Encryption of NMEA-2000 for Maritime Cybersecurity

Justin Liaw
Technology and Policy Program/Institute for Data, Systems, and Society
MIT
Cambridge, MA
liawj13@mit.edu

***Abstract*—The NMEA-2000 (N2K) protocol, the primary internal communication standard for maritime sensor and control networks, operates natively without authentication or encryption, leaving vessels susceptible to spoofing, command injection, and replay attacks. Automotive literature broadly dismisses authenticated encryption on Controller Area Network (CAN) bus as operationally infeasible due to latency and bandwidth overhead; however, this consensus is grounded in sub-millisecond automotive reaction deadlines that do not apply to maritime control loops, where heading adjustments operate on timescales of hundreds of milliseconds. This paper evaluates the feasibility of authenticated encryption on N2K networks through a discrete-event simulation of a ten-node representative vessel network, benchmarking three lightweight algorithms, ASCON-128, ChaCha20-Poly1305, and AES-128-CTR+CMAC, across three embedded processor classes representative of realistic marine hardware. Results demonstrate that cryptographic latency is not a barrier to deployment: all algorithms clear the maritime operational timing budget by at least one order of magnitude across all processor classes. The binding constraint is bus bandwidth, where authenticated encryption imposes a threefold increase in frame overhead driven by the fixed 16-byte authentication tag. ASCON-128 is recommended as the preferred algorithm, producing the lowest latency on every node class. A node priority matrix is proposed to guide selective deployment. To address the absence of publicly available N2K traffic datasets, this study contributes a NMEA-0183 to N2K traffic converter and a replicable simulation framework for future maritime cybersecurity research.**

***Index Terms*—Maritime, Cyber Security, Encryption, Lightweight**

I. Introduction

As the maritime industry accelerates its adoption of autonomous systems and advanced digital navigation, the cyber-physical attack surface of modern vessels has expanded. At the core of this vulnerability is the NMEA 2000 (N2K) standard, the primary internal communication protocol used for maritime sensors, displays, and control systems. Based on the Controller Area Network (CAN) bus architecture, N2K operates natively without authentication or encryption [1]. The protocol inherently trusts all traffic on the wire, leaving vessels susceptible to spoofing, replay attacks, and command injection if an adversary gains physical or logical access to the network backbone.

Historically, efforts to secure CAN-based networks have been dominated by the automotive industry. The prevailing consensus within automotive literature dictates that the computational latency and bus load overhead introduced by authenticated encryption are too expensive for real-time control systems [2]. Consequently, encryption on the CAN bus is largely dismissed as operationally infeasible, doubly so in the maritime environment which sees considerably less cybersecurity focus than the automotive industry. However, this assumption does not account for the unique operational realities of the maritime domain. Unlike the microsecond latency thresholds for automotive controls systems, maritime control loops, such as heading adjustments, are often measured in hundreds of milliseconds with the entire turn taking minutes rather than a car's seconds.

This paper challenges the automotive consensus by arguing that the distinct operational deadlines of maritime environments make authenticated encryption viable for NMEA 2000 networks. To evaluate this hypothesis, this research addresses two primary questions. First, can modern cryptographic algorithms, specifically AES-128-CTR paired with HMAC, ChaCha20-Poly1305, and ASCON-128, secure standard N2K payloads without exceeding the latency and bus utilization constraints of a 250 kbps network? Second, given these overhead calculations, which vessel network nodes should be prioritized for cryptographic protection to maximize operational resilience?

To answer these questions, this paper proceeds as follows: Section II reviews the existing literature on CAN bus security and identifies the gap in maritime-specific research. Section III defines the threat model and attack taxonomy. Section IV details the methodology, to include the specific design choices for implementing the cryptographic algorithms, the design choices, and a detailed outline of the simulated environment used for testing. Sections V and VI present and analyze the simulated performance metrics, and Section VII concludes with architectural recommendations, limitations and areas for future work.

II. Literature Review

The security vulnerabilities of the Controller Area Network (CAN) bus have been extensively documented, establishing a foundational consensus that retrofitting security is computationally and operationally expensive. CAN is a broadcast, message-based protocol that inherently lacks sender identification and encryption, making it highly susceptible to spoofing,

