Desenvolvimento de um *testbed* para implementação e avaliação de protocolos de segurança em redes veiculares

Laura Naves, Marco Amaral Henriques

Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Campinas, SP, Brasil

lnnaves@dac.unicamp.br, maah@unicamp.br

Abstract. *This work presents a testbed under development for the implementation and evaluation of security protocols in wireless vehicular networks. Built with Containernet and Mininet-WiFi, the environment enables the emulation of scenarios with nodes running real software inside Docker containers. As a proof of concept, the testbed evaluates a post-quantum variant of EDHOC protocol in a drone network scenario, measuring end-to-end handshake latency and captured traffic. The results demonstrate the feasibility of the environment for reproducible experiments and efective comparisons among security protocols.*

Resumo. *Este trabalho apresenta um testbed em desenvolvimento para implementação e avaliação de protocolos de segurança em redes veiculares sem fio. Construído com Containernet e Mininet-WiFi, o ambiente permite emular cenários com nós em contêineres Docker executando software real. Como prova de conceito, são avaliadas versões pré e pós-quânticas do protocolo EDHOC em rede de drones, avaliando a latência fim a fim do handshake e o tráfego capturado. Os resultados demonstram a viabilidade do ambiente para experimentos reprodutíveis e comparações efetivas entre protocolos de segurança.*

1. Introdução

As redes veiculares, frequentemente associadas ao conceito de V2X (*Vehicle-to-Everything*), representam um ecossistema de comunicação no qual veículos interagem com outros veículos (V2V ou *Vehicle-to-Vehicle*), infraestruturas viárias (V2I ou *Vehicle-to-infrastructure*), pedestres (V2P ou *Vehicle-to-pedestrian*) e serviços em rede ou nuvem (V2N/V2C ou *Vehicle-to-Network/Vehicle-to-Cloud*). Esse cenário oferece suporte a aplicações críticas e cooperativas, como alertas de colisão, coordenação de tráfego, compartilhamento de sensores e condução automatizada. Para isso, diferentes tecnologias de comunicação podem ser empregadas, incluindo enlaces sem fio dedicados, comunicação celular e arquiteturas futuras baseadas em redes 5G e além [Wang et al. 2019].

Devido à natureza aberta, dinâmica e sem fio dessas redes, a segurança torna-se um requisito fundamental. A comunicação veicular está sujeita a ameaças como negação

Trabalho realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – Processo nº 131788/2025-6. Contou com auxílio de IA na revisão do texto e na depuração de códigos.

de serviço, interferência, falsificação de identidade, ataques de repetição, manipulação de mensagens e exposição de dados sensíveis de localização. Nesse contexto, mecanismos criptográficos e protocolos de autenticação são essenciais para garantir o sigilo, integridade, autenticidade, com troca segura de chaves. A avaliação prática desses mecanismos em ambientes controlados é importante para compreender seu impacto em métricas como latência, sobrecarga computacional e custo de comunicação [Tariq and Ahanger 2025].

O objetivo deste trabalho é desenvolver um *testbed* flexível para implementar e avaliar protocolos de segurança em redes veiculares sem fio. O ambiente integra emulação sem fio, execução de *software* real em nós conteinerizados e coleta de métricas como latência, tráfego capturado e custo de comunicação. Como avaliação inicial, considera-se um cenário de rede veicular aérea com Sistemas Aéreos Não Tripulados (UAS ou *Unmanned Aircraft System*), no qual um drone (UAV ou *Unmanned Aerial Vehicle*) é autenticado por uma estação de controle em solo (GCS ou *Ground Control Station*). Como prova de conceito, são implementados e avaliados os protocolos de troca de chaves EDHOC (*Ephemeral Diffie-Hellman Over COSE*) e sua variante pós-quântica PQ-EDHOC (*Post-Quantum Ephemeral Diffie-Hellman Over COSE*).

Trabalhos recentes têm explorado plataformas de simulação para enxames de UAS, mas ainda com limitações quanto ao realismo experimental. Em [Mustafovski 2025], a avaliação é conduzida em MATLAB e NS3, sem execução de *software* real nos nós da rede e considerando uma escala reduzida, com 10 drones e alcance de 500 metros. Já em [Veara et al. 2025], os autores utilizam Gazebo, Docker e controladores ArduPilot conteinerizados para avaliar um mecanismo de segurança em enxames de UAS. Entretanto, a comunicação entre os nós ocorre sobre a rede Docker via UDP, sem emular fielmente um canal sem fio real.

