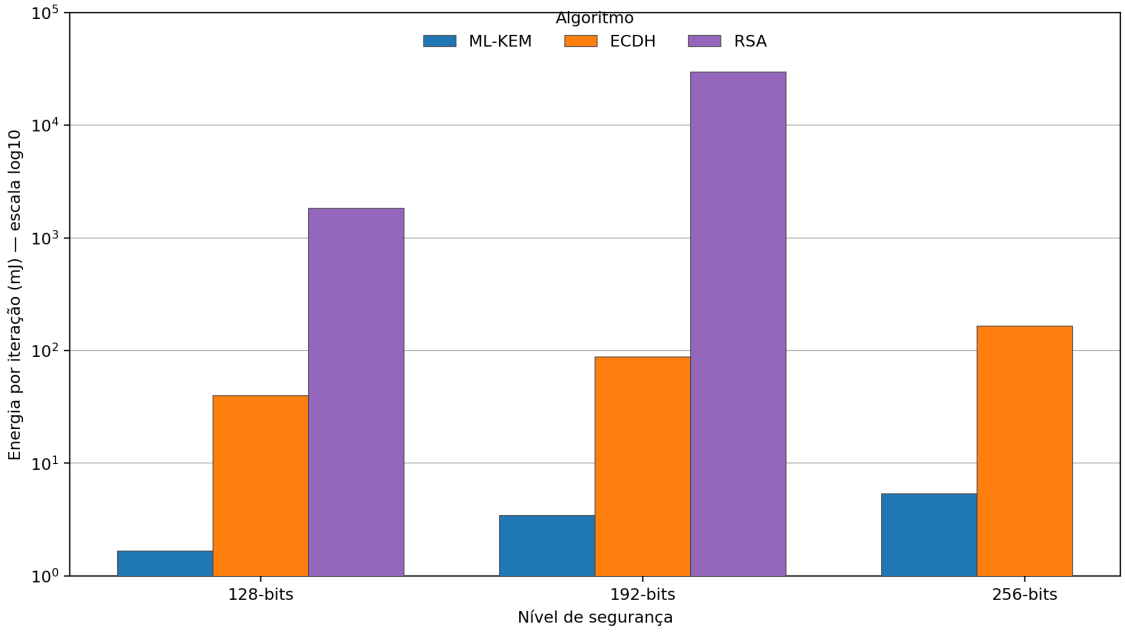

Figura 3. Comparação do consumo energético médio dos algoritmos avaliados.

4.3. Uso de Memória

A análise do uso de memória revelou o principal fator limitante do ML-KEM em dispositivos embarcados: o consumo elevado de *stack*. Conforme apresentado na Figura 4, o ML-KEM demandou aproximadamente quatro vezes mais *stack* do que o ECDH e o RSA. Mesmo na variante ML-KEM-512, foi necessária uma alocação mínima de cerca de 13 kB apenas para a tarefa criptográfica.

Por outro lado, as métricas relacionadas ao uso de *heap* global e ao alocador do mbedTLS não apresentaram diferenças relevantes entre os algoritmos. A implementação do ML-KEM via PQClean praticamente não utiliza alocação dinâmica, enquanto o mbedTLS administra o *heap* de forma eficiente para os algoritmos clássicos. Assim, o consumo de *stack* mostrou-se a métrica mais representativa para a comparação prática entre os esquemas.

Esse resultado indica que, embora o ML-KEM seja vantajoso em termos de tempo e energia, sua adoção em plataformas com memória muito restrita pode ser limitada. Em dispositivos com recursos intermediários, como o ESP32, o algoritmo permanece viável, desde que o projeto do sistema considere explicitamente o orçamento de memória disponível para as rotinas criptográficas.

O elevado consumo de *stack* do ML-KEM pode impactar o escalonamento de tarefas em sistemas baseados em *Real-Time Operating Systems* (RTOS), limitando o nível de concorrência ou exigindo particionamento cuidadoso de memória. Em casos extremos, a alocação insuficiente pode resultar em falhas de execução, reforçando a necessidade de um planejamento explícito do uso de memória.