message injection, and denial-of-service (DoS) attacks [3], [4]. Many proposals have suggested wrapping CAN frames in cryptographic protections, but these are consistently limited by the operational constraints unique to their sector. Automotive literature widely agrees that while cryptography provides necessary security, it incurs an undesirable or even unacceptable performance degradation. Adding an authentication tag or encrypting payloads introduces significant bandwidth overhead and decreases communication speed, often forcing single 8-byte CAN frames to split into multi-frame transmissions [5]. For automotive systems operating on sub-millisecond reaction deadlines, this latency renders full authenticated encryption operationally unviable [6]. Despite these constraints, the automotive industry has been a significant driver in pushing researchers toward developing lightweight security mechanisms. Recent implementations have introduced lightweight cryptographic schemes, such as optimized block ciphers and truncated hashing algorithms, specifically tailored to execute within the resource-constrained electronic control units (ECUs) of the CAN bus [6]. However, these solutions are almost exclusively anchored in automotive assumptions. Current state-of-the-art approaches to automotive network security primarily focus on Intrusion Detection Systems (IDS) rather than proactive encryption, or they rely on lightweight authentication protocols designed to minimize bandwidth [7]. For example, protocols like CAIBA and LEAP attempt to implement multicast source authentication or reactive bit-flipping to prevent spoofing without incurring the full cost of payload encryption [8], [9]. Other implementations, such as RT-RK CANsecure, achieve node authentication but require replacing or fundamentally altering the underlying hardware architecture of the network [10]. While these collective methods produce measurable security gains, they often sacrifice full cryptographic confidentiality or demand architecture overhauls that are operationally and economically unfeasible in the maritime domain. In the maritime domain, security research concerning the CAN-based N2K protocol remains sparse and largely adopts automotive mitigation strategies. Recent work has explored securing J1939, a communications protocol used for heavy-duty vehicles, and N2K networks, but these efforts predominantly focus on passive monitoring and detection rather than cryptographic prevention. For instance, CANDID proposes a rules-based Intrusion Detection System deployed via edge computing to monitor N2K traffic anomalies [11], and parallels my own thesis work focusing on holistically applying IDS to secure shipboard communications. While cryptographic protections have been researched in maritime environments, they predominantly focus on modern Ethernet-enabled protocols rather than legacy serial or CAN architectures [1].

A. Gap Statement

Consequently, a critical gap exists in the literature. No existing work evaluates the performance of modern, authenticated encryption algorithms (such as ChaCha20-Poly1305 or ASCON-128) on the NMEA 2000 protocol under realistic maritime traffic conditions and analyzes their potential use in maritime operating environments. A goal of this work is to provide the regulatory community with a proof of concept that encryption is feasible on shipboard networks.

III. Threat Model and Attack Taxonomy

The following threat model defines the vulnerabilities inherent in the N2K architecture. This section outlines the attack surface, characterizes the capabilities of a potential adversary, and categorizes specific attack vectors alongside their corresponding cryptographic defenses.

A. Maritime Attack Surface

The N2K network architecture presents a uniquely vulnerable attack surface compared to segmented IT networks. The standard generally utilizes a ”flat” physical backbone topology where no native segmentation exists between low-priority environmental sensors and high-risk control systems, such as the autopilot or electronic throttle. Because the protocol inherently trusts all traffic on the wire, every node on the bus is capable of hearing and broadcasting to every other node. Consequently, once an adversary gains physical or logical access to a single backbone drop point or a compromised wireless gateway, the entire network becomes susceptible to manipulation. This lack of hardware-based isolation or native access control mechanisms means that the security of the vessel is entirely dependent on the integrity of each individual node.

B. Notional Attacker Model

For the purposes of this study, the notional attacker is defined by a specific set of capabilities and limitations. The adversary is assumed to have obtained physical access to the N2K backbone, such as through an exposed spur line or a compromised peripheral device, or logical access via an external-facing bridge system such as Starlink. This attacker possesses the hardware and technical knowledge necessary to perform passive eavesdropping and active frame injection. However, a critical distinction is made between access to the communication medium and access to the cryptographic keys. For the scope of this evaluation, it is assumed the attacker has not compromised the internal non-volatile memory of the participating nodes. This assumption allows for the testing of encryption efficacy while highlighting that key distribution and security is vital to the efficacy of the cryptographic methods.

C. Attack Taxonomy and Defenses

The vulnerabilities inherent to the N2K protocol facilitate several attack classes that directly threaten vessel safety and operational integrity. This study focuses on four vectors of primary concern: spoofing attacks, node impersonation, replay attacks, and denial-of-service attacks. Spoofing represents the most operationally dangerous attack class and the primary concern of the maritime security community. In the N2K context, a spoofing attack occurs when an adversary constructs syntactically valid PGN frames bearing false navigational data, fabricated GPS coordinates, manipulated heading values, or

counterfeit AIS identity reports, and broadcasts them as though originating from a legitimate sensor. Because N2K performs no source verification, receiving nodes such as autopilot accept these frames unconditionally. Node impersonation is a related but distinct vector, in which the adversary assumes the source address of a trusted device causing the vessel to respond to unauthorized instructions without crew awareness. Replay attacks do not require frame construction; an adversary with passive bus access captures valid traffic and retransmits it at a later time, effectively freezing a sensor reading or reproducing a prior navigational state to deceive bridge systems. Denial-of-service attacks operate at the bus level, flooding the CAN arbitration mechanism with high-priority frames until legitimate traffic can no longer compete for bus time, rendering navigation and control systems unable to communicate. These classes of attacks make a strong argument for the utilization of authentication. However, as ships continue to integrate increasingly complex autonomous systems and remote telemetry, encryption becomes essential to prevent adversaries from eavesdropping on sensitive operational data to gain the situational awareness required for targeted attacks. By masking the plaintext payload, encryption ensures that internal vessel states, such as engine telemetry or fuel reserves, remain confidential. For these reasons we chose to require authenticated encryption as our defensive requirement. Denial-of-service attacks, which operate at the CAN arbitration layer rather than the application layer, fall outside the scope of this study, but an IDS would be an appropriate mitigation.

