GERAÇÃO DE CERTIFICADOS KEM COM ACME

Alexandre Boutrik[1], Alexandre Augusto Giron[2]

RESUMO: A criptografia moderna, usada para proteger informações sensíveis, baseia-se em problemas matemáticos complexos que são computacionalmente inviáveis para se resolver com a computação clássica. No entanto, com o surgimento da computação quântica, algoritmos como o de Shor podem resolver de maneira exponencialmente mais rápida esses problemas. Assim, torna-se mais complexo para a comunidade de segurança da informação proteger essas informações. O objetivo do projeto é verificar a viabilidade de uma proposta de melhoria ao protocolo ACME - utilizado para a geração de certificados digitais para a plataforma web - com *benchmarks* de algoritmos resistentes ao computador quântico. Foram realizados *benchmarks* na placa embarcada Raspberry Pi 3 Model B+ (servindo como cliente), executando o sistema Raspberry Pi OS (antigamente conhecido como Raspbian). O software escolhido é o Lego (implementado em linguagem Go) e inclui os algoritmos necessários para os testes, como por exemplo os algoritmos KEM (*Key Encapsulation Mechanism*). As medidas escolhidas para comparação foram: tempo de emissão de certificado (capturado com *patches*), tempo de execução total de um ciclo de emissão no protocolo (capturado pelo software hyperfine) e o consumo total de memória do processo (capturado usando um código python). Os resultados demonstram que o mecanismo KEM, proposto como melhoria, reduziu o tempo médio de emissão de certificado, o que o torna mais atrativo para ambientes que demandam alta eficiência e velocidade na geração de certificados. Constatou-se, porém, um aumento de consumo de memória pela proposta. Esse incremento o torna menos atrativo para ambientes embarcados, que possuem recursos limitados, e ambientes onde múltiplas operações de emissão de certificados são realizadas simultaneamente. Portanto, enquanto a melhoria no tempo de emissão seja uma vantagem, é essencial analisar o *trade-off* entre esse benefício e o aumento de consumo de memória, levando em consideração a infraestrutura disponível e as necessidades de cada ambiente.

Palavras-chave: criptografia pós-quântica, ACME, benchmark.

1 INTRODUÇÃO

A criptografia moderna, utilizada nos dias atuais, possui muitos de seus algoritmos baseados em problemas matemáticos considerados difíceis de resolver pela computação clássica, como a fatoração de grandes números e o logaritmo discreto. No entanto, esses problemas, que garantem a segurança de algoritmos como o RSA (RIVEST; SHAMIR; ADLEMAN, 1978), podem ser resolvidos de forma exponencialmente mais rápida por

[1] Graduando, Campus Toledo, UTFPR - Universidade Tecnológica Federal do Paraná, alexandreboutrik@alunos.utfpr.edu.br
[2] Professor, Campus Toledo, UTFPR - Universidade Tecnológica Federal do Paraná, alexandregiron@utfpr.edu.br

computadores quânticos, colocando em risco a segurança de sistemas que são amplamente utilizados nas comunicações atualmente.

Em meio a essa crescente ameaça, protocolos como o ACME se destacam. O ACME (do inglês *Automatic Certificate Management Environment*, que significa Ambiente de Gerenciamento Automático de Certificados), é um protocolo que padroniza o processo de emissão, renovação e revogação de certificados digitais, principalmente para servidores web. Ele permite a obtenção de certificados digitais de forma automatizada, sem nenhuma intervenção humana. Dessa forma, é possível que os servidores, junto a uma Autoridade Certificadora (CA), consigam validar suas identidades, garantindo assim a autenticidade de suas comunicações.

No entanto, atualmente, como definido em sua documentação (BARNES et al., 2019), o ACME utiliza algoritmos criptográficos modernos, que, conforme discutido anteriormente, são vulneráveis aos ataques de computadores quânticos. Estes, por meio de algoritmos como o de Shor (SHOR, 1994), são capazes de reduzir significativamente as complexidades de seus problemas matemáticos. A possibilidade de que esses algoritmos sejam quebrados em um futuro próximo coloca em risco a segurança dos certificados digitais emitidos por esse protocolo. Diante desse cenário, se torna necessário explorar alternativas como a criptografia pós-quântica (BERNSTEIN & LANGE, 2017), que podem oferecer maior resistência contra as ameaças trazidas pela computação quântica.

O objetivo deste artigo é realizar *benchmarks* comparativos entre duas abordagens de sistemas criptográficos pós-quânticos propostos pelo *draft* experimental (GIRON, 2023) para o protocolo ACME: certificados PQTLS e certificados KEMTLS. Os certificados PQTLS são gerados utilizando chaves de assinatura digital (dilithium assinada com dilithium), enquanto os certificados KEMTLS se baseiam na técnica de KEM (*Key Encapsulation Mechanism*). Através desses *benchmarks*, podemos avaliar o impacto de cada abordagem, oferecendo *insights* sobre a viabilidade dessas soluções na prática em diferentes ambientes de produção.

2 METODOLOGIA

Como ambiente de testes, foi utilizada a placa Raspberry Pi 3 Model B+ como máquina cliente, equipada com um processador ARMv8 de 1.4 GHz e 1 GB de memória, executando o software Lego (GO-ACME, 2024) escrito em linguagem Go. No lado do

servidor, foi empregado um notebook Acer Nitro 5 AN515-58-58W3. No entanto, o foco deste trabalho é exclusivamente na máquina cliente, tendo como objetivo a avaliação da eficiência do protocolo ACME em termos de tempo e consumo de memória por parte do cliente.