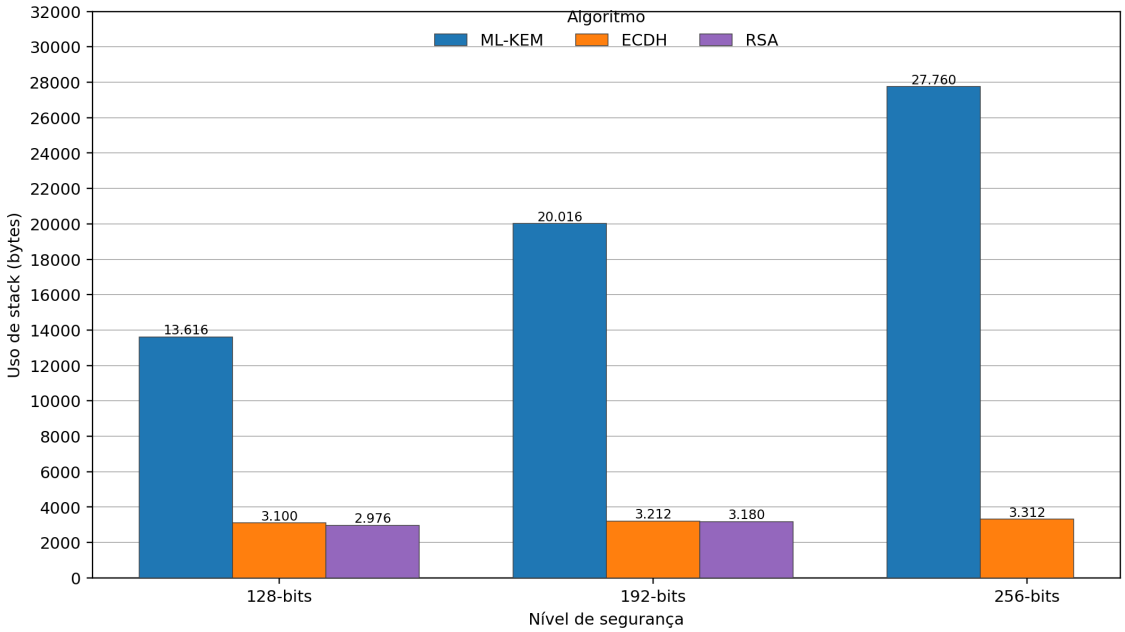

Figura 4. Comparação do consumo de memória de *stack*.

4.4. Discussão dos Resultados

Os resultados experimentais apresentados evidenciam diferenças marcantes entre os algoritmos clássicos de troca de chaves e o esquema pós-quântico ML-KEM quando avaliados em um contexto de dispositivos embarcados. A análise conjunta das métricas de tempo de execução, consumo energético e uso de memória permite compreender de forma mais aprofundada os trade-offs envolvidos na adoção de mecanismos *quantum-safe* em plataformas com recursos limitados.

Do ponto de vista de desempenho temporal, o ML-KEM apresentou vantagem consistente em todos os níveis de segurança avaliados. Mesmo quando comparado ao ECDH, amplamente considerado adequado para sistemas embarcados, o ML-KEM demonstrou tempos de execução significativamente menores. Essa diferença decorre, em grande parte, da natureza algorítmica dos esquemas: enquanto ECDH e RSA dependem de operações aritméticas custosas, como exponenciações modulares e operações em curvas elípticas, o ML-KEM baseia-se predominantemente em operações lineares e

aritmética modular de menor complexidade, mais compatíveis com a arquitetura do microcontrolador utilizado.

A distribuição do tempo entre as fases de execução reforça essa observação. No ML-KEM, as etapas de geração de chaves, encapsulamento e desencapsulamento apresentaram tempos relativamente equilibrados, sem a concentração de custo em uma única fase. Em contraste, no ECDH, praticamente todo o tempo de execução está concentrado nas fases de geração de chaves e cálculo do segredo compartilhado, enquanto no RSA a geração de chaves domina o custo total. Essa concentração torna os algoritmos clássicos mais sensíveis a variações de carga e menos previsíveis em cenários embarcados, especialmente em sistemas com múltiplas tarefas concorrentes.

O comportamento observado nas medições de consumo energético acompanha diretamente os resultados de tempo de execução. A pequena variação de potência média entre os algoritmos indica que o consumo total de energia é majoritariamente determinado pela duração das operações. Assim, a eficiência temporal do ML-KEM se reflete diretamente em menor gasto energético por operação, característica relevante em aplicações IoT alimentadas por bateria, particularmente em cenários nos quais o estabelecimento de chaves ocorre com frequência.

Em contrapartida, a análise de uso de memória revelou o principal ponto de atenção do ML-KEM no contexto embarcado. O consumo elevado de *stack*, observado mesmo na variante ML-KEM-512, impõe um custo fixo significativo à aplicação. Enquanto ECDH e RSA demandam quantidades relativamente modestas de *stack*, o ML-KEM exige um orçamento de memória consideravelmente maior, o que pode limitar sua adoção em microcontroladores com RAM reduzida ou em sistemas nos quais múltiplas tarefas competem por recursos de memória.

