APLICABILIDADE DA ENCRIPTAÇÃO AUTENTICADA NO PROTOCOLO CAN FD

APPLICABILITY OF AUTHENTICATED ENCRYPTION IN THE CAN FD PROTOCOL

Luan Phelippe Almeida Freire Santos[1], [i]
Fernando Simplicio de Sousa [2], [ii]
Luis Carlos Canno [3], [iii]
Leandro Poloni Dantas[4], [iv]

Data de submissão: (31/03/2025) Data de aprovação: (30/09/2025)

RESUMO

O protocolo de barramento CAN, desenvolvido pela empresa alemã Robert Bosch GmbH na década de 1980, tornou-se o padrão predominante para aplicações automotivas, permitindo a comunicação entre os diversos módulos que compõem os veículos. Projetado para oferecer alta confiabilidade e robustez no transporte de dados, o protocolo, entretanto, não considerou requisitos de segurança da informação devido às limitações e prioridades da época, o que levou à adoção da segurança por obscuridade pelos fabricantes. Com a crescente conectividade dos veículos no contexto atual, as vulnerabilidades relacionadas à integridade e confidencialidade dos dados tornaram-se mais evidentes, demandando soluções específicas. Este trabalho investiga a aplicabilidade de mecanismos de segurança da informação no protocolo CAN, com foco no CAN FD, uma evolução que possibilita campos de dados maiores e taxas de transmissão mais altas. O estudo avalia a eficácia e o impacto de medidas de segurança, por meio de testes realizados em um barramento simples com três nós: dois utilizam o algoritmo de encriptação autenticada AES-GCM, enquanto o terceiro realiza ataques de *eavesdropping* e *spoofing*. Para análise de desempenho, são coletados dados sobre a taxa de transmissão de mensagens e o tempo de execução dos processos de encriptação e decriptação. Os resultados demonstram que, embora os mecanismos de segurança implementados sejam eficazes na mitigação de riscos, sua aplicação reduz em 89,6% a taxa de transmissão de mensagens, indicando que a solução é adequada apenas para cenários que tolerem uma redução significativa na taxa de transmissão ou que disponham de maior capacidade computacional.

Palavras-chave: protocolo CAN FD; encriptação autenticada; AES-GCM.

[1] Pós-graduado em Sistemas Embarcados pelo Centro Universitário SENAI São Paulo – Campus Vila Mariana – “Anchieta”. E-mail: luan45000@gmail.com
[2] Professor Dr. no Centro Universitário SENAI São Paulo – Campus Vila Mariana – “Anchieta”. E-mail: fernando.simplicio@sp.senai.br
[3] Professor Especialista no Centro Universitário SENAI São Paulo – Campus Vila Mariana – “Anchieta”. E-mail: luis.canno@sp.senai.br
[4] Professor Dr. no Centro Universitário SENAI São Paulo – Campus Vila Mariana – “Anchieta”. E-mail: leandro.poloni@sp.senai.br.

ABSTRACT

The CAN bus protocol, developed by the German company Robert Bosch GmbH in the 1980s, has become the predominant standard for automotive applications, enabling communication between the various modules that compose vehicles. Designed to offer high reliability and robustness in data transmission, the protocol did not account for information security requirements due to the limitations and priorities of its time, leading manufacturers to rely on security through obscurity. With the growing connectivity of modern vehicles, vulnerabilities related to data integrity and confidentiality have become more apparent, requiring specific solutions. This study investigates the applicability of information security mechanisms to the CAN protocol, focusing on CAN FD, an evolution that allows for larger data fields and higher transmission rates. The research evaluates the effectiveness and impact of security measures through tests conducted on a simple bus with three nodes: two using the authenticated encryption algorithm AES-GCM and a third performing eavesdropping and spoofing attacks. For performance analysis, data is collected on message transmission rates and the processing time for encryption and decryption. The results show that while the implemented security mechanisms effectively mitigate risks, their application reduces message transmission rates by 89.6%, indicating that the solution is suitable only for scenarios that can tolerate a significant reduction in transmission rates or have access to higher computational power.

Keywords: CAN FD protocol; authenticated encryption; AES-GCM.

1 INTRODUÇÃO

Introduzido ao mercado na década de 1980, o protocolo de barramento CAN (*Controller Area Network*) foi desenvolvido pela empresa Robert Bosch GmbH em cooperação com a Intel, com foco em dar suporte a aplicações robustas de comunicação entre microcontroladores em veículos por conta da necessidade de reduzir a quantidade e a complexidade do cabeamento. O protocolo foi então padronizado através das normas ISO 11898-1 a ISO 11898-5 e desde então é o mais comum para aplicações automotivas, sendo utilizado amplamente também na aviação, em trens, em controles industriais e em aplicações militares (Lawrenz, 2013).

Entre os benefícios oferecidos pelo protocolo CAN estão a alta imunidade a interferências, a habilidade de autodiagnóstico, o reparo de erros e a capacidade de realizar arbitragem e priorização das mensagens (Torre, 2021).

1.1 Problema de pesquisa

Devido ao contexto tecnológico à época do desenvolvimento do protocolo CAN, foram implementados mecanismos que visam a integridade e a consistência no transporte dos dados, mas não foram levados em conta requisitos de segurança da informação. Atualmente, em um novo momento em que veículos têm se tornado cada vez mais conectados, impulsionados pela integração de diversas tecnologias embarcadas (visando, por exemplo,

conforto e entretenimento para os usuários), as vulnerabilidades se tornam mais evidentes já que a rede passa a estar mais exposta a ataques. Até agora, fabricantes têm adotado a abordagem de "segurança por obscuridade", mantendo sob sigilo os padrões proprietários utilizados nas aplicações, de forma que terceiros não consigam facilmente compreender e manipular o fluxo da rede (Buttigieg; Farrugia; Meli, 2017).

Existem estudos demonstrando que, pela falta de autenticação e encriptação, os veículos modernos que utilizam o protocolo CAN estão vulneráveis a ataques como *eavesdropping*, *spoofing* e *denial of service*. Isso possibilita o comprometimento da privacidade (sendo possível utilizar as informações que trafegam na rede para monitorar padrões de direção do motorista), permite a manipulação de informações exibidas ao condutor (como a exibição de valores falsos no painel de instrumentos) e a proposital inutilização de determinados módulos (o que pode causar desde falhas até acidentes) (Bozdal; Samie; Jennions, 2018).

Há casos noticiados de roubo de carros através da injeção de dados no barramento CAN, em que os malfeitores acessaram a ECU (*Electronic Control Unit*) responsável pela *smart key* do veículo através do cabeamento de rede do farol utilizando-se de uma ferramenta eletrônica desenvolvida justamente para este fim (Kovacs, 2023).

Diversos trabalhos abordam a chamada "engenharia reversa da CAN" (Huybrechts *et al.*, 2018) ou "*Car Hacking*" (Payne, 2019). Neles são conduzidos estudos visando interpretar e catalogar as mensagens que trafegam no barramento CAN de veículos diversos, logo, quebrando a segurança por obscuridade. As motivações por trás do desenvolvimento dessas soluções podem, a princípio, ser genuínas, como disponibilizar os dados para telemetria e gestão de frotas. Porém, isso também pode provocar como efeito colateral a exposição completa desses dados à entes mal-intencionados (Buscemi *et al.*, 2023).

1.2 Objetivos

A partir do problema de pesquisa delineado e da relevância do tema no contexto atual da segurança em sistemas automotivos, definem-se a seguir o objetivo geral e os objetivos específicos deste trabalho.

1.2.1 Objetivo Geral

Este trabalho de pesquisa tem como principal objetivo a avaliação da aplicabilidade de mecanismos de segurança da informação atuais no protocolo CAN FD.

