Resumo: KEMTLS vs. Post-Quantum TLS: Performance On Embedded Systems

Gabriel dos Santos Schmitz

Agosto 2024

TLS is widely used to secure communications across various devices, but current public key cryptosystems could become vulnerable to quantum computers. To address this, post-quantum cryptography (PQC) is being developed, with standards from NIST expected soon. However, some NIST PQC finalists pose challenges for small microcontrollers due to their large key sizes or hardware requirements.
KEMTLS is an alternative TLS handshake protocol that uses key encapsulation mechanisms (KEMs) for authentication instead of signatures, potentially offering better efficiency. This study compares KEMTLS with TLS 1.3 in an embedded environment using a Cortex-M4-based platform and the WolfSSL library. The comparison includes performance metrics such as runtime, memory usage, traffic volume, and code size across different network settings like broadband, LTE-M, and Narrowband IoT. Results indicate that KEMTLS can reduce handshake times by up to 38traffic volume compared to TLS 1.3.

Gabriel dos Santos Schmitz: UTFPR. Eu agradeço colegas e orientador por comentários e discussões úteis. Este trabalho foi apoiado por Grupo PQC UTFPR

1. Introdução

. Segurança de Camada de Transporte (TLS) utiliza um handshake entre servidores e clientes para troca e verificação de chaves de criptografia, empregando o algoritmo Diffie-Hellman com curvas elípticas efêmeras. As chaves públicas são trocadas e verificadas por certificados assinados por autoridades e instalados previamente nos dispositivos. Com o avanço da computação quântica, o NIST está desenvolvendo novos padrões de criptografia pós-quântica (PQ), incluindo Mecanismos de Encapsulamento de Chaves (KEMs). Os algoritmos híbridos combinam a troca clássica de chaves elípticas e PQ para maior segurança. Dilithium, um dos esquemas de assinatura, é lento e aumenta o handshake em 17 kb, enquanto Falcon, que exige operações de ponto flutuante duplo, é menor mas menos viável em hardware limitado. O KEMTLS propõe uma alternativa ao usar KEMs para autenticação, sendo mais eficiente em termos de recursos e reduzindo a base de código. Com a crescente conexão de dispositivos via IPv6, como visto no protocolo Matter, a criptografia para dispositivos embarcados enfrentará novos desafios.

1.1. Contribuição

. Este trabalho investiga se as vantagens do KEMTLS se aplicam a sistemas embarcados, comparando-o com o PQTLS em um ambiente embutido. Implementaram-se KEMTLS e PQTLS com todos os esquemas de assinatura e KEMs finalistas do NIST, exceto o Classic McEliece devido ao tamanho excessivo das chaves públicas, ultrapassando a ROM desponível na plataforma. A comparação direta de desempenho é possível devido à sobreposição significativa de código entre as implementações. O estudo avalia tempo de execução, uso de memória, tamanho do código e consumo de largura de banda das implementações em uma plataforma baseada em Cortex-M4. Os experimentos simulam uma conexão TLS entre um dispositivo embarcado e um servidor de alto desempenho, utilizando tecnologias de rede típicas. O trabalho apresenta a criptografia pós-quântica, o processo de padronização, os protocolos PQTLS e KEMTLS, suas diferenças e os detalhes da implementação e configuração experimental, culminando nos resultados e conclusões.

2. Plano de Fundo

. Nesta seção, abordaremos o desenvolvimento da criptografia pós-quântica, incluindo uma visão geral histórica e o processo de padronização do NIST. Também detalharemos o impacto da criptografia pós-quântica no TLS 1.3 e o desenvolvimento do KEMTLS.