TABLE I
CRYPTOGRAPHIC METHOD EFFECTIVENESS AGAINST N2K ATTACK VECTORS

Attack Type	Enc. Only	Auth. Only	Auth. Enc.
Spoofing		✓	✓
Node impersonation		✓	✓
Replay attack		✓	✓
Eavesdropping	✓		✓
Command injection		✓	✓
Denial of service			

Note: ✓ indicates mitigation is provided.

D. Expert Validation and Operational Context

To ground the experimental design in operational reality, expert input was solicited from two primary stakeholders in the maritime sector. A cybersecurity professor at the United States Naval Academy (USNA) provided critical insights into the prioritization of shipboard node security. A key recommendation was to evaluate cryptographic implementations on a per-node pair basis rather than as a global broadcast requirement. This granular approach serves to mitigate the risk of bus oversaturation, a significant concern in CAN-based N2K networks where total bandwidth is limited to 250 kbps. Furthermore, to ensure high fidelity and maritime realism, this study utilized ship design configurations provided by the Hyundai Shipbuilding Group. These configurations were used to ensure that the simulated environment remained within the bounds of realistic vessel topology and that the modeled computational limitations reflected the actual hardware constraints of modern maritime electronic control units (ECUs).

IV. Design & Methodology

The objective of this study is to evaluate the feasibility of integrating authenticated encryption into the N2K protocol under realistic maritime operating conditions. This section details the simulation environment, the original traffic generation converter, the cryptographic algorithms selected for benchmarking, the nonce construction scheme, and the analytical scaling methodology used to map simulation timing results onto resource-constrained marine hardware.

A. Simulation Environment and Traffic Generation

This research employs OMNeT++ 4.4.2 paired with the INET 2.3.0 framework to achieve high-fidelity CAN bus network modeling. The environment provides discrete-event simulation with nanosecond-resolution timing instrumentation, realistic CAN arbitration, bus contention modeling, and simultaneous multi-node transmission. OMNeT++ 4.4.2 was specifically selected for its compatibility with the INET CAN linklayer stack; subsequent INET versions do not provide equivalent CAN bus support. The NMEA-2000 application layer was partially implemented in this study for the applicable PGNs, sourced from the open-source CANBoat project, which has reverse-engineered the proprietary N2K protocol through network observation and assembly of publicly available data [12].

A persistent barrier to maritime cybersecurity research is the absence of publicly available N2K traffic datasets. To address this gap, an original C++ converter was developed to translate legacy NMEA-0183 plaintext sentences, for which real vessel logs are widely available, into N2K PGN-formatted payloads. After further verification, this converter is intended for open-source release to support future research. Traffic data was sourced from the MARitime SIMulated (MARSIM) dataset, which provides benign NMEA-0183 navigational recordings from simulated maritime scenarios [13]. Utilizing this data ensures the simulation operates on realistic navigational timing rather than synthetic random payloads, improving fidelity of the simulated model.

The OMNeT++ simulation models a ten-node representative vessel network. Nine publisher nodes transmit across 26 distinct PGNs at frequencies consistent with the NMEA-2000 specification; one subscriber node (ECDIS) receives all PGN traffic. Table II details the node topology, processor class assignments, and transmission frequencies. Each experimental run covers a 30-second simulation window, producing 3,154 message events per run and 37,848 events across the full 12-run dataset.

B. Algorithm Selection and Design

Three algorithms were selected to provide a spectrum from purpose-built lightweight cryptography to the well-understood

TABLE II
SIMULATED VESSEL NODE TOPOLOGY, PROCESSOR CLASS ASSIGNMENTS, AND PGN FREQUENCIES

Node	Type	Processor Class	PGNs Published	Freq. (Hz)
GPS	Navigation	M4 @ 168 MHz	129025, 129026, 129029, 129539, 129540, 126992	10 / 1
Autopilot	Navigation	M4 @ 168 MHz	127237, 127245, 127251, 127250	10
Gyro	Navigation	M3 @ 72 MHz	127250, 127251, 127257	10 / 1
VHF/DSC	Navigation	M3 @ 72 MHz	129808	1
AIS	Sensor	M0 @ 48 MHz	129038, 129794	0.1
Engine	Sensor	M0 @ 48 MHz	127488, 127489, 127505, 127508	10 / 1
Depth	Sensor	M0 @ 48 MHz	128267, 128275	1
Wind	Sensor	M0 @ 48 MHz	130306, 130310	1
SpeedLog	Sensor	M0 @ 48 MHz	128259, 128275	1
ECDIS	Subscriber	Host only	All PGNs (rx only)	—

industry standard. All three were required to satisfy the same criteria: symmetric key operation, authenticated encryption delivering both confidentiality and integrity in a single pass, and stream cipher or CTR-mode operation to eliminate padding requirements on 8-byte N2K payloads. Asymmetric cryptography was excluded as computationally infeasible on constrained marine ECUs during real-time operation.