1.2.2 Objetivos Específicos

- Implementar encriptação autenticada em um barramento CAN FD;
- Verificar a eficácia da implementação contra ataques em cenários simulados;
- Levantar dados estatísticos sobre o desempenho de um barramento CAN FD seguro em comparação a um inseguro.

1.3 Justificativa

Os aspectos que fundamentam a relevância deste estudo são:

- O impacto na segurança dos ocupantes e da estrutura viária;
- A discussão sobre a necessidade de modernização das redes de bordo automotivas.

2 REVISÃO DE LITERATURA

Esta seção apresenta os fundamentos teóricos que embasam o desenvolvimento da pesquisa. Inicialmente, descrevem-se as características do protocolo CAN e, em seguida, sua evolução para o padrão CAN FD. Na sequência, são abordados os conceitos de segurança da informação, contemplando princípios, ataques, serviços e mecanismos, para então estabelecer sua relação com o contexto automotivo. Por fim, analisam-se as vulnerabilidades específicas do protocolo CAN e algumas das soluções propostas na literatura, que servem de base para a avaliação realizada neste trabalho.

2.1 Características do protocolo CAN

O *Automotive Handbook* (Robert Bosch GmbH, 2022) define CAN como um protocolo de comunicação em barramentos que se estabeleceu como padrão para aplicações automotivas desde sua introdução em veículos motorizados em 1991. Opera com base no princípio multimestre com uma topologia linear conectando os diversos nós (chamados comumente de ECUs – *Electronic Control Units*) através de um par de fios que, a depender da aplicação, podem estar trançados ou não e, com menos frequência, ser reduzidos a um único fio. As vias são denominadas como “CAN *High*” e “CAN *Low*” ou abreviadas como “CAN H” e “CAN L”.

As taxas de transferência possíveis dividem o CAN em duas categorias, sendo a CAN-B, de baixa velocidade, que pode alcançar até 125 kbps, e a CAN-C, de alta velocidade que pode chegar até o máximo de 1 Mbps.

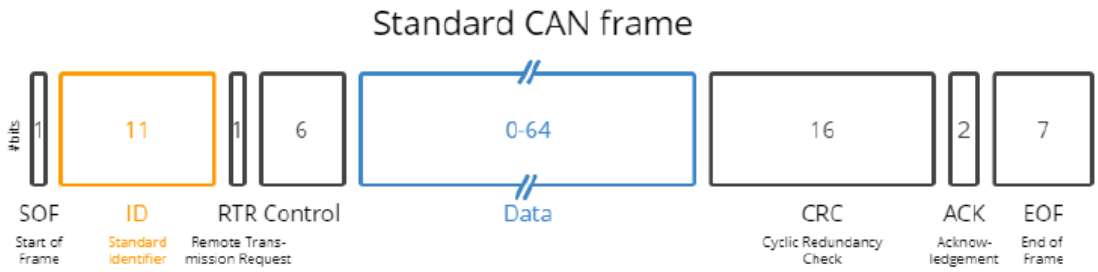
A CAN utiliza dois estados denominados “dominante” e “recessivo” para transmissão dos *bits* que codificam as informações, sendo que o estado dominante representa um *bit* de nível lógico baixo, e o estado recessivo representa um *bit* em nível lógico alto. Em valores nominais de tensão, num barramento de alta velocidade, o estado recessivo é representado por 2,5 V em ambos os fios; o estado recessivo configura 3,5 V e 1,5 V em CAN H e CAN L, respectivamente. Já em um barramento de baixa velocidade, para o estado recessivo haverá 0 V em CAN H e 5 V em CAN L; e para o estado dominante, 3,6 V em CAN H e 1,4 V em CAN L.

Essa configuração de níveis de tensão permite o funcionamento da arbitragem, uma das características da CAN que gerencia o acesso ao barramento, já que uma ECU que transmita um *bit* dominante irá sobrescrever quaisquer outras que estejam transmitindo um *bit* recessivo. Assim, um esquema de prioridade pode ser estabelecido baseando-se na mensagem a ser transmitida, ou seja, as mensagens iniciadas com a maior quantidade de *bits* recessivos possuem preferência para transmissão.

A Figura 1 representa a estrutura de um *frame* CAN padrão. Dois campos que são chave para compreender o funcionamento do protocolo são o campo de arbitragem e o campo de dados. Observa-se que o protocolo define um campo de 11 *bits* para identificação da mensagem. Isso implica que podem ser trafegadas até 2048 tipos de mensagens distintas. É

com base nesse campo que a disputa de arbitragem é resolvida, logo, mensagens com um número de identificação menor possuem maior prioridade. Há também o formato de *frame* estendido, em que os *bits* para identificação da mensagem são ampliados para 29, assim expandindo o número possível de mensagens. Já o campo de dados, de até 64 *bits* (8 *bytes*), é onde a ECU pode inserir os dados úteis que serão transmitidos na rede.

Figura 1 – *Frame* CAN padrão



Fonte: CSS Electronics (2025)

Ainda segundo o *Automotive Handbook* (Robert Bosch GmbH, 2022), outra característica importante do protocolo CAN é sua detecção de erros de transmissão através de diversos mecanismos, como o campo de CRC que compõe as mensagens, permitindo que cada ECU possa fazer a validação dos dados recebidos; a capacidade das ECUs de lerem a própria mensagem enquanto ela é transmitida, e comparar o *bit* escrito com o *bit* lido, podendo assim perceber falhas; o *Bit Stuffing*, que permite um número máximo de *bits* da mesma polaridade serem repetidos e determina a inserção de um *bit* de polaridade oposta quando esse número é excedido.

O tratamento dos erros detectados também é bastante robusto. Quando uma ECU detecta um erro, a transmissão é abortada e é enviada uma *flag* de erro para alertar a rede sobre a falha. As *flags* geradas são então contabilizadas estatisticamente, de forma que se um nó da rede excede um nível aceitável de erros ele é automaticamente desativado até que seja reinicializado.

2.2 CAN FD: uma evolução do CAN clássico

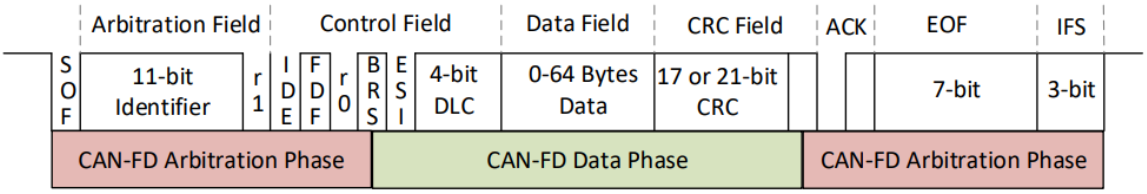
O aumento da complexidade dos sistemas automotivos mais modernos pode acabar sobrecarregando um barramento CAN. Para solucionar esse problema sem recorrer ao uso de múltiplos barramentos ou mudar completamente o protocolo usado, em 2012 a empresa Robert Bosch GmbH introduziu o CAN FD (*Controller Area Network with Flexible Data-Rate*), uma evolução do protocolo já muito bem estabelecido no mercado, implementando melhorias sem alterações na camada física (Hartwich, 2012; Robert Bosch GmbH, 2012).

Os principais pontos que careciam de melhora no CAN clássico são o tamanho do campo de dados que possui apenas 64 *bits* (8 *bytes*), e a taxa de transferência de dados que é de no máximo 1 Mbps. Para isso, foram implementadas modificações na estrutura do *frame* que permitem um campo de dados de até 512 *bits* (64 *bytes*), e taxas de transferência da ordem de Mbps (Hartwich, 2012).

