# Avaliação Experimental de Técnicas de Segurança em Comunicação de Borda

**Luka Alves Claro**[1], **Lailla Milainny Siqueira Bine**[1]

[1]Colegiado de Ciência da Computação – Universidade Estadual do Paraná (UNESPAR)
Apucarana – PR – Brasil

***Resumo.*** *Este trabalho apresenta uma avaliação experimental de segurança em ambientes de Computação de Borda, focando na vulnerabilidade do fluxo de dados entre dispositivos de borda e servidores. Foi desenvolvido um protótipo de controle de acesso veicular utilizando Raspberry Pi, submetido a ataques de interceptação (Man-in-the-Middle) e negação de serviço (DoS). Os resultados demonstram que a comunicação em texto claro é trivialmente explorável, enquanto a implementação de criptografia híbrida (AES+RSA) garante confidencialidade e integridade. Contudo, testes de estresse revelaram que a disponibilidade do dispositivo de borda permanece suscetível a ataques volumétricos, exigindo camadas adicionais de proteção.*

## 1. Introdução

A tecnologia avança de forma acelerada, impulsionando a automação nos mais diversos setores, sendo o controle de acesso veicular uma das áreas beneficiadas por essas inovações [4]. Com a crescente necessidade de soluções inteligentes e eficientes em ambientes como condomínios, empresas e shoppings, a automação tornou-se não apenas um meio de aumentar a agilidade do processamento de informações, mas uma necessidade para garantir a segurança e a integridade dos dados envolvidos. Diante desse cenário, a Computação de Borda surge como uma solução eficiente, cuja principal vantagem reside no pré-processamento de informações sensíveis próximo à fonte dos dados, antes do envio para a nuvem, o que minimiza riscos e reduz significativamente a latência [22]. Estudos estimam que uma parcela expressiva dos dados criados será processada fora dos data centers centralizados, isto é, nas bordas da rede, oferecendo respostas mais rápidas e maior controle [6].

No entanto, ao passo que soluciona questões de desempenho, a arquitetura distribuída introduz um novo e complexo desafio: a Segurança da Informação [5]. Com o aumento dos ciberataques que exploram vulnerabilidades em dispositivos de borda e meios de comunicação, a segurança torna-se um tema central. A tríade da segurança da informação, composta por Confidencialidade, Integridade e Disponibilidade, é o modelo fundamental para proteger dados em sistemas computacionais [2]. Isso é especialmente crítico ao lidar com dados sensíveis, como registros de entrada e saída e imagens capturadas por sistemas de monitoramento. Nesse contexto, tecnologias como câmeras IP e sistemas de reconhecimento de placas geram um grande volume de dados em tempo real [8]. A abordagem tradicional, que depende do envio integral dessas informações para a nuvem, enfrenta problemas como congestionamento de rede e atrasos na tomada de decisão. Por outro lado, a Computação de Borda, embora resolva a latência, expande a superfície de ataque. Dispositivos de borda, frequentemente localizados em ambientes fisicamente acessíveis, tornam-se alvos de interceptações e manipulações, comprometendo a operação de sistemas críticos e a privacidade dos dados.

A discussão central que impulsiona este projeto é a necessidade de conciliar os benefícios operacionais da Computação de Borda com os requisitos rigorosos de segurança. O argumento principal é que a simples migração do processamento para a borda é insuficiente, sendo essencial que essa migração seja acompanhada pela implementação de uma arquitetura de segurança específica para este modelo híbrido [3]. O problema principal abordado neste trabalho é a vulnerabilidade do fluxo de dados entre dispositivos de borda e servidores centrais, especialmente quando trafegam por redes locais suscetíveis a interceptações. Essa questão torna-se crítica no caso de uso adotado por esta pesquisa: um sistema de controle de acesso veicular, onde a ausência de mecanismos de segurança robustos no trânsito das informações pode comprometer a confidencialidade e a integridade de todo o sistema. Este trabalho se apoia no fato que, por meio da aplicação de técnicas de criptografia no fluxo de dados, é possível mitigar significativamente os riscos associados, construindo um sistema confiável e resiliente.

O objetivo geral deste trabalho é avaliar experimentalmente a segurança do fluxo de dados em um sistema de Computação de Borda, demonstrando vulnerabilidades práticas e validando a eficácia de uma solução de criptografia híbrida para mitigação de riscos. Os objetivos específicos incluem a construção de um protótipo funcional de controle de acesso veicular utilizando hardware de baixo custo como dispositivo de borda, a análise da vulnerabilidade da comunicação em um cenário inseguro por meio de interceptação passiva, a implementação de uma camada de criptografia híbrida para garantir a confidencialidade e integridade dos dados em trânsito, e a avaliação da resiliência do dispositivo de borda quanto à disponibilidade, submetendo-o a ataques de negação de serviço.