2.1. Criptografia Pós Quântica

. Em 1994, Peter Shor criou algoritmos quânticos que ameaçam a criptografia pública atual, incluindo o TLS, que seria comprometido por computadores quânticos grandes. O projeto de padronização de criptografia pós-quântica (PQC) do NIST começou em 2017, com mais de 60 propostas, das quais apenas 4 KEMs e 3 algoritmos de assinatura foram finalistas. Entre os KEMs baseados em redes estão Kyber, Saber e NTRU, enquanto Falcon e Dilithium são baseados em assinaturas de rede. O Classic McEliece, baseado em códigos, foi descartado do artigo por seu tamanho de chave grande. Os algoritmos são avaliados em três níveis de segurança: I, III e V, e o estudo foca apenas no nível I. Falcon é baseado em NTRU, Dilithium em Module-LWE e Short Integer Solution, e Rainbow é uma assinatura multivariada variante do UOV, recentemente quebrada por Beullens. A criptografia pós-quântica é desafiadora para dispositivos embarcados devido às limitações de hardware, e os testes foram realizados sem reduzir a frequência do Cortex-M4.

	bytes transmitted			**stored**	**computation (≈Kcycles)**		
Signatures	pubkey	sig.	sum	secret	keygen	sign	verify
Dilithium ★	1 312	2 420	3 732	2 528	1 597	4 095	1 572
Falcon ★	897	690	1 587	1 281	163 994	39 014	473
Rainbow †	161 600	66	161 666	103 648	94	907	238
KEMs	pubkey	ciph.	sum	secret	keygen	encaps	decaps
Kyber ★	800	768	1 568	1 632	440	539	490
NTRU †	699	699	1 398	953	2 867	565	538
SABER †	672	736	1 408	1 568	352	481	453

★: Scheme was selected for standardization.
†: Scheme was eliminated from the NIST standardization project.

FIGURE 1. Comparação dos finalistas da Rodada 3

Comparação dos finalistas da Rodada 3 do NIST PQC no nível de segurança I. Mostramos o tamanho (em bytes) dos dados transmitidos durante um handshake (chave pública, assinatura e ciphertext), dados offline (chaves secretas) e tempos de operação (de [9, 17]) no M4.

2.2. TLS Pós Quântico

. O TLS 1.3, amplamente utilizado no HTTPS, autentica o servidor para o cliente e oferece suporte opcional para autenticação mútua. O handshake unilateral autenticado por certificado utiliza uma troca de chaves Diffie-Hellman (DH) efêmera, seguida de uma assinatura sobre o handshake para autenticar, finalizando com autenticação adicional via MAC. Para adaptar o TLS 1.3 ao cenário pós-quântico, é possível substituir a geração de chaves DH do servidor pelo encapsulamento de uma chave usando KEMs (Key Encapsulation Mechanisms) e adotar algoritmos de assinatura pós-quânticos em vez de RSA ou assinaturas baseadas em curvas elípticas. Mesmo com essas mudanças, o TLS 1.3 mantém

sua eficiência, necessitando apenas de 1.5-RTT (round-trip time).

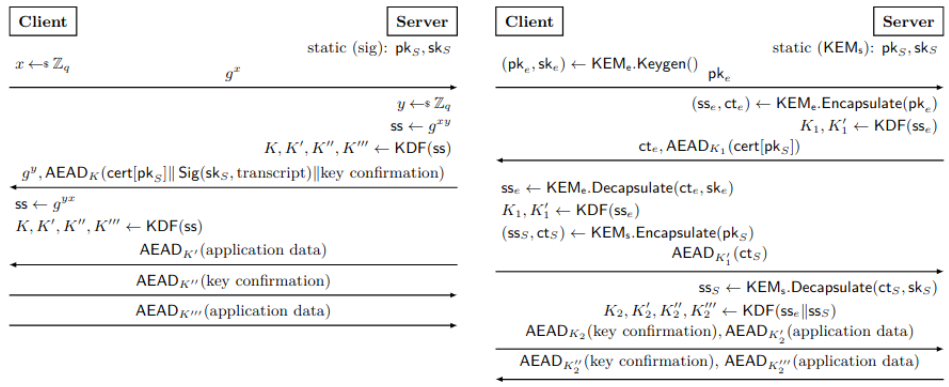

FIGURE 2. Fluxos de protocolo

Diagramas simplificados do fluxo do protocolo: (à esquerda) o handshake do TLS 1.3, usando assinaturas para autenticação do servidor; e (à direita) o handshake do KEMTLS, usando KEMs para autenticação do servidor.