O *frame* CAN FD é dividido em duas fases, como ilustra a Figura 2. A fase de arbitragem, no início e no final do *frame*, que possui os trechos de controle e em que a disputa pelo acesso

ao barramento é resolvida. A fase de dados é onde são transmitidas as informações úteis da mensagem. As taxas de transmissão para cada uma das fases podem manter-se as mesmas ou serem alteradas para uma taxa superior na fase de dados (Mutter, 2013).

Figura 2 – *Frame* CAN FD



Fonte: Joshi *et al*. (2017)

Assim, com a possibilidade de transferência mais rápida e uma quantidade de dados úteis por mensagem maior, o CAN FD permite ampliar os horizontes do protocolo ao permitir seu uso nas aplicações mais exigentes dos veículos modernos, que comportam cada vez mais sistemas eletrônicos, a possibilidade de atualização de ECUs através do barramento CAN FD, e a implementação de mecanismos de segurança (CSS Electronics, 2025).

2.3 Segurança da informação: princípios e ataques

A segurança da informação é definida como a proteção de dados e sistemas contra acesso, uso, publicação, perturbação, modificação ou destruição de forma não autorizada, com o objetivo de garantir sua confidencialidade, integridade e disponibilidade (Nieles; Dempsey; Pillitteri, 2017). Caracteriza-se assim a chamada tríade CIA (do acrônimo em inglês para *Confidentiality, Integrity and Availability*), conceito que evoca os três objetivos fundamentais da segurança da informação, cujas definições, segundo Stallings (2014), são:

> **Confidencialidade**: Preservar restrições autorizadas sobre acesso e divulgação de informação, incluindo meios para proteger a privacidade de indivíduos e informações privadas. Uma perda de confidencialidade seria a divulgação não autorizada de informação.
> **Integridade**: Prevenir-se contra a modificação ou destruição imprópria de informação, incluindo a irretratabilidade e autenticidade dela. Uma perda de integridade seria a modificação ou destruição não autorizada de informação.
> **Disponibilidade**: Assegurar acesso e uso rápido e confiável da informação. Uma perda de disponibilidade é a perda de acesso ou de uso da informação ou sistema de informação.

Ataques à segurança de um sistema, ou seja, qualquer ação que comprometa a segurança da informação, podem ser classificados como passivos (aqueles que tentam descobrir ou utilizar informações do sistema sem afetá-lo, como *sniffing* passivo e *eavesdropping*) ou ativos (aqueles que deliberadamente alteram o sistema ou a sua operação, como *sniffing* ativo, *spoofing* e *denial of service*) (Stallings, 2014).

A seguir, temos as definições dos ataques citados como exemplo, segundo Sinha; Rai; Bhushan (2019), e sua relação com a tríade CIA:

- *Sniffing*: escaneamento e captura de pacotes que estejam trafegando numa rede. Pode ser ativo ou passivo a depender do dispositivo de rede utilizado como ponto de acesso. Este ataque compromete especificamente a confidencialidade do sistema.
- *Eavesdropping*: escaneamento e captura do tráfego da rede de forma não autorizada com objetivo de extrair informações e/ou compreender o funcionamento da rede. É muitas vezes o ponto de partida para outros ataques. Assim como o *sniffing*, este ataque compromete a confidencialidade do sistema.
- *Spoofing*: é o ato de assumir a identidade de um nó da rede, com o objetivo de enganar os demais nós, assim podendo conseguir acessos ou injetar mensagens na rede como se fosse o nó original. O *spoofing* compromete a integridade do sistema.
- *Denial of service*: ataque com a intenção de comprometer completamente um nó ou toda a rede, deixando-os inacessíveis a usuários legítimos. Isso pode ser conseguido por diferentes métodos, como provocar uma sobrecarga do tráfego ou explorando mecanismos de contenção de falhas dos protocolos. O *DoS* compromete diretamente a disponibilidade do sistema.

2.4 Serviços e mecanismos de segurança

O ISO/IEC 27000 (2018), padrão para sistemas de gestão de segurança da informação, diz que a esta é alcançada através da implementação de controles que são definidos através de análises dos riscos do sistema, devendo ser especificados, implementados, monitorados, revistos e melhorados quando necessário para garantir que os requisitos de segurança da informação sejam sempre atendidos.

Recomendações como a X.800 (CCITT, 1991) listam serviços e mecanismos existentes para garantir a segurança adequada de sistemas e transferência de dados. Entre eles, destacam-se como relevantes para este estudo, conforme resumido por Stallings (2014):

- Mecanismos de codificação, que utilizam de algoritmos matemáticos para transformar o conteúdo de uma mensagem (chamado de “texto claro”, do inglês “*plaintext*”) em um formato que não seja facilmente inteligível (chamado de “texto cifrado”, do inglês “*ciphertext*”). O processo de transformar texto claro em texto cifrado é a encriptação, e o processo reverso é a decriptação. As técnicas para esses processos são amplamente estudadas na área da criptografia e visam, dentre outros objetivos, evitar ataques à confidencialidade, como *sniffing* e *eavesdropping*.
- Serviço de autenticação, cuja função é garantir ao destinatário que uma mensagem tem de fato a origem que afirma, que seu conteúdo não tenha sido alterado e opcionalmente sua sequência no tempo. Consiste em um valor agregado à mensagem que funciona como um autenticador, permitindo que um receptor possa determinar a validade da mensagem, e impedindo o sucesso de ataques à integridade, como o *spoofing*.

Conforme Abood; Guirguis (2018), os algoritmos existentes atualmente para encriptação e autenticação são classificados em três grupos (apud Alenezi; Alabdulrazzaq; Mohammad, 2020):

- Encriptação assimétrica: Utiliza um par de chaves pública/privada que são relacionadas matematicamente. Garantem um nível elevado de segurança, mas

possuem alto custo computacional. Um exemplo de cifra assimétrica é o conhecido RSA.

- Encriptação simétrica: Utiliza uma chave única comum e que é de conhecimento de todos os entes envolvidos na comunicação. Se subdivide em cifras de bloco (que trabalham os dados em blocos de tamanho fixo) e cifras de fluxo (que trabalham os dados em um *stream* de *bytes*). Demandam menor poder computacional em comparação com cifras assimétricas. Um exemplo de cifra simétrica é o conhecido AES.
- Funções *hash*: Algoritmos que permitem mapear uma entrada de tamanho qualquer para um bloco de tamanho fixo, o que é útil para armazenamento de senhas e verificação de integridade. Exemplos de funções *hash* são o MD5 e o SHA.

De forma similar às funções *hash*, o MAC (Código de Autenticação de Mensagem, do inglês *Message Authentication Code*) é um outro método que permite gerar um valor autenticador para mensagens, permitindo a verificação de integridade. Agrupa diversos algoritmos, como por exemplo o HMAC (MAC baseado em hash) e o CMAC (MAC baseado em cifra) (Stallings, 2014).

Ainda, existem esquemas de encriptação autenticada (AE, do inglês *Authenticated Encryption*), que permitem a proteção da confidencialidade e da integridade de forma simultânea. Segundo Zhang et al. (2018), uma das formas de se conseguir encriptação autenticada é combinando algoritmos que realizam cada uma dessas etapas separadamente. Assim, são possíveis as seguintes abordagens:

- *Encrypt-then-MAC*: O *plaintext* é encriptado e depois o *ciphertext* resultante é usado como base para geração do valor autenticador.
- *Encrypt-and-MAC*: O *plaintext* é encriptado e usado para geração do valor autenticador.
- *MAC-then-encrypt*: O *plaintext* é usado para geração do valor autenticador e este é encriptado juntamente com o *plaintext*.

As abordagens diferem na ordem em que as etapas são realizadas, e o estudo considera que *Encrypt-then-MAC* é superior em quesitos gerais de segurança em relação às demais, que possuem algumas vulnerabilidades.