1) ASCON-128: As the winner of the 2023 NIST Lightweight Cryptography standardization process, ASCON-128 is purpose-built for constrained IoT environments [14]. It operates on a 320-bit sponge-duplex state represented as five 64-bit words:

$$S = x_0 \parallel x_1 \parallel x_2 \parallel x_3 \parallel x_4 \tag{1}$$

Each permutation round applies three sequential layers: constant addition to x_2, a non-linear bitwise substitution layer (S-box evaluated in parallel across all 64 bit positions), and a linear diffusion layer applying XOR with rotated copies of each word. Two permutation variants are used: p^{12} for initialization and finalization, and the reduced p^6 for bulk data absorption. Critically, ASCON-128's 64-bit rate maps exactly to the 8-byte N2K DLC, absorbing one complete CAN frame payload per permutation call without alignment overhead. The S-box operates exclusively on bitwise operations with no lookup tables, making it naturally constant-time across all processor classes [15].

2) ChaCha20-Poly1305: Specified in RFC 8439, ChaCha20-Poly1305 is a software-optimized construction that performs well on processors lacking AES hardware acceleration, the common case for marine ECUs [16]. ChaCha20 operates on a 4×4 matrix of 32-bit words and generates a 512-bit keystream block through 20 rounds of the Quarter-Round (QR) function, an Add-Rotate-XOR (ARX) sequence:

$$a \mathrel{+}= b; \quad d \mathrel{\oplus}= a; \quad d \lll 16; \quad c \mathrel{+}= d; \quad b \mathrel{\oplus}= c; \quad b \lll 12 \tag{2}$$

Authentication is provided by the Poly1305 MAC, which evaluates a polynomial over the prime field $\mathbb{F}_{2^{130}-5}$. For each 16-byte message block m_i, the accumulator h is updated as:

$$h = (h + m_i) \cdot r \pmod{2^{130} - 5} \tag{3}$$

The Poly1305 one-time key is derived from the first ChaCha20 keystream block (counter $= 0$), with encryption beginning at counter $= 1$. This construction guarantees that the Poly1305 key is unique per message by derivation, satisfying the one-time key requirement without separate key distribution. As the simulation uses a 128-bit shared key, the 256-bit ChaCha20 key is constructed by concatenation:

$$K_{256} = K_{128} \parallel K_{128} \tag{4}$$

3) AES-128-CTR + AES-CMAC: AES-128 in Counter Mode with AES-CMAC authentication serves as the industry baseline. CTR mode generates a keystream by encrypting successive counter blocks:

$$KS_i = E_K(\text{nonce} \parallel i) \tag{5}$$

Plaintext is encrypted by XOR with the keystream, requiring no padding and supporting arbitrary payload lengths. Authentication is provided by AES-CMAC (NIST SP 800-38B), which generates subkeys K_1 and K_2 from $E_K(0^{128})$ and computes a CBC-MAC variant over the ciphertext under an encrypt-then-MAC construction [17]. AES-128-CTR+CMAC is expected to exhibit a performance advantage on nodes with hardware AES acceleration, which is modeled in this study, and a relative penalty on unaccelerated hardware.

C. Nonce Construction and Replay Resistance

Secure nonce management is essential on N2K networks, which provide no native sequence numbering. Nonce reuse under a shared key is catastrophic: for ChaCha20-Poly1305 and ASCON-128, reusing a nonce exposes the XOR of two plaintexts; for AES-CTR, it directly compromises keystream confidentiality. This implementation assigns each node a 64-bit monotonic message counter, incremented on every transmission. The CAN message ID is incorporated into the nonce construction for all three algorithms, providing per-node domain separation that ensures two nodes sharing the same key cannot produce an identical nonce. The authentication tag is

computed over the ciphertext with the sequence number and CAN ID included as Authenticated Associated Data (AAD):

$$\text{Tag} = \text{Auth}(K,\ \text{CT} \parallel \text{CAN\_ID} \parallel \text{seq}) \tag{6}$$

This construction ensures that a receiving node detects and discards replayed frames or impersonated source addresses before any plaintext is released. On a vessel, counters must persist across power cycles to prevent nonce reuse after node restart, which is identified as future work.

D. Key Management Assumptions

Authenticated encryption requires a shared secret key between communicating nodes. The full key management lifecycle, initial provisioning, rotation, and revocation across a heterogeneous node population over a 20-to-30 year vessel operational lifespan, is outside the scope of this study and is identified as the primary practical barrier to deployment, consistent with findings from the Hyundai Heavy Industries expert interview. For the purposes of this evaluation, a single 128-bit pre-shared key is assumed to be provisioned across all nodes prior to simulation, representing an ideal post-handshake state.