Para a análise de tempo, foram realizados 100 *benchmarks* utilizando o software hyperfine (PETER, 2022) coletando as métricas: tempo total de um ciclo completo de emissão do protocolo (em segundos), e tempo necessário para a emissão de um certificado (em milissegundos).

Também foram realizados, de forma isolada da outra experimentação, 25 outros *benchmarks* sob o monitoramento do script ps_mem.py (PIXELB, 2024), a fim de medir o consumo de memória (em kilobytes) pelo processo Lego durante todo período de execução.

3 RESULTADOS E DISCUSSÃO

Os resultados dos testes são apresentados nas Tabelas 1 e 2. Na primeira coluna de ambas as tabelas, a métrica utilizada refere-se ao ciclo completo do protocolo ACME, que inclui as etapas de criação de conta, resolução dos desafios e geração de certificado. A segunda coluna apresenta os tempos correspondentes exclusivamente à etapa de geração de certificado, isolando essa fase crítica do processo. Por fim, a terceira e última coluna mostra o consumo total de memória pelo processo.

Tabela 1 – *Benchmark* para geração de certificados PQTLS

MEDIDAS	Ciclo completo (s)	Emissão (ms)	Consumo de memória (KiB)
Média	41,4	2402	894
Mínimo	33,5	592	739
Máximo	54,9	8380	952
Desvio padrão	4,1	2482	48.8

Fonte:Autoria Própria.

Tabela 2 – *Benchmark* para geração de certificados KEMTLS

MEDIDAS	Ciclo completo (s)	Emissão (ms)	Consumo de memória (KiB)
Média	41,7	819	1074
Mínimo	35,6	624	806
Máximo	54,4	1802	1218
Desvio padrão	3,8	208	97,4

Fonte: Autoria Própria.

Os resultados demonstram que o mecanismo KEM oferece vantagens se comparado aos certificados PQTLS. A principal conclusão é que a implementação baseada em encapsulamento resulta em uma melhoria no tempo médio de emissão de certificado. Isso é particularmente relevante para ambientes que demandam alta eficiência e velocidade na geração de certificados.

Por outro lado, os resultados também demonstram que o uso de certificados KEM está relacionado a um aumento de consumo de memória. Esse incremento o torna menos atrativo para sistemas embarcados, que possuem recursos limitados, e ambientes onde múltiplas operações de emissão de certificados são realizadas simultaneamente.

Portanto, enquanto a melhoria no tempo de emissão seja uma vantagem, é essencial analisar o *trade-off* entre esse benefício e o aumento de consumo de memória, levando em consideração a infraestrutura disponível e as necessidades de cada ambiente.

4 CONSIDERAÇÕES FINAIS

Esse trabalho buscou apresentar uma análise comparativa entre as propostas de modelos pós-quânticos para o protocolo ACME, certificados PQTLS e KEMTLS, oferecendo uma clareza sobre quais tipos de ambientes são mais adequados para se utilizar cada um, visando proteger os certificados digitais contra as ameaças emergentes da computação quântica.

É importante destacar algumas limitações observadas durante a experimentação. Os resultados da Tabela 1 apresentam um desvio padrão muito próximo à média, o que sugere algum tipo inconsistência ou falha durante a execução dos testes, provavelmente relacionado a hibernação do sistema operacional. Além disso, foi percebido um aumento contínuo no consumo de memória ao longo da experimentação, em ambos os casos, nos certificados PQTLS e KEMTLS, indicando uma provável existência de *memory leak*, *unexpected behavior* por parte do *garbage collector* ou algum problema similar envolvendo o gerenciamento de memória do processo. Outra limitação observada foi a instabilidade na rede da UTFPR-TD, que mesmo configurada adequadamente de forma cabeada, pode ter impactado negativamente a precisão dos resultados obtidos.

Para trabalhos futuros, seria interessante aprimorar o ambiente de testes, adotando um sistema minimalista, sem interface gráfica, a fim de minimizar possíveis interferências na performance dos experimentos. Ademais, a realização dos testes em uma rede isolada seria ideal para evitar problemas na estabilidade da conexão entre a máquina cliente e o servidor. Outro aspecto relevante seria a inclusão do terceiro modelo criptográfico pós-quântico sugerido pelo *draft* experimental na comparação, o KEM-POP, que utiliza de provas de zero conhecimento para auxiliar na prova de posse de chave, uma técnica para geração de certificados KEM diferente da abordagem utilizada no KEMTLS.

REFERÊNCIAS

BARNES, R. et al. RFC 8555: Automatic Certificate Management Environment (ACME). Disponível em: <https://datatracker.ietf.org/doc/html/rfc8555/>. Acesso em 18 de setembro de 2024.

BERNSTEIN, Daniel J.; LANGE, Tanja. Post-quantum cryptography. Nature, v. 549, n. 7671, p. 188-194, 2017.

GIRON, Alexandre Augusto. ACME PQC Algorithm Negotiation. Disponível em: <https://datatracker.ietf.org/doc/html/draft-giron-acme-pqcnegotiation-02/>. Acesso em 18 de setembro de 2024.

GO-ACME. GitHub - go-acme/lego: Let's Encrypt/ACME client and library written in Go. Disponível em: <https://github.com/go-acme/lego>. Acesso em 18 de setembro de 2024.

PETER, D. hyperfine. Disponível em: <https://github.com/sharkdp/hyperfine>. Acesso em 18 de setembro de 2024.

PIXELB. GitHub - pixelb/ps_mem: A utility to accurately report the in core memory usage for a program. Disponível em: <https://github.com/pixelb/ps_mem>. Acesso em 18 de setembro de 2024.

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, v. 21, n. 2, p. 120–126, 1978.

SHOR, P. W. Algorithms for quantum computation: discrete logarithms and factoring. Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994.