2.5 Análise das vulnerabilidades do protocolo CAN e possíveis soluções

Os ataques a redes CAN documentados e noticiados são principalmente de *eavesdropping*, injeção de dados e *denial of service*, conforme levantado por Bozdal; Samie; Jennions (2018).

O *eavesdropping* num barramento CAN ameaça a confidencialidade, e visa principalmente acessar e compreender o tráfego da rede. As informações assim adquiridas podem ser exploradas para diversos fins através de engenharia reversa, objetivando a simples leitura e o consumo dos dados para gestão de frotas ou até mesmo a fundamentação de outros ataques à rede. Para mitigar esses riscos, podem ser utilizados mecanismos de codificação para tornar os dados trafegados ininteligíveis a qualquer um que consiga visualizá-los sem possuir a chave criptográfica. No contexto da eletrônica embarcada, em que os microcontroladores têm baixo poder de processamento em comparação com

microprocessadores utilizados em outras aplicações, entende-se como ideal o uso da encriptação simétrica.

A injeção de dados num barramento CAN consiste em forjar mensagens, num processo de roubo de identidade, ou *spoofing*, o que ameaça a integridade do sistema. Mensagens maliciosas injetadas na rede podem ter o objetivo de mascarar ou provocar falhas, exibir dados incorretos ou modificar o comportamento das ECUs, por exemplo. Para mitigar esses riscos, pode ser utilizado um serviço de autenticação, em que um valor autenticador é agregado às mensagens de forma que possa ser verificado pelos nós que irão recebê-la, permitindo que possam distinguir mensagens legítimas de mensagens maliciosas.

Já o *denial of service* num barramento CAN pode ser realizado de formas distintas, como, por exemplo, inundar a rede com mensagens repetitivas de forma que outros nós não consigam enviar suas mensagens, ou a injeção de *bits* durante a transmissão de um nó legítimo para provocar erros de escrita até que o controle de erros estatísticos do protocolo desabilite o nó em questão. Por explorar características específicas da CAN, a simples implementação de mecanismos de segurança tradicionais pode não ser efetiva, portanto o *denial of service* não será abordado neste trabalho.

3 METODOLOGIA

Tendo em vista as vulnerabilidades das redes CAN de veículos automotivos a ataques à confidencialidade e integridade, este trabalho explora o uso da encriptação autenticada para investigar a eficácia na mitigação de tais vulnerabilidades e o desempenho de uma rede segura em comparação a uma rede insegura. Especificamente, visa-se mitigar a possibilidade dos ataques de *eavesdropping* e *spoofing* utilizando o algoritmo de encriptação autenticada AES-GCM, que implementa a abordagem *Encrypt-then-MAC*. Sendo um algoritmo de encriptação simétrica, considera-se que a chave criptográfica fora previamente gerada e é de conhecimento dos nós legítimos.

Como um barramento CAN depende dos campos de controle das mensagens para realizar a arbitragem, o único trecho em que a autenticação encriptada será aplicada é no campo de dados. Optou-se pelo uso do protocolo CAN FD pela vantagem de possuir um campo de dados maior (de até 64 *bytes*), pois a autenticação exige que alguns *bytes* sejam utilizados para a transmissão do valor autenticador, e a encriptação exige a transmissão do vetor de inicialização. Isso prejudicaria o espaço disponível para dados úteis no CAN Clássico que dispõe de apenas 8 *bytes*.

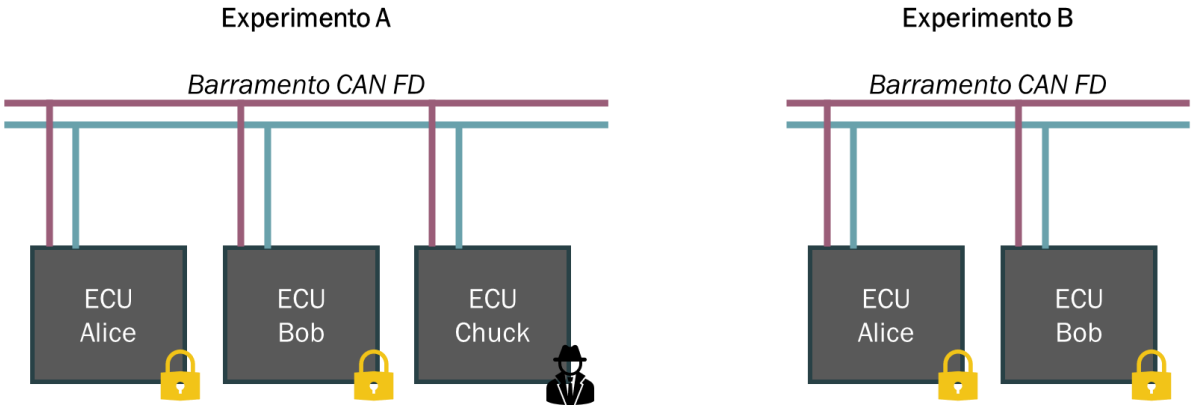
Foram realizados dois experimentos:

- Experimento A: validação da mitigação dos ataques, comparando dois cenários em que a encriptação autenticada foi utilizada, e outros dois em que ela foi desabilitada.
- Experimento B: comparativo de estatísticas de processamento e da taxa de transmissão de mensagens com e sem o uso da encriptação autenticada.

Para a realização dos experimentos, criou-se a disposição ilustrada pela Figura 3, em que duas ECUs (“Alice” e “Bob”) configuram um barramento CAN FD sendo nós legítimos da rede e compartilham a chave criptográfica. Apenas no Experimento A, uma terceira ECU (“Chuck”) estará presente como um nó malicioso, que também está conectado fisicamente ao barramento, mas não possui a chave criptográfica. Todas as ECUs são constituídas de um

microcontrolador STM32G431, com um *clock* de 16 MHz, e um *transceiver* CAN FD TCAN1044VDRQ1. A implementação do algoritmo AES-GCM é realizada por *software* através da biblioteca criptográfica STM32, certificada pelo *Cryptographic Algorithm Validation Program*, (2024).

Figura 3 – Topologias dos experimentos



Fonte: elaborado pelo autor (2025)

Para ambos os experimentos, nas mensagens não criptografadas, o comprimento do campo de dados é de 20 *bytes*. Nas mensagens criptografadas, além dos 20 *bytes*, somam-se 16 *bytes* que constituem o valor autenticador, e mais 12 *bytes* do vetor de inicialização, totalizando 48 *bytes*. Em qualquer caso, *bytes* não ocupados por dados úteis são preenchidos com o valor 0xFF.

3.1 Experimento A

No Experimento A, o *bit rate* configurado é de 500 kbps tanto para os campos de controle quanto para os campos de dados, um valor médio típico para barramentos CAN de alta velocidade.

A ECU Alice simula valores pseudoaleatórios, variando randomicamente com base numa função senoidal, mas programados para assumirem valores semelhantes às variáveis reais. Ela os envia em um conjunto de mensagens a Bob através do barramento CAN FD. As mensagens enviadas seguem as especificações indicadas no Quadro 1 e sua estrutura (comprimento dos dados, intervalos, fatores de conversão e unidades de medida) é baseada no *FMS Standard* (2017). A ECU Bob recebe e processa as mensagens, replicando os valores recebidos via UART de forma legível e análoga ao painel de instrumentos de um veículo. A ECU Chuck realiza ataques ao barramento em quatro diferentes cenários (A1 a A4), conforme o Quadro 2.

Nos cenários A1 e A2, são observadas as mensagens capturadas por Chuck, cujo conteúdo do campo de dados é encaminhado para a interface UART, e é analisada a inteligibilidade das mensagens para definir se ocorreu a quebra da confidencialidade.