Diante dessas limitações, este trabalho apresenta as seguintes contribuições: (i) implementação de um *testbed* que integra Containernet e Mininet-WiFi para emular redes sem fio e executar *software* real em nós conteinerizados; (ii) organização dos experimentos em um *pipeline* programável e reprodutível; (iii) instrumentação automática com captura de tráfego, *logs* e métricas experimentais; e (iv) avaliação preliminar de protocolos de segurança considerando tráfego capturado e latência de estabelecimento de sessão.

2. Ferramentas de simulação/emulação de redes consideradas durante a seleção do ambiente experimental.

Vários ambientes de simulação e emulação foram considerados e avaliados de forma exploratória, por meio da análise de documentação, exemplos disponíveis e testes preliminares de uso. A Tabela 1 resume as ferramentas analisadas e suas principais características em relação aos requisitos deste trabalho. A escolha do Containernet integrado ao Mininet-WiFi foi guiada por critérios eliminatórios definidos de acordo com esses requisitos.

Em primeiro lugar, a ferramenta deveria permitir emulação de rede, e não apenas simulação, uma vez que era necessário executar *software* real e utilizar recursos da pilha de rede do Linux, como interfaces virtuais, *namespaces*, roteamento e mecanismos de controle de tráfego. Em segundo lugar, o ambiente deveria oferecer suporte à emulação de redes sem fio e mobilidade, requisito importante para representar o cenário com drones. Em terceiro lugar, deveria permitir a execução de aplicações reais em nós isolados,

preferencialmente por meio de contêineres Docker, possibilitando a customização do *software* executado em cada nó. Além disso, era desejável controlar parâmetros do enlace e recursos computacionais dos nós.

Nesse contexto, ferramentas como OMNeT++ [Varga and Hornig 2008], Veins/SUMO [Al-Shareeda and Manickam 2023], UTSim [Al-Mousa et al. 2019] e ns-3 [Tropea et al. 2025, Campanile et al. 2020] foram descartadas por serem mais orientadas à simulação e/ou ou visualização. Por outro lado, Mininet [Dholakiya et al. 2020], Mininet-WiFi [dos Reis Fontes and Rothenberg 2019] e Containernet [Peuster et al. 2016], quando considerados isoladamente, não atendem simultaneamente todos os requisitos: o Mininet não oferece suporte nativo a redes sem fio, o Mininet-WiFi não permite a execução de nós como contêineres Docker e o Containernet não fornece suporte sem fio e mobilidade como o Mininet-WiFi. Assim, a combinação Mininet-WiFi com Containernet foi selecionada por reunir emulação sem fio, mobilidade, uso de recursos reais da pilha de rede do Linux e execução de *software* real em contêineres.

Tabela 1. Comparativo entre ferramentas de simulação/emulação de redes.

Ferramenta	Tipo	Foco	Características
OMNeT++	Simulador de eventos discretos	Modelagem de redes	Exige modelagem dos componentes, em vez da execução direta de software real.
Veins/SUMO	Simulação veicular	Mobilidade e tráfego urbano	Foca mobilidade veicular, mas não execução direta de aplicações reais em contêineres.
ns-3	Simulador de eventos discretos	Protocolos de rede	Possui modelos detalhados, mas é orientado principalmente à simulação.
UTSim	Simulador de drones	Simulação de tráfego	Simulador sem foco em pilha Linux real ou protocolos conteinerizados.
Mininet	Emulador cabeado/SDN	SDN e OpenFlow	Não possui suporte nativo a redes sem fio.
Mininet-WiFi	Emulador sem fio	mobilidade e SDN	Não enfatiza, isoladamente, a execução de nós como contêineres Docker.
Containernet	Emulador com contêineres	Docker, SDN e aplicações reais	Não fornece, isoladamente, suporte sem fio e mobilidade como o Mininet-WiFi.
Mininet-WiFi + Containernet	Emulador sem fio com contêineres	mobilidade, SDN e Docker	Integra emulação sem fio com nós Docker, permitindo executar protocolos reais em cenários controláveis.

3. Protocolos e cenário de avaliação

A criptografia pós-quântica (PQC) reúne algoritmos projetados para resistir a ataques quânticos. Sua importância decorre do fato de que esquemas tradicionais de chave pública, como RSA e ECDSA, se tornarão ineficazes diante de computadores quânticos. Embora computadores quânticos de grande escala ainda não sejam uma realidade, a migração para PQC exige estudo antecipado, pois o desenvolvimento, a padronização e a adoção de novos protocolos demandam tempo [Nguyen et al. 2025]. Com o objetivo de mostrar as funcionalidades e capacidades do *testbed*, é feita uma implementação e testes do protocolo EDHOC e PQ-EDHOC.