. O KEMTLS é uma proposta de handshake TLS pós-quântico que substitui assinaturas tradicionais por mecanismos de encapsulamento de chave (KEMs), que são menores e mais eficientes em termos computacionais. No KEMTLS, os certificados contêm chaves públicas para KEMs, enquanto as assinaturas ainda são usadas para verificar a cadeia de certificados de forma offline. Isso otimiza o tamanho das chaves públicas e assinaturas, melhorando a eficiência computacional. Inspirado pelo OPTLS de Krawczyk e Wee, o KEMTLS evita penalizações de desempenho ao permitir que o cliente envie sua solicitação na mesma etapa do TLS 1.3, mantendo o protocolo eficiente com apenas 1,5 RTT.

3. Configuração Experimental

. A seção a seguir descreve o ambiente experimental utilizado para obter nossos resultados. Ambos os protocolos foram avaliados em termos de tempo de handshake, tempo de execução dos algoritmos, uso máximo de memória, tamanho do código e tráfego de rede. Os tempos de handshake foram medidos em três ambientes de rede relevantes para o IoT: internet "broadband", LTE-M e Narrowband-IoT (NB-IoT). As características desses ambientes são detalhadas na Tabela 2. O ambiente "broadband" é baseado em tempos realistas de comunicação cliente-nuvem na Europa Ocidental, enquanto LTE-M e NB-IoT têm suas características baseadas em números do 3GPP.

Name	Abbrev.	Bandwidth	RTT time
Broadband	BB	1 Mbit	26 ms
LTE Machine Type Communication	LTE-M	1 Mbit	120 ms
Narrowband-IoT	NB-IoT	46 kbit	3 s

FIGURE 3. Características de conexão de acordo com 3GPP [1]

. Como o KEMTLS é um protocolo pós-quântico, os experimentos não consideraram certificados mistos ou métodos híbridos (pós-quântico e curva elíptica). Tanto KEMTLS quanto PQTLS utilizam uma autoridade certificadora (CA) para assinar certificados, mas diferem no tipo de chave pública incluída nos certificados folha: KEMs em KEMTLS e chaves de assinatura em PQTLS. Foram avaliadas combinações dos finalistas do NIST, exceto Classic McEliece devido ao tamanho excessivo de suas chaves públicas. Apenas os parâmetros do nível de segurança mais baixo, o nível I do NIST, foram utilizados por serem mais compactos e eficientes.

3.1. Implementação

. Todos os benchmarks foram realizados em uma placa Silicon Labs STK3701A, conhecida como "Giant Gecko", equipada com um processador ARM Cortex-M4F de 72 MHz e memória suficiente para suportar as chaves públicas do Rainbow. O Cortex-M4 é a plataforma de referência do NIST para dispositivos embarcados, com implementações otimizadas para a maioria dos algoritmos finalistas fornecidas pelo projeto PQM4. As implementações de criptografia pós-quântica (PQC) utilizadas foram retiradas do projeto PQM4, com pequenas modificações, e compiladas usando GCC 11.1 com otimização de velocidade.

. O Giant Gecko atuou como cliente (KEM)TLS, conectado a um servidor backend, com a comunicação feita via Ethernet. O servidor simulava diferentes ambientes de rede usando o framework netem do Linux. O sistema usou o Zephyr RTOS, um sistema operacional em tempo real de código aberto com suporte para TCP/IP e TLS. A implementação do servidor KEMTLS foi baseada no código de Wiggers, e o WolfSSL foi escolhido pela sua eficiência de memória e suporte para TLS 1.3. A integração de algoritmos pós-quânticos no WolfSSL foi simplificada pela API clara do WolfCrypt, com modificações para suportar chaves KEM.

. Em ambos os experimentos, PQTLS e KEMTLS, o cliente realizou apenas a verificação de assinaturas, sem necessidade de código de assinatura. Os resultados dos benchmarks foram recebidos via comunicação serial, e todos os experimentos utilizaram a cifra TLS 1.3 TLS_CHACHA20_POLY1305_SHA256.