Nos cenários A3 e A4, Chuck injeta valores falsos de velocidade do motor no barramento enviando mensagens com conteúdo vazio (todos os *bytes* do campo de dados têm valor 0xFF), o que deve resultar em um valor discrepante e constante, diferente dos dados legítimos que são variáveis. Nestes cenários, são analisadas as mensagens lidas por Bob,

verificando se os valores maliciosos são considerados ou descartados para definir se há quebra da integridade.

Quadro 1 – Configuração das mensagens do Experimento A

Variável	ID da mensagem	Intervalo de atualização	Intervalo de envio	Posição no campo de dados	Unidade de medida	Fator de conversão	Faixa de valores simulados
Velocidade do motor	0x006F	25 ms	25 ms	*Bytes* 4 a 5	RPM	0,125/*bit* 0 *offset*	800 a 7000
Velocidade do veículo	0x014D	1 s	100 ms	*Bytes* 6 a 7	km/h	1/256/*bit* 0 *offset*	0 a 120
Temperatura do líquido de arrefecimento	0x0309	10 s	1 s	*Byte* 7	°C	1/*bit* -40 *offset*	60 a 100
Nível do tanque de combustível	0x03E7	60 s	1 s	*Byte* 1	%	0,4/*bit* 0 *offset*	20 a 100
Odômetro	0x07B5	10 s	1 s	*Bytes* 0 a 3	m	5/*bit* 0 *offset*	0 a 10^9

Fonte: elaborado pelo autor (2025)

Quadro 2 – Cenários do Experimento A

	A1	A2	A3	A4
Ataque	*Eavesdropping*	*Eavesdropping*	*Spoofing*	*Spoofing*
Encriptação autenticada	Não	Sim	Não	Sim

Fonte: elaborado pelo autor (2025)

3.2 Experimento B

No Experimento B, o *bit rate* configurado é de 2 Mbps tanto para os campos de controle quanto para os campos de dados, um valor possível somente em barramentos CAN FD, escolhido para reduzir os tempos de transmissão e melhor evidenciar impactos provenientes do processamento da encriptação autenticada.

A ECU Alice envia uma mesma mensagem ininterruptamente através da rede, contendo um contador incremental e um *timestamp* em microssegundos. Simultaneamente, é medido o tempo gasto na etapa de encriptação em ciclos de *clock*. A ECU Bob lê as mensagens e as registra em um *log* interno. Simultaneamente, é medido o tempo gasto na etapa de decriptação em ciclos de *clock*.

Posteriormente, os valores dos *logs* são resgatados, permitindo sua análise para o levantamento das estatísticas:

Taxa de transmissão (R) das mensagens, em mensagens por segundo, através da relação (1):

$$R = \frac{N}{\left(\frac{(T_N - T_0)}{10^6}\right)} \quad (1)$$

Onde N é a quantidade de mensagens registradas no *log*, T_0 é o *timestamp* da primeira mensagem e T_N é o *timestamp* da última mensagem.

Intervalo médio ($\bar{I}$) entre mensagens, em microssegundos, dado por (2):

$$\bar{I} = \frac{1}{N-1}\sum_{i=1}^{N-1}(T_{i+1} - T_i) \quad (2)$$

Onde N é a quantidade de mensagens registradas no *log*, e T_i é o timestamp da i-ésima mensagem.

Tempos médios gastos na encriptação e na decriptação ($\bar{D}_{enc}$ e $\bar{D}_{dec}$), ciclos de *clock*, dados por (3):

$$\bar{D} = \frac{1}{N}\sum_{i=1}^{N}\frac{D_i}{\left(\frac{C}{10^6}\right)} \quad (3)$$

Onde D é a duração das etapas de encriptação ou decriptação em ciclos de *clock*.

Também é apresentado o tempo médio gasto nas etapas em microssegundos, dado por (4):

$$E = \frac{\bar{D}}{\left(\frac{C}{10^6}\right)} \quad (4)$$

Onde C é o valor de *clock* utilizado, em Hertz. Para este experimento, o valor é de 16 MHz.

Tais estatísticas foram coletadas em três cenários que diferem no comprimento do campo de dados e no uso ou não da encriptação autenticada, conforme descrito na Tabela 3.

Tabela 3 – Cenários do Experimento B

	B1	**B2**	**B3**
Encriptação autenticada	Não	Sim	Não
Comprimento do campo de dados (*bytes*)	20	48	48

Fonte: elaborado pelo autor (2025)

O cenário B1 representa uma aplicação sem nenhum mecanismo de segurança, portanto seu campo de dados é menor, com apenas 20 *bytes*. Já o cenário B2 implementa encriptação autenticada, e por isso possui um campo de dados de 48 *bytes*. O cenário B3 simula um campo de dados também de 48 *bytes*, porém sem o uso da encriptação autenticada, para efeitos de comparação do impacto apenas dos processos de encriptação e decriptação.

Por fim, as taxas de transmissão dos diferentes cenários são comparadas para chegar ao impacto percentual devido à aplicação da encriptação autenticada.

4 RESULTADOS E DISCUSSÕES

Esta seção apresenta os resultados obtidos a partir das simulações realizadas, bem como a discussão de seus impactos, iniciando-se com o Experimento A (cenários A1 a A4), nos quais se avaliam ataques de *eavesdropping* e *spoofing* em barramentos seguros e inseguros,

e seguindo-se com o Experimento B, que reúne estatísticas de transmissão relacionadas ao desempenho do barramento.

4.1 Cenário A1: *Eavesdropping* em um barramento inseguro

O Quadro 4 mostra um trecho do *log* de mensagens lidas por Chuck, estando inserido no barramento CAN FD enquanto Alice as transmite em texto plano (sem encriptação). É possível distinguir claramente quais dos 20 *bytes* possuem informações úteis e quais não são utilizados.

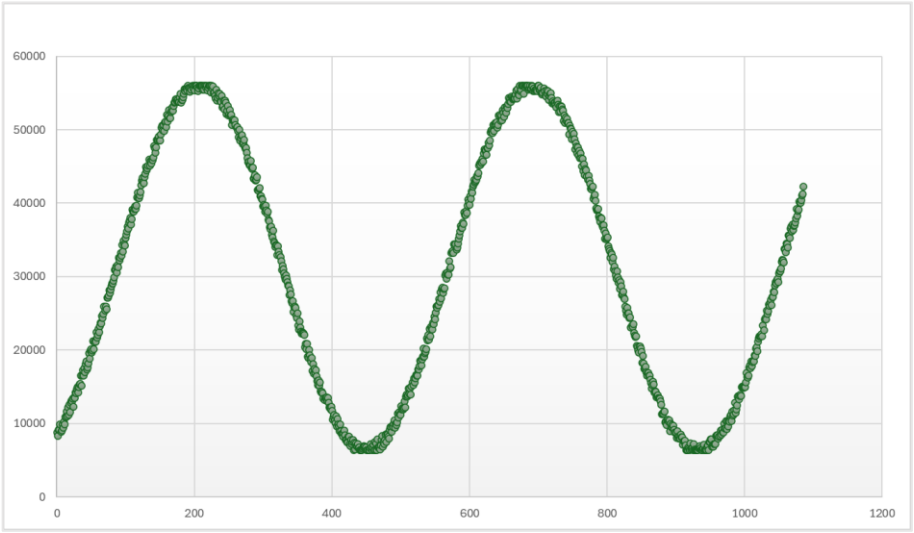
Neste cenário, a partir das mensagens coletadas através do ataque de *eavesdropping*, é possível delimitar a posição dos diferentes dados e traçar seu comportamento, eventualmente possibilitando a interpretação e catalogação dos dados por métodos de engenharia reversa. O gráfico na Figura 4 exibe uma plotagem da evolução dos valores contidos nos *bytes* 4 e 5 da mensagem 0x006F, evidenciando o padrão inteligível da função senoidal simulada por Alice.