E. Processor Model and Analytical Scaling

Simulation executes on higher-performance development hardware (AMD Ryzen 7 PRO 4750U, 4.1 GHz maximum single-core boost). Direct timing measurements are therefore not representative of marine ECUs, which operate at much more constrained values. To represent hardware limitation, the Cortex-M4, M3, and M0 were used as substitutes for nodes. Measured wall-clock latencies are scaled to target hardware as:

$$T_{\text{target}} = T_{\text{measured}} \times \frac{f_{\text{host}}}{f_{\text{target}}} \times P_{\text{arch}} \tag{7}$$

where P_{arch} is a conservative architectural penalty factor accounting for the absence of hardware multiply instructions on lower-tier processors. Node assignments and scaling parameters are detailed in Table III. P_{arch} values for AES-128-CTR+CMAC are approximated from published embedded benchmarks [18]; ASCON-128 and ChaCha20-Poly1305 carry no penalty as their bitwise-only operations are unaffected by hardware multiply availability [19], [20].

F. Limitations

Several simplifying assumptions were made to isolate cryptographic latency as the experimental variable. First, a single pre-shared key is used across all nodes, representing a post-handshake state. Key distribution, rotation, and revocation are identified as future work. Second, the simulation calculates frame overhead analytically rather than transmitting the authentication tag as physical CAN frames, the +2 frame penalty for the 16-byte tag is applied as a derived metric rather than a simulated bus event. Actual multi-frame transmission under load may introduce additional arbitration latency not captured here. Third, processor scaling derives from clock frequency ratios and published benchmark data rather than hardware-in-the-loop measurement; architectural differences beyond clock speed may affect absolute latency values.

TABLE III
NODE-TO-PROCESSOR ASSIGNMENTS AND ANALYTICAL SCALING PARAMETERS

Node(s)	Proc.	Clock	Rationale	P_a [a]	P_a [b]
GPS, Autopilot	M4	168 MHz	Nav.-grade ECU	1.0	1.0
Gyro, VHF/DSC	M3	72 MHz	Mid-range marine	1.2	1.0
AIS, Engine, Depth, Wind, SpeedLog	M0	48 MHz	Cost-opt. sensor	2.5	1.0

[a]AES-128-CTR+CMAC (penalized for absent hardware multiply). [b]ASCON-128 and ChaCha20-Poly1305 (bitwise-only; no penalty). M0 is worst-case threshold.

V. Analysis & Results

This section presents the simulated performance metrics for each authenticated encryption algorithm across the modeled vessel network and evaluates their implications against the two primary research questions.

A. Latency and Bus Utilization Results

Figure 1 presents the mean encryption and decryption latencies for each algorithm, scaled to the assigned processor class of each node. The 100 ms timing budget is derived from the 10 Hz transmission frequency of the highest-priority navigation nodes, representing the inter-message interval and the most conservative feasibility threshold applied in this study. The results confirm that all three algorithms operate well within the operational timing constraints of the N2K network. The worst-case result across all algorithms and processor classes is AES-128-CTR+CMAC on the Cortex-M0, which produces a total round-trip latency of approximately 127 µs, consuming 0.127% of the available 100 ms budget. ASCON-128 on the same processor class produces a total round-trip latency of approximately 24 µs. Neither result approaches the operational threshold.

B. Bus Utilization

Baseline bus utilization across all nine publisher nodes at their specified NMEA-2000 transmission frequencies is 5.26% of the available 250 kbps bandwidth. Under authenticated encryption, bus utilization increases to 15.77% for all three algorithms, a threefold increase driven entirely by the fixed overhead of the 16-byte authentication tag, which requires two additional CAN frames per message regardless of algorithm choice. The frame overhead is therefore algorithm-independent: the choice of ASCON-128, ChaCha20-Poly1305, or AES-128-CTR+CMAC had the same effect on bus load. The threefold increase in bus utilization warrants careful consideration in the context of real vessel deployment. The ten-node topology modeled in this study is representative of a small to mid-size vessel. Larger commercial vessels, such as container ships or tankers, may operate N2K backbones