Esse resultado evidencia um trade-off claro: os ganhos em tempo de execução e eficiência energética são obtidos ao custo de um aumento significativo no consumo de memória. Na prática, isso implica que a escolha do algoritmo deixa de ser baseada exclusivamente em desempenho e passa a depender diretamente do perfil de recursos da aplicação. Em plataformas intermediárias, como o ESP32 utilizado neste trabalho, esse custo mostrou-se administrável, desde que o projeto considere explicitamente a alocação de memória para as rotinas criptográficas. Em sistemas mais restritos, entretanto, o consumo de *stack* pode se tornar um fator limitante, exigindo otimizações adicionais ou a adoção de esquemas alternativos.

Sob a perspectiva de projeto de sistemas, esses resultados sugerem que o ML-KEM é mais adequado a dispositivos IoT com recursos moderados, nos quais o orçamento de memória não é o principal gargalo e há demanda por eficiência energética ou menor latência no estabelecimento de chaves. Por outro lado, em dispositivos com restrições severas de memória, ou em aplicações com alto grau de concorrência entre tarefas, algoritmos como o ECDH permanecem relevantes, uma vez que apresentam menor pressão sobre o uso de *stack*, ainda que com maior custo computacional.

Adicionalmente, deve-se considerar que as implementações utilizadas influenciam diretamente os resultados observados. A utilização da biblioteca PQClean, não otimizada para plataformas embarcadas, pode amplificar o consumo de *stack* do ML-KEM. Dessa forma, é possível que implementações otimizadas reduzam parcialmente esse custo, alte-

rando o equilíbrio observado entre desempenho e uso de memória.

Os resultados obtidos estão alinhados com a literatura recente, que também aponta vantagens de desempenho de esquemas baseados em reticulados em relação ao ECDH [Mighri et al. 2024].

De maneira ampla, observa-se que, no contexto de sistemas embarcados, a memória tende a assumir um papel central na adoção de criptografia pós-quântica. Em particular, o consumo de *stack* emerge como um dos principais fatores de restrição prática, podendo influenciar diretamente decisões de arquitetura, como o número de tarefas concorrentes e o particionamento de recursos no sistema.

Em síntese, o ML-KEM apresenta um perfil de desempenho favorável para dispositivos embarcados com recursos moderados, oferecendo ganhos expressivos em tempo de execução e consumo energético em relação aos algoritmos clássicos. No entanto, sua adoção prática depende de uma análise cuidadosa do orçamento de memória disponível, o que reforça a necessidade de considerar não apenas métricas de desempenho, mas também as restrições estruturais da plataforma alvo no projeto de arquiteturas de segurança *quantum-safe*.

4.5. Validação da Arquitetura Pós-Quântica

A arquitetura de segurança proposta foi validada em um cenário IoT composto por dois dispositivos ESP32. No ensaio realizado, a troca de chaves via ML-KEM-512, seguida da derivação de chaves e da proteção das mensagens com AES-128-CTR e HMAC-SHA-256, ocorreu corretamente, permitindo a recuperação íntegra da mensagem no dispositivo receptor.

O tempo total observado para o processamento completo da mensagem foi de aproximadamente 181 ms, com consumo de *stack* em torno de 16 kB para a tarefa responsável pela arquitetura de segurança. Considerando que o próprio ML-KEM-512 exige cerca de 13 kB de *stack*, os resultados indicam que a arquitetura é viável em dispositivos que disponham desse orçamento mínimo de memória.

Do ponto de vista prático, o tempo de resposta obtido é compatível com aplicações interativas e reforça a viabilidade do uso do ML-KEM como mecanismo de estabelecimento de chaves em arquiteturas de comunicação seguras para dispositivos embarcados em um cenário pós-quântico.

4.6. Ameaças à Validade

Esta seção discute as principais ameaças à validade dos resultados apresentados, considerando os aspectos de validade interna, externa e de construção, conforme recomendado para estudos experimentais em sistemas computacionais.