4. Resultados

. O texto discute os trade-offs importantes para desenvolvedores de sistemas embarcados entre tamanho de código, uso de memória, tráfego de rede e tempo de CPU ao utilizar KEMTLS e TLS 1.3 com finalistas do NIST PQC. O tempo de execução dos algoritmos afeta o consumo de energia, especialmente em dispositivos alimentados por bateria. O tráfego de rede também impacta o consumo de energia e pode ser caro dependendo da tecnologia sem fio. Os benchmarks apresentados focam em plataformas Cortex-M4 e incluem resultados de tamanho de código, uso de memória, tráfego e duração do handshake, e tempo de execução das primitivas PQC. Rainbow, um dos finalistas do NIST, está incluído apenas nos resultados do KEMTLS devido ao grande tamanho de suas chaves públicas. Todos os algoritmos PQC utilizados foram otimizados para velocidade.

4.1. Armazenamento e Consumo de Memória

. As implementações dos protocolos KEMTLS e TLS 1.3 têm tamanhos de código similares, ambos com aproximadamente 111 kB sem as primitivas pós-quânticas. A Tabela 3 apresenta as combinações de algoritmos PQC e seus respectivos tamanhos de código. KEMTLS é mostrado com uma única instância de KEM para troca de chaves efêmeras e autenticação, enquanto TLS 1.3 utiliza dois algoritmos de verificação de assinatura distintos, se usados algoritmos diferentes para certificados CA e de folha. NTRU se destaca com um tamanho de código superior a 200 kB quando usado para troca de chaves efêmeras, mas não quando utilizado apenas para autenticação. O certificado CA com chave pública Rainbow ocupa entre 33do espaço de armazenamento devido ao seu grande tamanho, mas Rainbow ainda é viável em sistemas embarcados devido ao tamanho pequeno das assinaturas e tempos rápidos de verificação.

	KEX	Auth.	CA	PQC code (%)	CA size (%)	Memory
KEMTLS	Kyber	Kyber	Dilithium	29.0 kB (20.1%)	3.9 kB (2.7%)	49.7 kB
	Kyber	Kyber	Falcon	25.7 kB (18.6%)	1.7 kB (1.2%)	52.8 kB
	Kyber	Kyber	Rainbow	29.8 kB (9.8%)	161.8 kB (53.4%)	167.0 kB
	NTRU	NTRU	Dilithium	203.4 kB (63.9%)	3.9 kB (1.2%)	49.7 kB
	NTRU	NTRU	Falcon	200.0 kB (63.9%)	1.7 kB (0.6%)	52.8 kB
	NTRU	NTRU	Rainbow	204.0 kB (42.8%)	161.8 kB (33.9%)	182.9 kB
	SABER	SABER	Dilithium	31.5 kB (21.5%)	3.9 kB (2.7%)	49.7 kB
	SABER	SABER	Falcon	28.2 kB (20.0%)	1.7 kB (1.2%)	52.8 kB
	SABER	SABER	Rainbow	32.2 kB (10.5%)	161.8 kB (53.0%)	167.9 kB
PQTLS	Kyber	Dilithium	Dilithium	29.0 kB (20.1%)	4.0 kB (2.8%)	58.0 kB
	Kyber	Falcon	Dilithium	34.4 kB (23.0%)	4.0 kB (2.7%)	60.7 kB
	Kyber	Falcon	Falcon	25.8 kB (18.6%)	1.8 kB (1.3%)	56.2 kB
	NTRU	Dilithium	Dilithium	203.4 kB (63.8%)	4.0 kB (1.3%)	56.6 kB
	NTRU	Falcon	Dilithium	208.7 kB (64.4%)	4.0 kB (1.2%)	59.3 kB
	NTRU	Falcon	Falcon	200.1 kB (63.9%)	1.8 kB (0.6%)	54.8 kB
	SABER	Dilithium	Dilithium	31.5 kB (21.5%)	4.0 kB (2.7%)	58.0 kB
	SABER	Falcon	Dilithium	36.8 kB (24.2%)	4.0 kB (2.6%)	60.7 kB
	SABER	Falcon	Falcon	28.2 kB (20.0%)	1.8 kB (1.3%)	56.2 kB

FIGURE 4. Comparação KEMTLS e PQTLS nível I NIST

Tamanhos de código e certificado CA (e como porcentagem do tamanho total da ROM), e uso máximo de memória nos experimentos. Os conjuntos de parâmetros usados são do nível I do NIST.