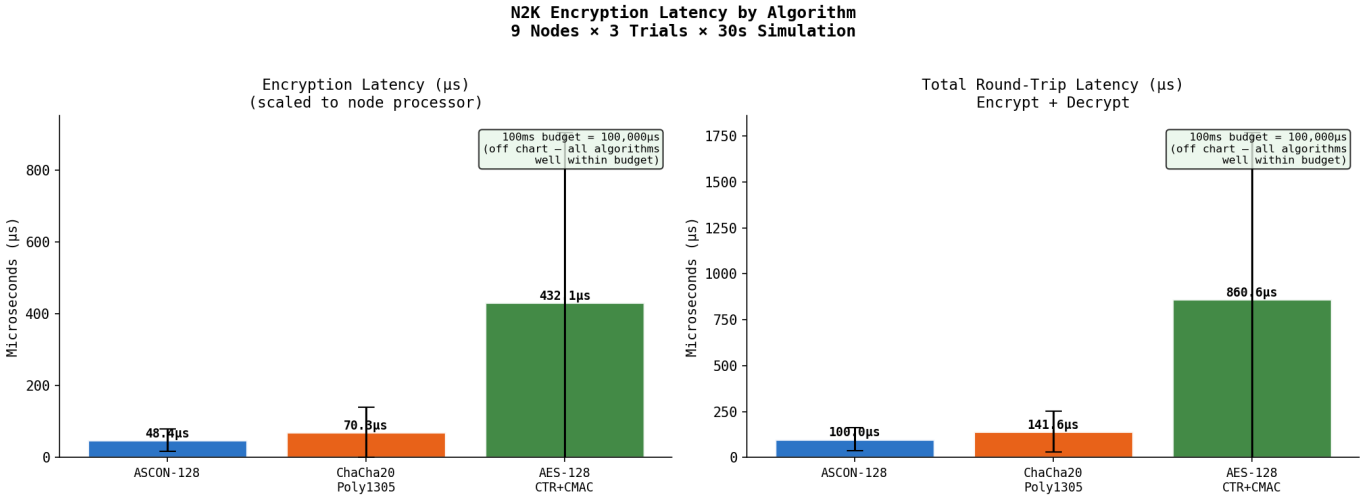

Fig. 1. Per-algorithm encryption and decryption latency scaled to node processor class.

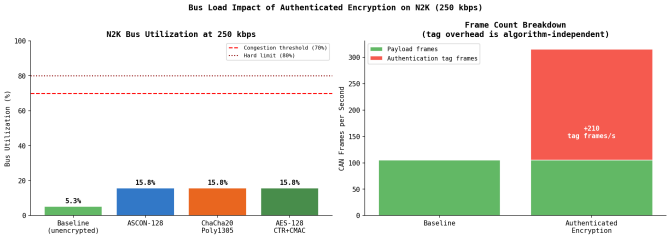

Fig. 2. N2K bus utilization at 250 kbps under baseline and authenticated encryption conditions. All three algorithms produce identical bus load due to the fixed 16-byte authentication tag overhead. Dashed lines indicate practical congestion thresholds at 70% and 80%.

with significantly more nodes and higher aggregate message rates. A network already operating under heavy sensor load could approach congestion thresholds under full authenticated encryption, particularly if high-frequency nodes such as radar or engine monitoring systems are added beyond the scope of this model. However, the bus load result motivates a selective deployment strategy consistent with the node prioritization recommendation in Section IV-D applying authenticated encryption to the highest-consequence nodes first .This approach, independently recommended by the USNA expert interview, limits the bus load increase to the subset of traffic where the security benefit is greatest, preserving headroom for larger or more densely instrumented vessel networks.

C. Node Prioritization Analysis

Having established that authenticated encryption is feasible from a latency perspective but perhaps not an overhead one, the second research question concerns which nodes should be prioritized for cryptographic protection. The answer requires combining the latency results with the threat model established in Section III.

The autopilot and GPS nodes constitute the critical priority tier. The autopilot publishes heading track control, rudder angle, and rate of turn at 10 Hz on a Cortex-M4 processor, producing a worst-case total latency of approximately 12.7 µs under AES-128-CTR+CMAC and 2.4 µs under ASCON-128. Authenticated encryption on these nodes directly defeats command injection and position spoofing attacks identified in Section III as the most operationally dangerous vectors. The GPS node shares the same processor class and frequency profile, and spoofed position data represents the maritime attack class recieving the most attention.

The AIS transceiver and gyrocompass constitute the high priority tier. AIS transmits at 0.1 Hz on a Cortex-M0 processor, producing the most favorable timing profile of any node in the study despite its constrained hardware. The gyrocompass transmits heading and rate of turn at 10 Hz on a Cortex-M3, producing worst-case latencies of approximately 29 µs under AES-128-CTR+CMAC. Both nodes are high-value spoofing targets whose compromise produces systematic navigational deception rather than isolated sensor errors.

The remaining sensor nodes, engine monitor, depth sounder, wind sensor, and speed log, constitute the medium priority tier. All operate on Cortex-M0 processors at 1 Hz or below, with the exception of the engine rapid update PGN at 10 Hz. Their compromise produces operationally significant but less immediately safety-critical consequences than navigation node compromise. Authentication on these nodes is recommended as a second-phase deployment following the critical and high priority tiers.