EDHOC (*Ephemeral Diffie-Hellman Over COSE*) é um protocolo de segurança do IETF, desenvolvido no grupo LAKE (*Lightweight Authenticated Key Exchange*), voltado a dispositivos com recursos limitados e projetado como alternativa mais leve ao TLS 1.3/DTLS 1.3. Seu *handshake* é composto por três mensagens principais: a primeira enviada pelo iniciador com parâmetros e chave pública efêmera; a segunda enviada pelo

respondente com sua chave efêmera e autenticação; e a terceira enviada pelo iniciador com suas credenciais autenticadas [Selander et al. 2020, Selander et al. 2024].

PQ-EDHOC é uma variante do EDHOC com foco em resistência pós-quântica, substituindo primitivas assimétricas tradicionais por mecanismos de PQC. O esquema é semelhante, mas usa criptografia pós-quântica: a primeira mensagem envia a chave pública efêmera do KEM (*Key Encapsulation Mechanism*) do iniciador; na segunda, o respondente encapsula um segredo com essa chave, envia o texto cifrado resultante e se autentica com uma assinatura pós-quântica; por fim, a terceira contém os dados de autenticação pós-quântica do iniciador [Fraile et al. 2025]. Neste trabalho, o EDHOC original utiliza ECDSA com curva NIST P-256 e SHA-256, enquanto o PQ-EDHOC utiliza ML-KEM Nível 1 para estabelecimento de segredo e ML-DSA Nível 2 para autenticação.

4. Apresentação do *testbed*

O *testbed* desenvolvido é uma plataforma flexível e instrumentada para experimentação em redes veiculares sem fio. A partir da integração entre Mininet-WiFi e Containernet, os nós são representados como entidades independentes e configuráveis, capazes de executar *software* real, incluindo aplicações, serviços e protocolos de segurança. O ambiente permite a configuração programável dos experimentos, com ajuste de número de nós, posicionamento, mobilidade, parâmetros do canal, interferência e topologia. Com isso, é possível avaliar protocolos de segurança sob diferentes condições de conectividade e complexidade, incluindo cenários com enlaces *ad-hoc* e comunicação *multi-hop*.

A instrumentação inclui captura automática de tráfego em arquivos PCAP (*Packet Capture*), geração de *logs* e análise posterior em ferramentas como o Wireshark. Isso contribui para a observabilidade, depuração e validação dos resultados. Como limitação, o ambiente se restringe à emulação de redes iguais ou similares à Wi-Fi, não possuindo ainda características específicas de tecnologias celulares 4G, 5G ou 6G. Ainda assim, dentro desse escopo, o *testbed* oferece uma base adequada para experimentação em redes veiculares, combinando flexibilidade de configuração, execução de *software* real, como o controlador de voo ArduPilot [Baldi et al. 2022], avaliação de protocolos de segurança, escalabilidade e análise detalhada do comportamento da rede.

A Figura 1 apresenta a arquitetura geral do *testbed* proposto. A base da arquitetura é composta pelo *host* físico, responsável por fornecer o ambiente de execução com Linux, Docker e suporte à virtualização. Sobre essa base, o ambiente Containernet + Mininet-WiFi permite instanciar nós conteinerizados, como a GCS e os UAVs, executando *software* real e protocolos de segurança. Os enlaces sem fio emulados representam os canais de comunicação entre os nós, enquanto os contêineres adicionais indicam a possibilidade de expansão para múltiplos UAVs (ou outros componentes de redes veículares) e diferentes cenários experimentais. Por fim, a camada superior reúne os mecanismos de instrumentação e análise, com o Perf (para processamento), capturas PCAP, Wireshark, *logs* e *benchmarks*, utilizados para coletar e avaliar as métricas dos experimentos.

4.1. Arquitetura de um experimento

Esta seção descreve a configuração usada na avaliação inicial do *testbed*, considerando um cenário específico de autenticação em rede veicular aérea. Essa configuração representa uma instância da plataforma, validando a execução de protocolos reais e a coleta de métricas de desempenho. Foram utilizados dois contêineres: um representando