Validade interna. Uma possível ameaça à validade interna está relacionada à implementação dos algoritmos criptográficos. As rotinas do ML-KEM utilizadas neste trabalho foram obtidas diretamente da biblioteca PQClean, que não é especificamente otimizada para dispositivos embarcados. Embora essa escolha garanta conformidade com o padrão e facilite a reprodutibilidade, ela pode impactar diretamente métricas como tempo de execução e, principalmente, consumo de *stack*. No entanto, a mesma abordagem foi

adotada de forma consistente para todas as medições, e os algoritmos clássicos foram implementados utilizando bibliotecas amplamente consolidadas, reduzindo o risco de viés sistemático nas comparações.

Outra ameaça diz respeito às medições de tempo e energia. Apesar do uso de instrumentação dedicada para monitoramento de consumo energético, variações decorrentes de ruído elétrico, resolução do sensor e interferência de outras tarefas do sistema não podem ser completamente eliminadas. Para mitigar esse efeito, os experimentos foram repetidos múltiplas vezes e conduzidos em ambiente controlado, buscando garantir consistência nos resultados observados.

Validade externa. A validade externa do estudo é limitada pela utilização de uma única plataforma de *hardware*, o ESP32, com uma configuração específica de microcontrolador, memória e frequência de operação. Embora o ESP32 represente uma classe relevante de dispositivos embarcados intermediários, os resultados obtidos não podem ser diretamente generalizados para microcontroladores mais restritos ou para arquiteturas substancialmente diferentes. Além disso, o desempenho dos algoritmos pode variar de acordo com otimizações específicas de compilador, arquitetura de CPU ou bibliotecas criptográficas utilizadas.

Validade de construção. As métricas adotadas, tempo de execução, consumo de energia e uso de memória, são amplamente empregadas na avaliação de algoritmos criptográficos em sistemas embarcados e refletem de forma adequada os custos práticos da adoção de mecanismos de segurança. Ainda assim, outras métricas relevantes, como latência em cenários de rede mais complexos ou impacto em aplicações multitarefa, não foram consideradas. Dessa forma, embora os resultados capturem os principais custos computacionais e energéticos, eles não abrangem todos os aspectos possíveis de uso em aplicações reais.

Apesar dessas limitações, as escolhas metodológicas adotadas permitem uma análise consistente e reprodutível do comportamento dos algoritmos avaliados, fornecendo evidências sólidas sobre os trade-offs envolvidos na adoção do ML-KEM em dispositivos embarcados de porte intermediário.

5. Conclusões

Este artigo apresentou uma avaliação prática do *Module Lattice-based Key Encapsulation Mechanism* (ML-KEM) em comparação com algoritmos clássicos de troca de chaves, como ECDH e RSA, em um cenário realista de dispositivos embarcados baseado no ESP32. A análise considerou métricas de tempo de execução, consumo de energia e uso de memória, permitindo avaliar de forma objetiva os impactos da adoção de um esquema pós-quântico em plataformas com recursos limitados.

Os resultados experimentais mostraram que o ML-KEM apresenta desempenho significativamente superior aos algoritmos clássicos em termos de tempo de execução e consumo energético, sendo substancialmente mais eficiente que o ECDH e, sobretudo, que o RSA. O consumo de energia acompanhou essa tendência, indicando que a eficiência temporal do algoritmo se traduz diretamente em menor custo energético por operação, característica particularmente relevante em aplicações IoT com restrições de energia ou com necessidade de estabelecimento frequente de chaves.

Em contrapartida, a análise de uso de memória evidenciou o principal fator limitante do ML-KEM: o consumo elevado de *stack*. Mesmo na configuração ML-KEM-512, observou-se a necessidade de aproximadamente 13 kB de *stack* apenas para a execução do algoritmo, o que impõe restrições à sua adoção em dispositivos com memória reduzida ou em sistemas com forte concorrência entre tarefas. Dessa forma, embora o desempenho computacional seja favorável, o orçamento de memória se estabelece como um aspecto central no projeto de sistemas que pretendem incorporar criptografia pós-quântica.

A viabilidade prática do esquema foi reforçada pela implementação e validação de uma arquitetura de segurança *quantum-safe* em um cenário IoT composto por dois dispositivos ESP32. O fluxo completo de leitura de dados, troca de chaves com ML-KEM, cifragem, autenticação e recuperação da mensagem ocorreu corretamente, com consumo de *stack* em torno de 16 kB e tempo de resposta aproximado de 181 ms. Esses resultados indicam que, em plataformas com recursos intermediários, a arquitetura proposta é compatível com requisitos práticos de desempenho e uso.