D. Algorithm Recommendation

Of the three algorithms tested in this study, ASCON-128 is best recommendation. It produces the lowest latency across all three processor classes, with performance advantages most pronounced on the Cortex-M0 nodes that constitute the majority of the vessel network. ChaCha20-Poly1305 is an acceptable alternative, producing low latency results. AES-128-CTR+CMAC is recommended only where hardware AES

TABLE IV
NODE PRIORITY MATRIX FOR AUTHENTICATED ENCRYPTION DEPLOYMENT

Node	Primary Threat	Processor	Priority
Autopilot	Command injection	M4	Critical
GPS	Position spoofing	M4	Critical
AIS	Identity spoofing	M0	High
Gyro	Heading spoofing	M3	High
Engine	False engine data	M0	Medium
SpeedLog	False speed data	M0	Medium
Depth	False depth reading	M0	Medium
Wind	False weather data	M0	Low
VHF/DSC	False distress call	M3	Low

Note: Priority tier reflects consequence of successful attack on each node, informed by USNA expert interview findings. All nodes remain within operational timing budgets across all three algorithms; ASCON-128 is the recommended deployment choice on all tiers.

TABLE V
PER-ALGORITHM LATENCY SCALED TO NODE PROCESSOR CLASS (μS)

Algorithm	Encrypt (μs)		Decrypt (μs)		Round-Trip (μs)	
	Mean	*Std*	*Mean*	*Std*	*Mean*	*Std*
ASCON-128	48.42	30.96	51.59	38.73	100.01	63.69
ChaCha20-Poly1305	70.33	69.53	71.24	60.38	141.57	111.01
AES-128-CTR+CMAC	432.13	472.59	428.48	448.47	860.62	908.43

Maritime feasibility threshold: 100 ms (10 Hz inter-message interval)

Algorithm	Worst-case round-trip	% of 100 ms budget
ASCON-128	2.09 ms	2.09%
ChaCha20-Poly1305	5.28 ms	5.28%
AES-128-CTR+CMAC	12.04 ms	12.04%

Note: Values scaled from host measurements (AMD Ryzen 7 PRO 4750U, 4.1 GHz) to assigned node processor class using Eq. (7). Worst-case values reflect AES-128-CTR+CMAC and ChaCha20-Poly1305 on Cortex-M0 nodes; ASCON-128 worst case on Cortex-M0. All algorithms remain within the 100 ms inter-message budget of 10 Hz navigation nodes.

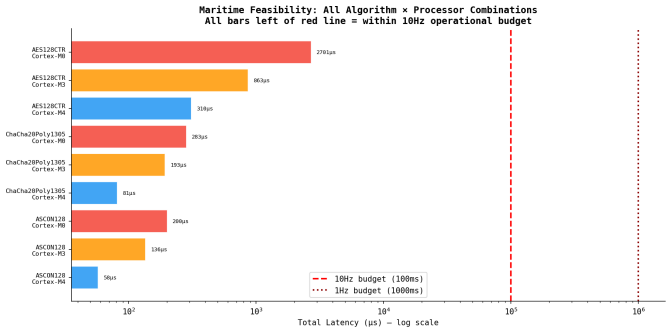

Fig. 3. Maritime feasibility summary: total round-trip latency for all algorithm and processor class combinations (log scale). The vertical reference line at 100 ms represents the conservative inter-message budget of 10 Hz navigation nodes. All combinations fall within the operational threshold by at least one order of magnitude.

acceleration is confirmed present, as its software performance on Cortex-M0 class processors is approximately five times slower than ASCON-128 on the same hardware.

The central finding of this study is that authenticated encryption on NMEA-2000 is not meaningfully constrained by cryptographic latency. Every algorithm, on every processor class modeled, clears the maritime operational timing budget by at least one order of magnitude. The binding constraint is bus bandwidth, not compute.

This bandwidth constraint is, however, circumventable. We could accept a security tradeoff by truncating the tags from 128 bits to 64 bits, reducing our needed additional frames by 50%. As was discussed, implementing prioritized encryption on certain nodes offers a middle ground between security and overhead. Finally designing with the constraint and limiting devices is an option. The third strategy is, however, in tension with the trajectory of the maritime industry. The trend toward Software Defined Vessels and increasing automation is expanding, not contracting the number of networked nodes aboard modern vessels. Designing around the bandwidth constraint by limiting network devices runs counter to this direction. This makes tag truncation and selective encryption the more practically viable mitigations, and identifies right-sized authentication tag length for maritime N2K as a concrete open question for future work.

VI. CONCLUSION & REFLECTION

This study evaluated the feasibility of authenticated encryption on NMEA-2000 CAN bus networks under realistic maritime operating conditions. The results demonstrate that cryptographic latency is not a barrier to deployment: all three algorithms — ASCON-128, ChaCha20-Poly1305, and AES-128-CTR+CMAC, clear the operational timing budget by at least one order of magnitude across all modeled processor classes. The binding constraint is bus bandwidth, where authenticated encryption imposes a threefold increase in frame overhead.

To address the persistent absence of publicly available N2K datasets, this study produced two open contributions: an OMNeT++ discrete-event simulation of a representative ten-node vessel network, and an NMEA-0183 to N2K converter that generates realistic PGN-formatted traffic. Both artifacts are intended for open-source release to support replication and extension of this work by the maritime cybersecurity research community.