Quadro 4 – Cenário B1: Trecho de *log* coletado por Chuck

ID	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18	B19
006F	FF	FF	FF	FF	30	22	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	5C	20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
014D	FF	FF	FF	FF	FF	FF	79	6E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	DB	23	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	96	23	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	B3	26	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
014D	FF	FF	FF	FF	FF	FF	79	6E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	A4	22	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	28	23	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	A6	26	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
014D	FF	FF	FF	FF	FF	FF	79	6E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	F0	24	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	FC	27	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	7A	26	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0309	FF	FF	FF	FF	FF	FF	FF	85	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
03E7	FF	73	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
07B5	BE	08	52	01	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	75	2A	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
014D	FF	FF	FF	FF	FF	FF	B3	71	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
006F	FF	FF	FF	FF	62	2A	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Fonte: elaborado pelo autor (2025)

Figura 4 – Cenário B1: Visualização dos valores dos *bytes* 4 e 5 da mensagem 0x006F



Fonte: elaborado pelo autor (2025)

4.2 Cenário A2: *Eavesdropping* em um barramento seguro

O Quadro 5 mostra um trecho do *log* de mensagens lidas por Chuck, estando inserido no barramento CAN FD enquanto Alice as transmite em texto cifrado (com encriptação). Desta vez, já não é possível notar os conjuntos de *bytes* que comporiam uma informação, e nem mesmo quais dos *bytes* não são utilizados. Neste cenário, um atacante que tenha a intenção de analisar as mensagens fica impossibilitado de fazê-lo, o que demonstra a mitigação do ataque de *eavesdropping* e a preservação da confidencialidade através da encriptação.

Quadro 5 – Cenário B2: Trecho de *log* coletado por Chuck

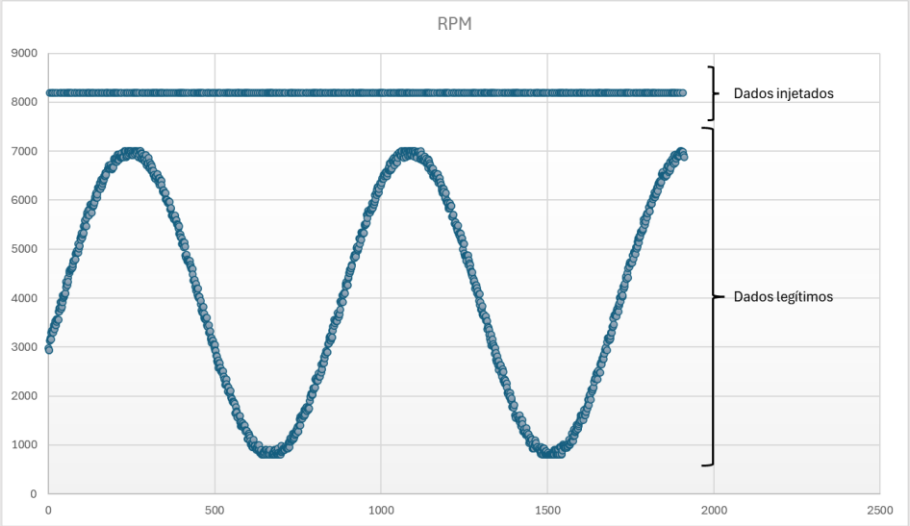
ID	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18	B19
006F	B5	41	0D	AE	94	86	89	09	21	46	15	2C	D8	30	E9	72	39	6C	2B	7D
006F	7A	19	28	2D	0F	0B	90	13	AE	96	C3	D5	91	87	BD	15	B6	BF	3F	DE
014D	F9	67	D2	C5	C2	85	CA	60	44	D7	CD	FE	01	B1	A4	B7	58	94	67	74
006F	6F	09	FA	E6	FC	FA	60	B1	4A	EF	E6	B1	38	C9	4D	80	29	DD	52	63
006F	08	F2	5D	04	D8	FD	26	42	36	57	F9	29	C7	8E	B7	B8	F0	44	C4	B8
006F	73	52	C4	CC	83	9B	97	9A	48	CE	15	B8	19	25	A3	1A	26	3C	32	36
014D	67	DE	AA	86	32	3A	8B	F7	A5	FD	AB	02	CB	6E	0E	96	1E	82	6F	05
006F	B3	9A	3E	3C	E9	62	99	F9	11	87	22	C1	DC	C5	6A	3E	DC	DD	70	C0
006F	FE	8F	76	BC	D1	12	D5	C0	7F	80	44	D8	58	AC	47	63	6E	4A	2E	85
006F	6D	3A	10	B2	E2	D7	25	29	72	22	59	BE	39	E8	2E	89	CC	F6	05	ED
014D	FB	2D	2D	EF	84	E3	AF	6A	C2	52	8B	94	E8	3F	45	4F	56	BE	3C	A2
006F	97	F8	AD	9C	24	4C	9A	10	C1	E3	24	92	AF	17	00	90	6C	0E	DA	8B
006F	0D	8C	F2	13	03	C9	6C	E8	44	51	A6	4D	8D	92	E1	F6	BB	51	12	6B
006F	3B	23	BB	C2	D1	0C	2D	BD	09	BC	C0	79	24	2A	1C	32	B3	67	F0	C7
014D	31	38	8A	E6	99	E5	75	30	F0	DB	A6	49	DF	C8	AD	7F	0B	42	56	B7
006F	B9	0B	90	F2	B8	E3	83	FF	8E	34	DD	3B	33	AC	FF	28	AE	7B	B9	67
006F	DF	FD	BC	09	DE	9D	A6	C4	85	DE	ED	58	7D	A3	68	6B	A3	35	46	8B
006F	A7	2A	30	6C	94	6F	AD	15	DB	E2	BB	D9	75	C7	B2	53	7A	8E	AA	28
014D	2C	CC	68	A6	64	41	FF	58	4C	F5	3E	FC	BB	27	7D	EC	4E	6C	37	96
0309	9D	05	EC	C9	11	20	1D	28	DF	41	27	89	0B	9E	11	D2	98	2C	D0	67

Fonte: elaborado pelo autor (2025)

4.3 Cenário A3: *Spoofing* em um barramento inseguro

A Figura 5 mostra uma plotagem dos valores de velocidade do motor em RPM lidos por Bob. É possível verificar que, além da curva senoidal esperada, há valores discrepantes, que são dados falsos injetados por Chuck (mensagens vazias que acarretam valores fora da faixa esperada). Neste cenário, o ataque de *spoofing* é bem-sucedido já que para Bob não há formas de distinguir as mensagens legítimas enviadas por Alice das mensagens maliciosas enviadas por Chuck.

Figura 5 – Cenário B3: Velocidade do motor lida por Bob

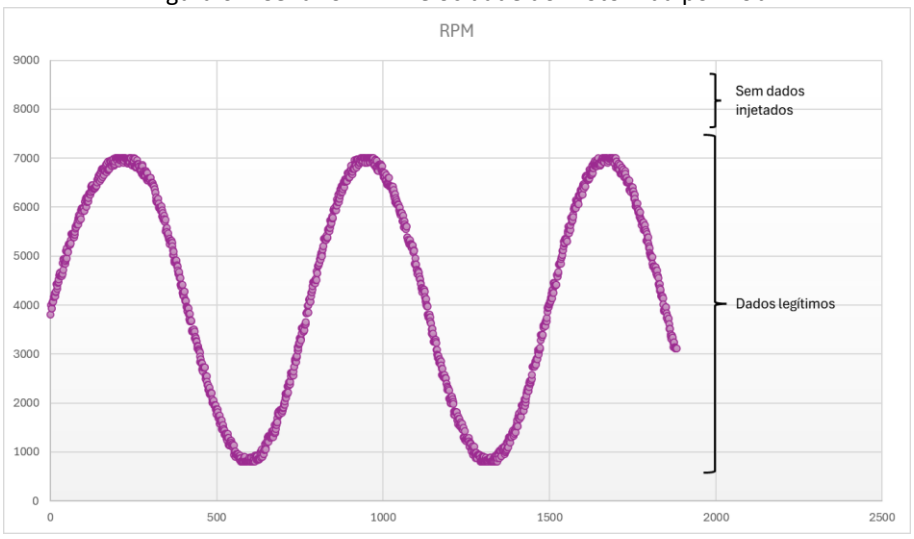


Fonte: elaborado pelo autor (2025)

4.4 Cenário A4: *Spoofing* em um barramento seguro

A Figura 6 mostra uma plotagem dos valores de velocidade do motor em RPM lidos por Bob. É possível verificar que neste cenário em que há autenticação, Bob consegue filtrar apenas as mensagens legítimas de Alice, portanto não estão presentes os valores discrepantes, que seriam os dados injetados por Chuck, o que demonstra que o ataque de *spoofing* foi mitigado, preservando a integridade através da autenticação das mensagens.