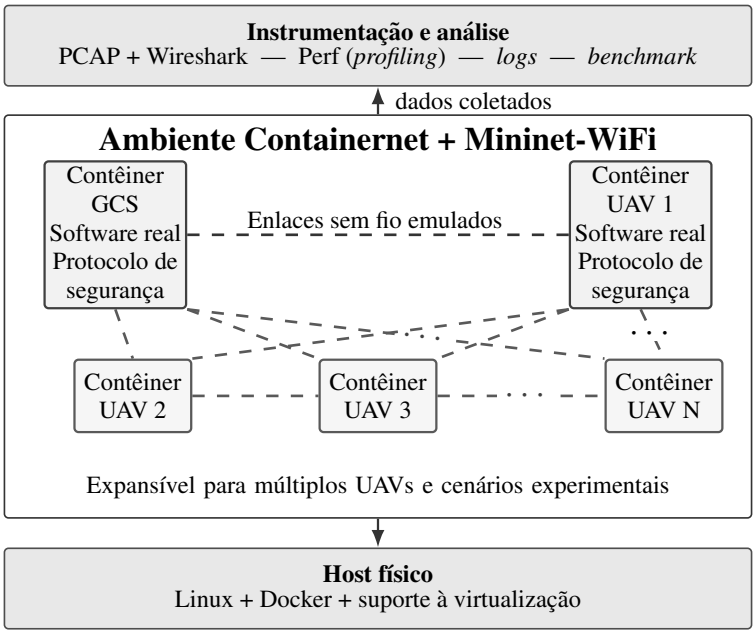

Figura 1. Arquitetura do *testbed* proposto, integrando Mininet-WiFi e Containernet para a emulação de enlaces sem fio e a execução de nós Docker.

a GCS, baseado em Ubuntu, e outro representando o UAV 1. Em ambos os contêineres, foram executadas as implementações do EDHOC e do PQ-EDHOC, permitindo avaliar o estabelecimento de sessão segura entre a GCS e o UAV. Além da execução isolada dos protocolos, também foram realizados testes com ArduPilot, a fim de observar o impacto da autenticação quando o UAV executa tarefas associadas a uma operação de voo.

No Mininet-WiFi, os enlaces foram configurados em modo *ad-hoc*, com *BATMAN-adv* para roteamento em camada 2 e modelo de propagação *log-distance* com expoente 3,5. As posições dos nós foram definidas no plano de simulação, permitindo controlar a distância entre a GCS e o drone e, assim, as condições do enlace sem fio. Esses parâmetros de rede podem ser configurados antecipadamente, a fim de criar diferentes cenários experimentais para avaliar o comportamento dos protocolos de segurança sob condições bem variadas. Os experimentos foram executados em uma máquina x86_64 com processador Intel Core i5-10210U, 4 núcleos físicos, 8 *threads*, frequência base de 1,60 GHz, *turbo* de até 4,20 GHz, 16 GB de RAM e suporte a Intel VT-x.

5. Resultados e discussões

A natureza emulada do *testbed* favorece a execução de medições comparativas e reprodutíveis sob condições controladas. Com isso, torna-se possível variar configurações, repetir cenários e observar tendências de desempenho antes de validações posteriores em ambientes reais.

A Tabela 2 apresenta alguns dos resultados observáveis no *testbed*, separando-os em tempo total de processamento, tempo estimado de comunicação e tempo total da execução fim a fim do protocolo. Para a medição do processamento foi usada a ferramenta Perf, e para a comunicação foi usada a ferramenta tcpdump para captura do tráfego e analisado o arquivo PCAP gerado no Wireshark com base nas capturas temporais. E para a estimativa total do *handshake* foi somado o tempo de processamento com o tempo

de comunicação. Para cada métrica, são apresentados a média, a mediana e o percentil 95 de 30 execuções. As condições com e sem ArduPilot representam, respectivamente, a execução simultânea ou não de operações de voo pelo ArduPilot durante a medição do processamento. Dessa forma, o *testbed* permite comparar os diferentes custos computacionais, de comunicação e totais dos protocolos sob diferentes condições operacionais.

Mais do que avaliar o EDHOC e o PQ-EDHOC, os resultados demonstram a capacidade do *testbed* de produzir dados experimentais em um ambiente controlado e reprodutível. A plataforma permite instanciar nós conteinerizados, variar a topologia e a quantidade de nós, configurar recursos computacionais e características dos enlaces. Além de capturar tráfego e registrar *logs*, o *testbed* pode calcular estatísticas básicas do sistema, indicando o desempenho geral, o comportamento típico e os 5% maiores picos para revelar os valores reais. Assim, o ambiente pode ser utilizado na avaliação de outros protocolos, aplicações e cenários envolvendo redes veiculares e comunicações sem fio emuladas. Embora os resultados não substituam experimentos em campo, a arquitetura de testes possibilita extensões como maior mobilidade, múltiplos UAVs e GCSs, enlaces com perdas e atrasos controlados, novos protocolos de segurança e execução distribuída, características de complexa avaliação na prática. Este trabalho está disponível em `https://github.com/regras/vnx-security-testbed`.

Tabela 2. Resultados do tempos de processamento e comunicação dos protocolos PQ-EDHOC e EDHOC.