. Além disso, os esquemas baseados em redes (lattice-based) têm bom desempenho em termos de consumo de memória, que é principalmente impulsionado pelo uso da pilha dos algoritmos de assinatura PQC. A exceção é Rainbow, cuja chave pública grande precisa ser mantida na memória durante a verificação de assinatura, exigindo grande alocação de heap. Seria possível otimizar isso armazenando a chave pública já em um formato utilizável na flash, mas essa otimização não foi incluída para permitir resultados comparáveis de código reutilizável.

4.2. Termos de Handshake

. Os tempos de handshake são fundamentais em ambientes embarcados, assim como o consumo de armazenamento e memória. A Tabela 4 apresenta os tempos de handshake medidos em milhões de ciclos para diferentes tecnologias de transmissão, e a Figura 2 ilustra esses tempos e o tráfego para cenários de banda larga e NB-IoT. Em uma implantação real, o dispositivo poderia entrar em modo de baixo consumo durante transmissões lentas, dependendo das especificações do sistema.

. Nos experimentos, o CPU operou a 72 MHz para garantir resultados consistentes. A Tabela 4 também destaca a porcentagem de ciclos dedicados às primitivas de criptografia pós-quântica (PQC), enquanto os ciclos restantes foram utilizados pela máquina de estado

TLS, operações de memória ou espera por I/O.

. Em configurações de banda larga e LTE-M, as operações criptográficas têm um impacto significativo. No entanto, na NB-IoT, essas operações representam uma pequena porcentagem do tempo total (0,8largura de banda e alta latência, o tamanho das transmissões de certificados e chaves públicas é o principal fator que determina o tempo de execução. Além disso, o carregamento de grandes chaves públicas da memória afeta a eficiência do algoritmo Rainbow, que, de outra forma, seria rápido.

	KEX	Auth.	CA	Handshake traffic	Handshake time in Mcycles (% of crypto) BB (%)	LTE-M (%)	NB-IoT (%)
KEMTLS	Kyber	Kyber	Dilithium	6.3 kB	17.1 (30.2%)	34.0 (15.2%)	593.6 (0.9%)
	Kyber	Kyber	Falcon	4.5 kB	12.3 (27.2%)	25.7 (13.0%)	467.8 (0.7%)
	Kyber	Kyber	Rainbow	3.9 kB	11.3 (25.1%)	20.4 (13.9%)	459.0 (0.6%)
	NTRU	NTRU	Dilithium	6.0 kB	21.3 (46.0%)	38.1 (25.6%)	595.8 (1.6%)
	NTRU	NTRU	Falcon	4.2 kB	16.6 (47.8%)	25.9 (30.6%)	469.7 (1.7%)
	NTRU	NTRU	Rainbow	3.6 kB	15.7 (47.4%)	24.7 (30.1%)	361.6 (2.1%)
	SABER	SABER	Dilithium	6.0 kB	16.3 (29.4%)	33.3 (14.4%)	590.8 (0.8%)
	SABER	SABER	Falcon	4.2 kB	11.6 (25.5%)	21.0 (14.1%)	464.8 (0.6%)
	SABER	SABER	Rainbow	3.6 kB	10.7 (23.1%)	19.8 (12.5%)	356.8 (0.7%)
PQTLS	Kyber	Dilithium	Dilithium	8.4 kB	19.9 (35.9%)	36.8 (19.5%)	818.1 (0.9%)
	Kyber	Falcon	Dilithium	6.3 kB	15.5 (33.0%)	29.0 (17.6%)	586.4 (0.9%)
	Kyber	Falcon	Falcon	4.5 kB	10.9 (30.1%)	21.0 (15.6%)	464.6 (0.7%)
	NTRU	Dilithium	Dilithium	8.3 kB	24.3 (47.6%)	41.1 (28.1%)	821.3 (1.4%)
	NTRU	Falcon	Dilithium	6.1 kB	19.9 (47.8%)	33.4 (28.5%)	590.6 (1.6%)
	NTRU	Falcon	Falcon	4.3 kB	15.2 (50.3%)	25.4 (30.2%)	468.0 (1.6%)
	SABER	Dilithium	Dilithium	8.3 kB	19.7 (35.2%)	36.6 (19.0%)	817.3 (0.8%)
	SABER	Falcon	Dilithium	6.1 kB	15.3 (32.0%)	28.8 (17.0%)	586.2 (0.8%)
	SABER	Falcon	Falcon	4.3 kB	10.7 (28.5%)	20.9 (14.6%)	464.0 (0.7%)

FIGURE 5. Tempo de handshake e execução

Tráfego de handshake e tempo de execução para vários cenários. Os conjuntos de parâmetros usados são do nível I do NIST.