Figura 6 – Cenário B4 – Velocidade do motor lida por Bob



Fonte: elaborado pelo autor (2025)

4.5 Experimento B: Estatísticas de transmissão

A partir dos *logs* coletados por Alice e Bob, e da análise estatística das mensagens recebidas por Bob, foram levantados os resultados apresentados no Quadro 6.

Quadro 6 – Resultados do Experimento B

			B1	B2	B3
Parâmetros do cenário		Encriptação autenticada	Não	Sim	Não
		Comprimento do campo de dados (*bytes*)	20	48	48
Dados coletados	$T_N - T_0$	Tempo de coleta do *log* (µs)	186337	1792013	263856
	N	Quantidade de mensagens coletadas	999	999	999
	D_{enc}	Tempo médio de encriptação (ciclos de *clock*)	N/A	25953,8	N/A
	D_{dec}	Tempo médio de decriptação (ciclos de *clock*)	N/A	25694,4	N/A
Estatísticas calculadas	R	Taxa de mensagens processadas (mensagens/s)	5361,3	557,5	3786,2
	$\bar{I}$	Intervalo médio entre mensagens (µs)	186,5	1793,8	264,1
	E_{enc}	Tempo médio de encriptação a 16MHz (µs)	N/A	1625,5	N/A
	E_{dec}	Tempo médio de decriptação a 16MHz (µs)	N/A	1607,4	N/A

Fonte: elaborado pelo autor (2025)

Os resultados da comparação entre os cenários B1 e B2 indicam que ao introduzir a encriptação autenticada, ocorre uma queda de 89,6% da taxa de envio de mensagens. O impacto é proveniente tanto dos processos de encriptação e decriptação, quanto do aumento da quantidade de *bytes* a serem transmitidos.

Já na comparação dos cenários B2 e B3, em que os campos de dados das mensagens possuem o mesmo comprimento, a queda foi de 85,3%, o que demonstra que o maior impacto é devido ao processamento da encriptação autenticada.

5 CONCLUSÃO

O presente trabalho realizou a verificação da aplicabilidade e da eficácia do algoritmo AES-GCM em redes CAN FD contra os ataques de *eavesdropping* e *spoofing*, aos quais as redes automotivas estão sujeitas atualmente devido à falta de mecanismos de segurança da informação.

Os testes realizados no Experimento A mostraram que a implementação da autenticação encriptada AES-GCM é possível e eficaz contra os ataques testados. Nos cenários em que foi realizado o ataque de *eavesdropping*, ficou demonstrada a preservação da confidencialidade através da encriptação, ao impedir que um nó ilegítimo que não possua a chave criptográfica consiga dados inteligíveis ao espionar a rede. Nos cenários em que foi realizado o ataque de *spoofing*, ficou demonstrada a preservação da integridade das mensagens através da autenticação, ao permitir que os nós legítimos da rede identificassem e descartassem mensagens maliciosas.

Apesar do sucesso da implementação e da mitigação dos ataques, através das medições realizadas no Experimento B verificou-se que o impacto de agregar a encriptação autenticada ao protocolo CAN FD é consideravelmente grande, reduzindo a taxa de transmissão de mensagens em 89,6% em relação a uma aplicação não segura. Isso ocorre devido à adição dos *bytes* necessários para a transmissão do valor autenticador e do vetor de inicialização, além do custo computacional dos processos de encriptação e decriptação. Este último fator é o mais determinante, visto que em comparação a um cenário em que a mesma quantidade de *bytes* é transmitida, porém sem encriptação autenticada, a redução da taxa de transmissão ainda é de 85,3%.

A partir destes dados, pode-se concluir que a encriptação autenticada é uma solução para os ataques de *eavesdropping* e *spoofing*, desde que a aplicação possa dispor de uma taxa de transmissão consideravelmente menor, já que aplicações típicas, compostas por algumas dezenas de nós e com a necessidade de propagação das mensagens em tempo real para sistemas críticos dos veículos, podem ser prejudicadas.

Também foram levantados dados sobre os tempos gastos nos processos de encriptação e decriptação num microcontrolador de 16 MHz. Em investigações futuras visando reduzir tal impacto, podem ser exploradas alternativas como o aumento do poder de processamento, já que com um *clock* mais alto os tempos de encriptação de decriptação podem ser reduzidos; o uso de implementações mais leves e eficientes de bibliotecas criptográficas, como o IOVCipher, desenvolvido e abordado no trabalho de Huang et al., (2024); ou o uso de hardware criptográfico especializado dos microcontroladores que possuam tal funcionalidade, como o STM32G441, permitindo melhor desempenho nas etapas de encriptação e decriptação.

Levando em conta o contexto atual da tecnologia automotiva, há de se considerar uma modernização dos protocolos de rede automotivos e de projeto das ECUs, de forma a garantir poder computacional suficiente para garantir a segurança da informação.

REFERÊNCIAS

ABOOD, Omar G.; GUIRGUIS, Shawkat K. A Survey on Cryptography Algorithms. **International Journal of Scientific and Research Publications (IJSRP)**, [*s. l.*], v. 8, n. 7, p. 410–415, 2018.

Disponível em: https://www.ijsrp.org/research-paper-0718.php?rp=P797600. Acesso em: 25 mar. 2025.

ACEA. **FMS-Standard description**. [*S. l.: s. n.*], 2017. Disponível em: https://www.fms-standard.com/Truck/down_load/fms%20document_v_04_vers.13.10.2017.pdf. Acesso em: 25 mar. 2025.

ALENEZI, Mohammed N; ALABDULRAZZAQ, Haneen; MOHAMMAD, Nada Q. Symmetric encryption algorithms: review and evaluation study. **Article in International Journal of Communication Networks and Information Security**, [*s. l.*], v. 12, n. 2, 2020. Disponível em: https://www.ijcnis.org/index.php/ijcnis/article/view/4698. Acesso em: 25 mar. 2025.

BOZDAL, Mehmet; SAMIE, Mohammad; JENNIONS, Ian. A Survey on CAN Bus Protocol: Attacks, Challenges, and Potential Solutions. **International Conference on Computing, Electronics & Communications Engineering (ICCECE)**, [*s. l.*], p. 201–205, 2018. Disponível em: https://ieeexplore.ieee.org/document/8658720. Acesso em: 25 mar. 2025.

BUSCEMI, Alessio; Turcanu, Ion; Castignani, German; Panchenko, Andriy; Engel, Thomas; Shin, Kang G. A survey on controller area network reverse engineering. **IEEE Communications Surveys & Tutorials**, [*s. l.*], v. 25, n. 3, p. 1445–1481, 2023. Disponível em: https://ieeexplore.ieee.org/document/10092880. Acesso em: 25 mar. 2025.