Protocolo	ArduPilot	Processamento (ms)			Comunicação (ms)		
		Média	P95	Mediana	Média	P95	Mediana
PQ-EDHOC	Sem	4,1	4,5	4,0	3,3	6,1	2,9
	Com	4,3	4,7	4,2			
EDHOC	Sem	15,1	15,5	15,0	0,7	2,0	0,5
	Com	16,8	20,1	16,3			

6. Considerações finais

Este trabalho apresentou um *testbed* em desenvolvimento para experimentação em redes veiculares sem fio, integrando Mininet-WiFi e Containernet para emular comunicações e executar *software* real em nós conteinerizados. A avaliação inicial com UAS, GCS, ArduPilot e protocolos de segurança demonstrou a viabilidade da plataforma para coletar diferentes métricas. Como limitação, o ambiente ainda não reproduz com precisão tecnologias celulares como 5G ou 6G. Os trabalhos futuros incluem a ampliação da infraestrutura, com o aumento do número de nós, incorporação de novos protocolos e coleta de métricas adicionais, como consumo mais preciso de recursos, mobilidade e custo computacional.

Referências

Al-Mousa, A., Sababha, B. H., Al-Madi, N., Barghouthi, A., and Younisse, R. (2019). Utsim: A framework and simulator for uav air traffic integration, control, and communication. *International Journal of Advanced Robotic Systems*, 16(5):1729881419870937.

Al-Shareeda, M. A. and Manickam, S. (2023). A systematic literature review on security of vehicular ad-hoc network (vanet) based on veins framework. *IEEE Access*, 11:46218–46228.

Baldi, S., Sun, D., Xia, X., Zhou, G., and Liu, D. (2022). Ardupilot-based adaptive autopilot: Architecture and software-in-the-loop experiments. *IEEE Transactions on Aerospace and Electronic Systems*, 58(5):4473–4485.

Campanile, L., Gribaudo, M., Iacono, M., Marulli, F., and Mastroianni, M. (2020). Computer network simulation with ns-3: A systematic literature review. *Electronics*, 9(2):272.

Dholakiya, D., Kshirsagar, T., and Nayak, A. (2020). Survey of mininet challenges, opportunities, and application in software-defined network (sdn). In *International Conference on Information and Communication Technology for Intelligent Systems*, pages 213–221. Springer.

dos Reis Fontes, R. and Rothenberg, C. R. E. (2019). Emulando redes sem fio com mininet-wifi. Technical report, Edição do Autor.

Fraile, L. P., Tasopoulos, G., Koulamas, C., Zhao, R. K., Sultan, N. H., Regazzoni, F., and Fournaris, A. P. (2025). Enabling quantum-resistant edhoc: Design and performance evaluation. *IEEE Access*, 13:75861–75884.

Mustafovski, R. (2025). Evaluating the operational impact of secudronecomm: Simulation-based assessment of secure uav communication in military environments. *Scientific Technical Review*, 75(1):11–18.

Nguyen, H., Huda, S., Nogami, Y., and Nguyen, T. T. (2025). Security in post-quantum era: A comprehensive survey on lattice-based algorithms. *IEEE access*.

Peuster, M., Karl, H., and van Rossem, S. (2016). Medicine: Rapid prototyping of production-ready network services in multi-pop environments. In *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 148–153.

Selander, G., Mattsson, J., and Palombini, F. (2020). Ephemeral diffie-hellman over cose (edhoc). IETF 108, LAKE WG. draft-ietf-lake-edhoc-00.

Selander, G., Preuß Mattsson, J., and Palombini, F. (2024). Ephemeral Diffie-Hellman Over COSE (EDHOC). RFC 9528.

Tariq, U. and Ahanger, T. A. (2025). Enhancing intelligent transport systems through decentralized security frameworks in vehicle-to-everything networks. *World Electric Vehicle Journal*, 16(1):24.

Tropea, M., Aldana, C. E. Q., de Freitas, E. P., and De Rango, F. (2025). Sfem3: A performance comparative analysis for sdn-based fanet emulation using mininet-wifi and ns-3. *Ad Hoc Networks*, 177:103859.

Varga, A. and Hornig, R. (2008). An overview of the omnet++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, pages 1–10.

Veara, J., Jain, M., Moy, K., and Ranganathan, A. (2025). Tbrd: Tesla authenticated uas broadcast remote id. *arXiv preprint arXiv:2510.11343*.

Wang, J., Shao, Y., Ge, Y., and Yu, R. (2019). A survey of vehicle to everything (v2x) testing. *Sensors*, 19(2):334.