Future work should examine minimum acceptable authentication tag lengths for maritime N2K and validate the analytical scaling results through hardware-in-the-loop experimentation on physical marine ECUs.

A. Who did what

My group was disbanded the week before presentation, so this is a single project. I was able to utilize my associated work and knowledge from my thesis and RA but all parts of this project were made and completed for this project. The interviews I was able to conduct were, in reality, just a few extra questions I was able to ask during work associated with my RA.

I had a great time with the project and am hoping to present and/or publish this work after some further scrutiny by my advisors. Any feedback is appreciated!

References

References

[1] C. Hemminghaus, J. Bauer, and K. Wolsing, "SIGMAR: Ensuring integrity and authenticity of maritime systems using digital signatures," in *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, 2021, pp. 1–6.

[2] E. Aliwa, O. Rana, C. Perera, and P. Burnap, "Cybersecurity defenses for in-vehicle networks: The state of the art," *IEEE Access*, vol. 9, pp. 1469–1489, 2020.

[3] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010, pp. 447–462.

[4] E. Aliwa *et al.*, "Robust detection framework for adversarial threats in autonomous vehicle platooning," *Frontiers in Big Data*, 2025. [Online]. Available: https://www.frontiersin.org/journals/big-data/articles/10.3389/fdata.2025.1617978/full

[5] Y. Zhang *et al.*, "Analyzing CAN's timing under periodically authenticated encryption," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE)*. IEEE, 2022.

[6] S. Chen *et al.*, "Research on lightweight dynamic security protocol for intelligent in-vehicle CAN bus," *Sensors (Basel)*, vol. 24, no. 17, p. 5556, 2024.

[7] J. Liao *et al.*, "A survey and comparative analysis of security properties of CAN authentication protocols," *IEEE Access*, 2024.

[8] F. Wagner *et al.*, "CAIBA: Multicast source authentication for CAN through reactive bit flipping," in *COMSYS RWTH Aachen*, 2025.

[9] S. Radhakrishnan *et al.*, "LEAP: Lightweight encryption and authentication protocol for in-vehicle networks," in *IEEE International Conference on Communications (ICC)*. IEEE, 2014.

[10] D. Oladimeji, A. Rasheed, M. Baza, and N. K. Shashidhar, "CANsecure: A secure lightweight framework for CAN protocol in modern vehicles," in *2025 International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2025, pp. 380–387.

[11] M. Rogers, P. Weigand, J. Happa, and K. Rasmussen, "Detecting can attacks on j1939 and nmea 2000 networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 2406–2420, 2023.

[12] K. Verruijt, "CANboat: NMEA 2000 and NMEA 0183 data parser and converter," https://github.com/canboat/canboat, 2024, accessed: May 2026.

[13] J. Spravil, C. Hemminghaus, M. von Rechenberg, E. Padilla, and J. Bauer, "MARSIM: MARitime SIMulated dataset for GPS spoofing detection," https://zenodo.org/records/8202936, 2023, benign NMEA-0183 navigational logs used for simulation traffic generation.

[14] National Institute of Standards and Technology, "Lightweight cryptography standardization process: ASCON," NIST, Tech. Rep., 2023.

[15] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "ASCON v1.2: Submission to NIST lightweight cryptography standardization," https://eprint.iacr.org/2021/1574, IACR Cryptology ePrint Archive, Tech. Rep. 2021/1574, 2021.

[16] Y. Nir and A. Langley, "ChaCha20 and Poly1305 for IETF protocols," RFC 8439, Internet Engineering Task Force, June 2018. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8439

[17] Dworkin, Morris, "Recommendation for block cipher modes of operation: The CMAC mode for authentication," https://doi.org/10.6028/NIST.SP.800-38B, National Institute of Standards and Technology, Tech. Rep. NIST SP 800-38B, 2016.

[18] N. Mouha, B. Mennink, A. Van Herrewege, D. Watanabe, B. Preneel, and I. Verbauwhede, "Chaskey: A MAC algorithm for microcontrollers," in *Proc. Selected Areas in Cryptography (SAC) 2014*, ser. Lecture Notes in Computer Science, vol. 8781. Springer, 2014, pp. 244–269.

[19] M. J. Kannwischer, P. Schwabe, D. Whiting, and A. Wiggers, "Optimized software implementations of Ascon, Grain-128AEAD, and Tiny-JAMBU on ARM Cortex-M0," https://eprint.iacr.org/2022/1079, IACR Cryptology ePrint Archive, Tech. Rep. 2022/1079, 2022.

[20] L. Cardoso dos Santos, J. Großschädl, and A. Biryukov, "FELICS-AEAD: Benchmarking of lightweight authenticated encryption algorithms," in *Proc. NIST Lightweight Cryptography Workshop 2019*, Gaithersburg, MD, USA, November 2019.