De forma geral, os resultados indicam que o ML-KEM é uma alternativa viável para troca de chaves em dispositivos embarcados com recursos moderados, especialmente em cenários nos quais eficiência energética e baixa latência são requisitos relevantes. Por outro lado, em sistemas com restrições severas de memória, o consumo de *stack* pode limitar sua adoção, tornando algoritmos clássicos como o ECDH ainda opções pertinentes dependendo do contexto de aplicação.

Assim, a adoção de mecanismos *quantum-safe* em sistemas embarcados não deve ser guiada exclusivamente por métricas de desempenho, mas sim por uma análise conjunta das restrições de recursos da plataforma e dos requisitos da aplicação. Nesse sentido, o trabalho contribui ao evidenciar que, embora tecnicamente viável, a incorporação do ML-KEM implica trade-offs que devem ser considerados explicitamente no projeto de sistemas IoT.

5.1. Limitações e Trabalhos Futuros

Como continuidade deste trabalho, destacam-se as seguintes direções de pesquisa:

- Avaliação da arquitetura proposta em cenários IoT mais complexos, incluindo múltiplos nós, diferentes padrões de tráfego e integração com serviços externos.
- Otimização das implementações do ML-KEM para dispositivos embarcados, com foco na redução do consumo de *stack*, mesmo que à custa de desempenho.
- Incorporação de esquemas de assinatura digital pós-quântica, como o ML-DSA, para autenticação da troca de chaves e das mensagens.
- Replicação da metodologia experimental em outras plataformas de *hardware*, permitindo uma comparação mais ampla dos custos de adoção da criptografia pós-quântica em diferentes microcontroladores.
- Investigação do impacto do tamanho de mensagens e do custo de comunicação associado aos esquemas baseados em reticulados, especialmente em cenários com largura de banda limitada.
- Refinamento das medições de consumo energético, considerando taxas de amostragem mais elevadas e cenários de uso mais próximos de aplicações reais.

Agradecimentos

Os autores gostariam de agradecer todo o apoio e infraestrutura fornecida pela Universidade Federal de Itajubá. Este artigo contou com o uso de ferramentas de Inteligência Artificial Generativa como apoio à escrita e revisão. As ferramentas foram utilizadas exclusivamente para apoio linguístico e estrutural, não tendo sido empregadas na geração de dados experimentais, resultados, análises quantitativas, gráficos ou decisões metodológicas. Os autores assumem total responsabilidade pelo conteúdo apresentado neste trabalho.

Referências

Barker, E. (2020). Recommendation for key management: Part 1 – general. Technical report, NIST, Gaithersburg, MD.

Boeckl, K., Fagan, M., Fisher, W., Lefkovitz, N., Megas, K., Nadeau, E., Piccarreta, B., O'Rourke, D. G., and Scarfone, K. (2019). Considerations for managing internet of things (iot) cybersecurity and privacy risks. NIST Interagency/Internal Report (NISTIR) 8228, National Institute of Standards and Technology, Gaithersburg, MD.

Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., and Smith-Tone, D. (2016). Report on post-quantum cryptography. Technical report, NIST, Gaithersburg, MD.

Kannwischer, M. J., Rijneveld, J., Schwabe, P., and Stoffelen, K. (2019). pqm4: Testing and benchmarking NIST PQC on ARM cortex-m4. Cryptology ePrint Archive, Paper 2019/844.

Kleidermacher, D. and Kleidermacher, M. (2012). *Embedded Systems Security: Practical Methods for Safe and Secure Software and Systems Development*. Elsevier, Netherlands, first edition. eBook.

Mighri, M. A., Benfarah, A., and Meddeb, A. (2024). Performance evaluation and benchmarking of pqc crystals-kyber on embedded devices. In *2024 IEEE/ACS 21st International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7.

Myroshnyk, Y. (2024). State of iot – summer 2024. Market Report. 171-page report including market updates, forecasts, and trends.

Nagy, N., Alnemer, S., Alshuhail, L. M., Alobiad, H., Almulla, T., Alrumaihi, F. A., Ghadra, N., and Nagy, M. (2025). Module-lattice-based key-encapsulation mechanism performance measurements. *Sci*, 7(3).

NIST (2024). Module-lattice-based key-encapsulation mechanism standard. Federal Information Processing Standards Publication FIPS 203, National Institute of Standards and Technology, Gaithersburg, MD.

NIST (2025). Post-quantum cryptography. CSRC Project Page. Acesso em: 30 Jun. 2025.

Pranjal and Chaturvedi, A. (2024). Post-quantum cryptography.

Segatz, F. and Hafiz, M. I. A. (2025). Efficient implementation of crystals-kyber key encapsulation mechanism on esp32.

Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509.