. Os KEMs superam o Dilithium na autenticação devido ao tempo de verificação mais rápido e ao menor tamanho de chave e assinatura, o que reduz o tempo de transmissão. O Rainbow, quando usado como certificado de autoridade certificadora (CA), oferece os tempos de handshake mais rápidos, especialmente em KEMTLS. O Falcon se destaca no servidor, mas tem operações de assinatura mais lentas que os KEMs. Em cenários de banda larga e LTE-M, o desempenho do PQTLS com Falcon e SABER é comparável ao do KEMTLS com Rainbow e SABER.

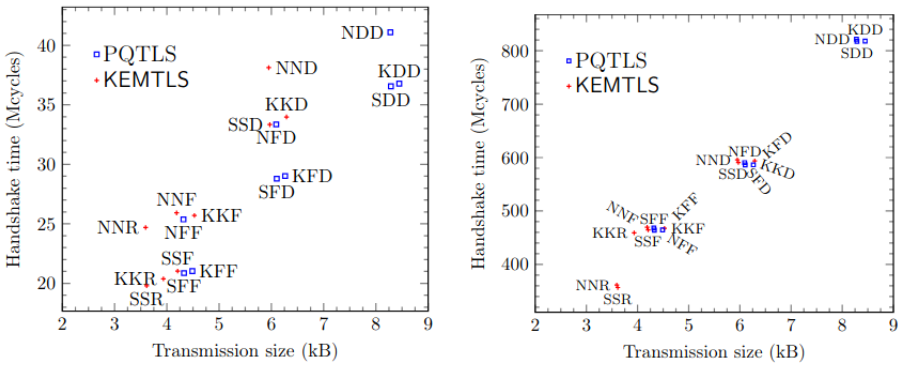

FIGURE 6. Tempo de handshake e tráfego para instâncias

Tempos de handshake e tráfego para instâncias de KEMTLS e PQTLS. As letras representam os algoritmos (D)ilithium, (F)alcon, (K)yber, (N)TRU, (R)ainbow e (S)ABER nos papéis de troca de chaves efêmeras, autenticação de handshake e CA, nessa ordem.

5. Discussão

. Os resultados mostram que o KEMTLS com autenticação apenas do servidor consome menos memória que o PQTLS e possui tamanhos de código semelhantes. O PQTLS com Falcon se destaca em termos de eficiência de verificação, superando outras instâncias, exceto em cenários com SABER ou NTRU combinados com Rainbow, onde o KEMTLS é mais eficiente devido ao menor uso de memória. Falcon também se sai melhor que Dilithium no lado do cliente em qualquer cenário, especialmente em ambientes de largura de banda muito baixa, como NB-IoT. No entanto, a assinatura do Falcon pode não ser adequada para autenticação pós-quântica sem suporte de hardware, onde uma combinação de Dilithium e Falcon pode ser mais eficaz

6. Conclusão e Trabalho Futuro

. Neste artigo, comparou-se o desempenho de KEMTLS e TLS 1.3 usando finalistas de criptografia pós-quântica (PQC) do NIST em um ambiente embarcado com um cliente Cortex-M4 e um servidor de classe desktop. O KEMTLS consome menos memória que o TLS 1.3 devido ao menor uso de memória dos KEMs, embora o tamanho do código seja similar. As primitivas PQC exigem mais de 50protocolos. No ambiente NB-IoT, as assinaturas pequenas do Rainbow são vantajosas, enquanto o Falcon oferece bom desempenho devido à sua verificação eficiente, apesar de ser caro para assinaturas. Futuras pesquisas devem explorar a autenticação do cliente e servidores embarcados, e a redução de largura de banda com chaves pré-distribuídas.