BUTTIGIEG, Robert; FARRUGIA, Mario; MELI, Clyde. Security issues in controller area networks in automobiles. In: INTERNATIONAL CONFERENCE ON SCIENCES AND TECHNIQUES OF AUTOMATIC CONTROL AND COMPUTER ENGINEERING (STA), 18., [*s. l.*], v. 2018-January, p. 93–98, 2017. Disponível em: https://ieeexplore.ieee.org/document/8314877. Acesso em: 25 mar. 2025.

CCITT. **Recommendation X.800**. Geneva: The International Telegraph and Telephone Consulative Commitee, 1991. Disponível em: https://www.uio.no/studier/emner/matnat/ifi/IN2120/h20/docs/x800.pdf. Acesso em: 25 mar. 2025.

CRYPTOGRAPHIC ALGORITHM VALIDATION PROGRAM. **STM32 Cryptographic library validation**. [*S. l.*], 2024. Disponível em: https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13545. Acesso em: 12 jan. 2025.

CSS ELECTRONICS. CAN Bus Explained - A Simple Intro. Aabyhoej, Denmark, 2025. Disponível em: https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial. Acesso em: 24 mar. 2025.

HARTWICH, Florian. CAN with flexible data-rate. **CAN in Automation**, Reutlingen, 2012. Disponível em: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=56c76cf8d21ce99814e11f3d0c6a3a8f339d4c73. Acesso em: 25 mar. 2025.

HUANG, Xiantong; Li, Lang; Zhang, Hong; Yang, Jinling; Kuang, Juanli. IoVCipher: A low-latency lightweight block cipher for internet of vehicles. **Ad Hoc Networks**, [*s. l.*], v. 160, p. 103524, 2024. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S1570870524001355. Acesso em: 25 mar. 2025.

HUYBRECHTS, Thomas; Vanommeslaeghe, Yon; Blontrock, Dries; Van Barel, Gregory; Hellinckx, Peter. Automatic Reverse Engineering of CAN Bus Data Using Machine Learning Techniques. *In*: LECTURE NOTES ON DATA ENGINEERING AND COMMUNICATIONS TECHNOLOGIES. Antwerp: Springer Science and Business Media Deutschland GmbH, 2018. v. 13, p. 751–761.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 27000**. Information technology — security techniques — information security management systems — Overview and vocabulary. Geneva: 2018.

JOSHI, Prachi, Zeng, Haibo; Bordoloi, Unmesh D.; Samii, Soheil; Ravi, S. S.; Shukla, Sandeep K. The multi-domain frame packing problem for CAN-FD. **Leibniz International Proceedings in Informatics, LIPIcs**, [*s. l.*], v. 76, p. 121–1222, 2017. Disponível em: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ECRTS.2017.12. Acesso em: 25 mar. 2025.

KOVACS, Eduard. **Thieves Use CAN Injection Hack to Steal Cars**. [*S. l.*], 2023. Disponível em: https://www.securityweek.com/thieves-use-can-injection-hack-to-steal-cars/. Acesso em: 25 mar. 2025.

LAWRENZ, Wolfhard. **CAN System Engineering**. 2. ed. Wolfenbüttel: Springer, 2013.

MUTTER, Arthur. Robustness of a CAN FD Bus System-About Oscillator Tolerance and Edge Deviations. **CAN in Automation**, [*s. l.*], 2013. Disponível em: https://www.bosch-semiconductors.com/media/ip_modules/pdf_2/papers/icc14_2013_paper_mutter_1.pdf. Acesso em: 25 mar. 2025.

NIELES, Michael; DEMPSEY, Kelley; PILLITTERI, Victoria Yan. **An introduction to information security**. Gaithersburg, MD: National Institute of Standards and Technology, 2017. Disponível em: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-12r1.pdf. Acesso em: 25 mar. 2025.

PAYNE, Bryson R. Car Hacking: Accessing and Exploiting the CAN Bus Protocol. **Journal of Cybersecurity Education, Research and Practice**, [*s. l.*], v. 2019, n. 1, 2019. Disponível em: https://digitalcommons.kennesaw.edu/jcerp/vol2019/iss1/5. Acesso em: 25 mar. 2025.

ROBERT BOSCH GMBH. **Automotive Handbook**. 11. ed. Friedrichshafen and Karlsruhe: Wiley, 2022.

ROBERT BOSCH GMBH. **CAN with Flexible Data-Rate Specification Version 1.0**. 1. ed. Gerlingen: [*s. n.*], 2012. Disponível em: https://tekeye.uk/downloads/can_fd_spec.pdf. Acesso em: 25 mar. 2025.

SINHA, Preeti; RAI, Amit kumar; BHUSHAN, Bharat. Information Security threats and attacks with conceivable counteraction. **2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)**, Kannur, p. 1208–1213, 2019. Disponível em: https://ieeexplore.ieee.org/document/8993384/. Acesso em: 25 mar. 2025.

STALLINGS, William. **Criptografia e segurança de redes**. 6. ed. São Paulo: Pearson Education do Brasil, 2014.

TORRE, David García. **CAN Bus**. 2021. Tese Final. Faculty of Electrical Engineering and Communication. Brno University of Technology, Brno, 2021. Disponível em: https://repositorio.unican.es/xmlui/bitstream/handle/10902/22076/435253.pdf . Acesso em: 25 mar. 2025.

ZHANG, Fan; Liang, Zi-yuan; Yang, Bo-lin; Zhao, Xin-jie; Guo, Shi-ze; Ren, Kui. Survey of design and security evaluation of authenticated encryption algorithms in the CAESAR competition. **Frontiers of Information Technology & Electronic Engineering**, [*s. l.*], v. 19, n. 12, p. 1475–1499, 2018. Disponível em: http://link.springer.com/10.1631/FITEE.1800576. Acesso em: 25 mar. 2025.

Sobre os Autores:

———————————

[i] **Luan Phelippe Almeida Freire Santos**

Graduado em Engenharia da Computação pela Faculdade de Informática e Administração Paulista (2019), pós-graduado em Sistemas Embarcados pelo Centro Universitário SENAI São Paulo (2025). Tem experiência na área de telemetria e rastreamento veicular como analista de engenharia na Newtec Telemetria, e é desenvolvedor de software embarcado para soluções de monitoramento agrícola na Tarvos S.A. https://orcid.org/0009-0006-9197-7205

[ii] **Fernando Simplicio de Sousa**

Professor Centro Universitário SENAI São Paulo no curso de Pós-Graduação em Sistemas Embarcados. Doutor em Engenharia Elétrica pela Universidade Federal do ABC (UFABC). Graduado em Gestão de Pequenas e Médias Empresas pela Universidade Paulista (UNIP) e em Projetos Mecânicos pela Faculdade de Tecnologia de São Paulo (UNESP/FATEC-SP). https://orcid.org/0009-0009-5760-4845

[iii] **Luis Carlos Canno**

Possui graduação em Tecnologia de Automação Industrial (2009) com Especialização em Gestão Empresarial pela Universidade Nove de Julho (2012), e Especialização em Docência na Educação Profissional e Tecnológica pelo Centro de Tecnologia da Indústria Química e Têxtil SENAI CETIQT (2015). Atualmente é professor Centro Universitário SENAI São Paulo. https://orcid.org/0000-0001-9331-9309

[iv] **Leandro Poloni Dantas**

Engenheiro (2004) e Doutor (2018) em Engenharia Elétrica pelo Centro Universitário FEI. Atuou por 15 anos na indústria eletrônica no desenvolvimento de novos produtos. Desde 2009, vem lecionando em cursos de pós-graduação, graduação e de nível técnico em diferentes instituições paulistanas. Atualmente é professor no Centro Universitário SENAI São Paulo e no Insper. https://orcid.org/0000-0003-3674-336X