## 2. Trabalhos Relacionados

Diversos trabalhos abordam a intersecção entre Computação de Borda, IoT e Segurança, contextualizando a presente pesquisa. Cesar [1] propõe uma arquitetura de controle de acesso para IoT utilizando Edge Computing para melhorar a disponibilidade, focando em autenticação e políticas de acesso, mas não na criptografia do fluxo de dados. Silverio e Guardia [17] exploram o conceito de "Edge Security" por meio da filtragem de pacotes na borda, visando eficiência energética e proteção contra tráfego malicioso, uma abordagem complementar à criptografia. Kraus [9] valida experimentalmente a redução de latência proporcionada pela borda em redes 5G industriais, reforçando a motivação de desempenho que fundamenta este trabalho. Schenfeld [15] propõe uma arquitetura híbrida Fog/Edge e implementa segurança na comunicação via TLS/DTLS, uma abordagem próxima à deste trabalho, porém utilizando protocolos padrão em vez de uma implementação de criptografia híbrida customizada como defesa contra ataques MitM.

## 3. Desenvolvimento

A fundamentação teórica deste trabalho baseia-se nos conceitos de Computação em Nuvem e sua evolução para a Computação de Borda, bem como nos princípios de Segurança da Informação e Criptografia. A Computação em Nuvem representa um modelo de entrega de serviços computacionais pela internet, caracterizado pela elasticidade e escalabilidade [12]. Contudo, este modelo centralizado enfrenta limitações em cenários que exigem baixa latência [13]. A Computação de Borda surge para endereçar essas limitações, aproximando o processamento da fonte de dados [16]. No contexto de segurança, a

aplicação de técnicas criptográficas é essencial. A criptografia simétrica utiliza uma chave única para cifrar e decifrar, sendo eficiente para grandes volumes de dados, enquanto a criptografia assimétrica utiliza um par de chaves para troca segura de informações [11]. A solução padrão da indústria é a criptografia híbrida, que combina a segurança da assimétrica com a velocidade da simétrica [19].

A metodologia empregada classifica-se como aplicada e experimental. Para a realização dos experimentos, foi configurada uma infraestrutura de rede local composta por três componentes principais: um dispositivo de borda, representado por um Raspberry Pi 4 Model B operando em modo headless e equipado com webcam; um servidor central, executado em um Desktop PC com Ubuntu Server e Docker; e uma estação do atacante, utilizando um notebook com Arch Linux. A topologia de rede consistiu na conexão cabeada desses dispositivos a um mesmo switch, garantindo estabilidade para os testes. O desenvolvimento do software foi dividido em componentes de cliente e servidor. O cliente de borda, desenvolvido em Python, utilizou a biblioteca OpenCV para captura de vídeo e uma biblioteca de reconhecimento de placas refatorada para processamento local [10]. O servidor foi implementado com o framework Flask, conteinerizado via Docker, responsável por validar as placas recebidas.

Para facilitar a compreensão do roteiro metodológico adotado, a Figura 1 apresenta visualmente a sequência das etapas percorridas.

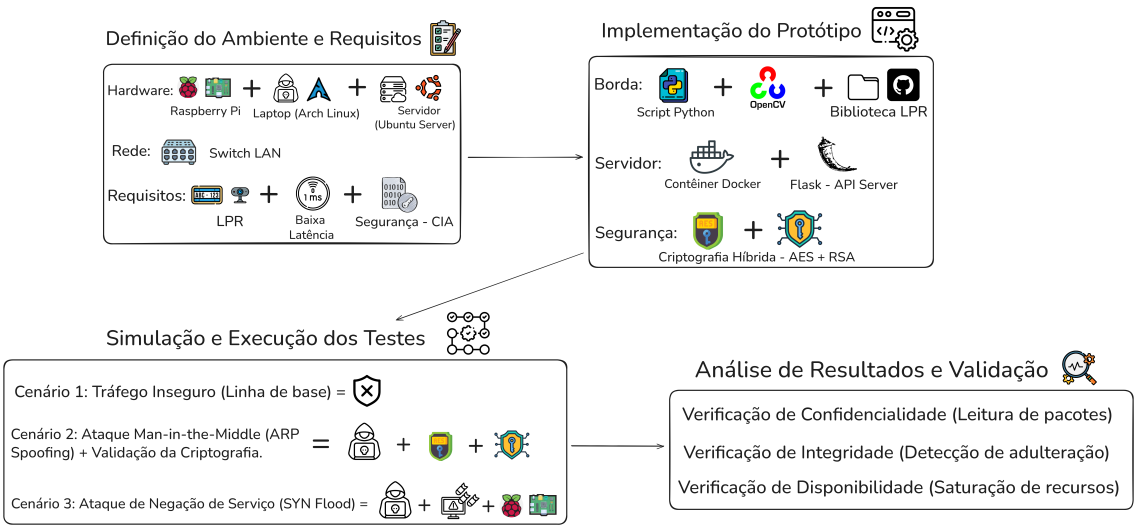


**Figura 1. Fluxograma das etapas do desenvolvimento da pesquisa**

Fonte: Imagem do Autor

Para mitigar as vulnerabilidades de interceptação, foi implementado um esquema de criptografia híbrida. O processo inicia-se com a geração de um par de chaves RSA no servidor. Durante a requisição, o cliente gera uma chave de sessão AES única, criptografa os dados da placa com essa chave e, em seguida, cifra a chave de sessão com a chave pública RSA do servidor. O servidor utiliza sua chave privada para recuperar a chave de sessão e decifrar os dados. A resposta segue o mesmo processo inverso, garantindo que todo o tráfego seja ilegível para terceiros.

## 4. Resultados

A avaliação da segurança foi realizada através da definição de três cenários experimentais. As ferramentas utilizadas para a execução e análise dos ataques incluíram o Wireshark [21] e tshark [20] para captura de pacotes, hping3 [7] para geração de tráfego de ataque DoS e arpspoof [18] para execução do envenenamento de cache ARP, viabilizando a interceptação. A análise dos resultados foi conduzida de forma qualitativa, observando o comportamento do sistema e a natureza dos dados capturados sob a ótica das métricas de Confidencialidade, Integridade e Disponibilidade.

No primeiro cenário experimental, o sistema foi executado sem criptografia. A captura de tráfego realizada pelo atacante revelou que os pacotes HTTP continham o payload JSON com os dados da placa em texto puro. Essa evidência comprovou a existência de uma falha crítica de confidencialidade, permitindo que qualquer agente na rede local tivesse acesso aos dados sensíveis. Além disso, a exposição em texto claro indicou uma vulnerabilidade de integridade, pois um atacante capaz de ler o formato dos dados poderia facilmente injetar pacotes adulterados no sistema utilizando ferramentas como o Scapy [14].

No segundo cenário, foi validada a eficácia da criptografia híbrida. O ataque Man-in-the-Middle foi executado utilizando ARP Spoofing para desviar o tráfego entre a borda e o servidor. A análise dos pacotes capturados demonstrou que, diferentemente do cenário anterior, o conteúdo das mensagens estava cifrado e codificado em Base64, tornando-se ininteligível para o atacante. A chave de sessão AES, protegida pela criptografia RSA, garantiu que apenas o servidor autorizado pudesse decifrar o conteúdo. Adicionalmente, o uso do modo de operação AES-GCM forneceu uma tag de autenticação, assegurando a integridade dos dados; qualquer tentativa de alteração no payload cifrado resultaria em falha na verificação da tag, levando ao descarte do pacote pelo servidor. Observou-se, contudo, um aumento no tamanho dos pacotes devido ao overhead dos metadados de segurança, um compromisso necessário para garantir a proteção.

O terceiro cenário focou na métrica de disponibilidade. O dispositivo de borda foi submetido a um ataque de negação de serviço do tipo SYN Flood. O monitoramento dos recursos do sistema evidenciou um aumento abrupto na carga da CPU e a saturação da pilha de rede do sistema operacional. Como consequência, a aplicação de borda tornou-se incapaz de estabelecer novas conexões com o servidor, resultando na interrupção do serviço de controle de acesso. Este resultado demonstrou que, embora a criptografia proteja os dados em trânsito, ela não mitiga riscos associados à exaustão de recursos em dispositivos com hardware limitado.

## 5. Considerações Finais

Este trabalho alcançou seu objetivo geral de avaliar experimentalmente a segurança em Computação de Borda, validando a eficácia da criptografia híbrida na proteção do fluxo de dados. Em relação aos objetivos específicos, o protótipo utilizando Raspberry Pi foi implementado com sucesso, permitindo a simulação realista de um cenário de controle de acesso. A análise de vulnerabilidade confirmou, por meio de interceptação passiva, a exposição crítica de dados em redes inseguras. A implementação da criptografia híbrida mitigou esse risco, garantindo a confidencialidade e integridade das informações, conforme demonstrado pela ilegibilidade dos pacotes capturados no cenário seguro.

Por fim, os testes de estresse demonstraram as limitações de disponibilidade do dispositivo de borda frente a ataques de negação de serviço. A execução do ataque SYN Flood foi capaz de paralisar a operação do sistema, evidenciando que a segurança em IoT exige uma abordagem em camadas que vá além da proteção de dados. Como limitações, este estudo focou na validação funcional da criptografia sem uma análise quantitativa exaustiva do overhead de desempenho e não implementou contramedidas ativas para o ataque de DoS.

Para trabalhos futuros, recomenda-se a realização de uma análise de desempenho detalhada para medir o impacto da latência introduzida pela criptografia, bem como a implementação e teste de mecanismos de mitigação de DoS diretamente na borda, utilizando ferramentas como iptables ou filtros de pacotes de alto desempenho. Conclui-se que a Computação de Borda exige uma estratégia de segurança deliberada e que a criptografia híbrida, embora não seja uma solução para todos os vetores de ataque, é um componente indispensável e eficaz para proteger a confidencialidade e a integridade dos dados em trânsito.

## Declaração sobre IA generativa

Este trabalho contou com o apoio de ferramentas de Inteligência Artificial generativa (Gemini, modelo 2.5, do Google), utilizadas para revisão linguística sob supervisão e validação do autor.

## Referências


[1] Rogerio Lopes Vieira Cesar. UMA ARQUITETURA DE CONTROLE DE ACESSO PARA INTERNET DAS COISAS. Dissertação (mestrado acadêmico em computação), Universidade Federal do Ceará, Quixadá, 2022.

[2] K S Couto et al. Os três pilares da segurança da informação na internet chinesa. *Journal of Technology & Information (JTnI)*, 2(2), 2022.

[3] Schahram Dustdar, V C Pujol, and P K Donta. On distributed computing continuum systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):4092–4105, 2022.

[4] Maria Augusta B Ferasoli. Detecção e reconhecimento de placas de automóveis para controle de acesso em condomínios residenciais por meio de visão computacional. 2017.

[5] Edison Luiz Goncalves Fontes. *Segurança da informação*. Saraiva Educação SA, 2017.

[6] B Gill and S Rao. Technology insight: Edge computing in support of the internet of things, 2017.

[7] Kali Linux. hping3. `https://www.kali.org/tools/hping3/`, 2025. Acesso em: 3 nov. 2025.

[8] M M Khan et al. License plate recognition methods employing neural networks. *IEEE Access*, 11:73613–73646, 2023.

[9] Dener Kraus. Computação de borda para indústria utilizando a rede 5G. Trabalho de conclusão de curso (graduação em engenharia de controle e automação), Universidade Federal de Santa Catarina, Blumenau, 2021.

[10] Gabriel Lima and Carlos Andrino. Reconhecimento e leitura de placa de carro (lpr) em python com opencv. `https://github.com/Gabriellimmaa/`

reconhecimento-e-leitura-placa-carro-ptBR, 2023. Acesso em: 8 nov. 2025.

[11] A A L Queiroz et al. Autenticação com suporte à computação de borda 5g para a internet de veículos. *Brazilian Journal of Development*, 9(5):14613–14631, 2023.

[12] M R Rahimi et al. Mobile cloud computing: A survey, state of art and future directions. *Mobile Networks and Applications*, 19(2):133–143, 2014.

[13] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[14] Scapy. Scapy: The python-based interactive packet manipulation program & library. https://scapy.net/, 2025. Acesso em: 8 nov. 2025.

[15] Matheus Crespi Schenfeld. FOG E EDGE COMPUTING: UMA ARQUITETURA HÍBRIDA EM UM AMBIENTE DE INTERNET DAS COISAS. Dissertação (mestre em ciência da computação), Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2017.

[16] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[17] Arthur Eugenio Silverio and Hélio Crestana Guardia. FILTRAGEM DE PACOTES NA BORDA DA REDE: UMA ANÁLISE COMPARATIVA COM FOCO NO CONSUMO DE ENERGIA. Relatório técnico, Universidade Federal de São Carlos (UFSCar), São Carlos, 2023.

[18] SMIKIMS. arpspoof: A simple arpspoofer in python. https://github.com/smikims/arpspoof, 2025. Acesso em: 3 nov. 2025.

[19] J Varella. Computação em névoa. *Revista EDUC-Faculdade de Duque de Caxias*, 6(1):31–46, 2019.

[20] Wireshark Foundation. Tshark(1) manual page. https://www.wireshark.org/docs/man-pages/tshark.html, 2025. Acesso em: 3 nov. 2025.

[21] Wireshark Foundation. Wireshark: Go deep. https://www.wireshark.org/, 2025. Acesso em: 3 nov. 2025.

[22] Fatma Yildirim, Yunus Yalman, Kamil Cagatay Bayindir, and Erman Terciyanli. Comprehensive review of edge computing for power systems: State of the art, architecture, and applications. *Applied Sciences*, 15(8), 2025.