

Comprehensive Survey on CAN, CAN FD, and Automotive Security: Evolution, Challenges, and Future Directions

Sandhya Lagade¹

¹*Department of Electronics and Telecommunication Engineering,
TPCT's College of Engineering, Dharashiv, Maharashtra, India*

Abstract - The Controller Area Network (CAN) protocol has been the backbone of in-vehicle communication systems for over three decades. As modern vehicles evolve into highly connected cyber-physical systems with increased electronic content, advanced driver assistance capabilities, and connectivity features, automotive communication networks face unprecedented challenges related to bandwidth limitations, real-time performance, and cybersecurity threats. This comprehensive survey presents a systematic review of CAN protocol fundamentals, the evolution toward CAN with Flexible Data-Rate (CAN FD), automotive network security vulnerabilities, attack vectors, defense mechanisms, and emerging trends in secure automotive communication. The paper analyzes classical CAN architecture, frame structure, arbitration mechanisms, and error-handling capabilities. It examines CAN FD enhancements including increased payload capacity, dual bit-rate transmission, and backward compatibility considerations. A detailed investigation of automotive cybersecurity threats—including message injection, replay attacks, denial-of-service attacks, ECU spoofing, and remote exploitation—is presented along with cryptographic countermeasures, intrusion detection systems, secure gateways, and authentication protocols. The survey also explores standardization efforts including ISO 11898, SAE J1939, AUTOSAR security specifications, and emerging technologies such as Automotive Ethernet, Time-Sensitive Networking (TSN), and blockchain-based security frameworks. Critical analysis of over 30 research contributions reveals that while CAN FD addresses bandwidth limitations, comprehensive security solutions require multi-layered defense strategies combining authentication, encryption, anomaly detection, and secure software update mechanisms. Future research directions include lightweight cryptography for resource-constrained ECUs, machine-learning-based intrusion detection, quantum-resistant security protocols, and integration with Vehicle-to-Everything (V2X) communication infrastructure. This survey provides researchers, automotive engineers, and security practitioners with a consolidated understanding of state-of-the-art automotive communication technologies and security methodologies essential for designing next-generation secure connected vehicles.

Keywords: Automotive cybersecurity, CAN protocol, CAN FD, in-vehicle network security, intrusion detection, vehicle cyber-attacks.

1. INTRODUCTION

A. Background and Motivation

The automotive industry has undergone a profound transformation from purely mechanical systems to sophisticated cyber-physical platforms integrating hundreds of microprocessors, sensors, actuators, and communication networks. Modern vehicles contain 70 to 150 Electronic Control Units (ECUs) managing critical functions including powertrain control, chassis dynamics, advanced driver assistance systems (ADAS), infotainment, body electronics, and telematics [1]. These distributed ECUs must exchange sensor data, control commands, diagnostic information, and status updates reliably and efficiently through standardized in-vehicle communication protocols.

The Controller Area Network (CAN) protocol, developed by Robert Bosch GmbH in the mid-1980s and standardized as ISO 11898, has become the dominant communication technology in automotive systems [2]. CAN provides robust multi-master broadcast communication with priority-based arbitration, error detection, fault confinement, and automatic retransmission capabilities [3]. Its widespread adoption stems from deterministic behavior, low implementation cost, proven reliability, and suitability for real-time distributed control applications.

However, classical CAN faces significant limitations in contemporary automotive applications:

1. Limited data payload: Maximum 8 bytes per frame restricts efficient transmission of high-resolution sensor data, camera images, and software updates.
2. Bandwidth constraints: Maximum bit rate of 1 Mbps becomes insufficient when numerous ECUs transmit frequent updates simultaneously.
3. Lack of native security: CAN was designed assuming a closed trusted network without authentication, encryption, or access control mechanisms.
4. Vulnerability to cyber-attacks: Broadcasting nature and absence of source authentication enable message injection, spoofing, and denial-of-service attacks [4].
5. Scalability challenges: Increasing number of ECUs and communication requirements strain network capacity and real-time guarantees.

To address bandwidth and payload limitations, CAN with Flexible Data-Rate (CAN FD) was introduced, supporting payloads up to 64 bytes and bit rates up to 8 Mbps during the data phase while maintaining backward compatibility with classical CAN [5].

Automotive cybersecurity has emerged as a critical concern following demonstrations of remote vehicle exploits, unauthorized control of critical functions, and potential safety implications [6][7]. The shift toward connected vehicles, over-the-air software updates, vehicle-to-everything (V2X)

communication, and autonomous driving intensifies security requirements.

B. Scope and Contributions

This comprehensive survey provides an in-depth analysis of:

- CAN protocol architecture: Detailed examination of CAN frame structure, arbitration mechanisms, error handling, fault confinement, and higher-layer protocols.
- CAN FD evolution: Technical improvements, performance benefits, implementation considerations, and compatibility challenges.
- Automotive security landscape: Threat models, attack vectors, vulnerability analysis, real-world exploits, and impact assessment.
- Defense mechanisms: Cryptographic authentication, intrusion detection systems, secure gateways, hardware security modules, and secure boot mechanisms.
- Standardization and regulations: ISO 11898, SAE standards, AUTOSAR security modules, ISO/SAE 21434, and regulatory compliance requirements.
- Emerging technologies: Automotive Ethernet, Time-Sensitive Networking, software-defined vehicles, and next-generation secure architectures.
- Research challenges: Lightweight cryptography, real-time intrusion detection, key management, secure firmware updates, and privacy preservation.

C. Survey Methodology

This survey systematically reviews over 30 peer-reviewed publications including journal articles, conference papers, technical standards, industry reports, and security advisories published between 1990 and 2024. Literature was collected from IEEE Xplore, ACM Digital Library, ScienceDirect, Springer, SAE International, ISO standards, NHTSA reports, and automotive cybersecurity repositories.

Papers were categorized into:

- CAN protocol fundamentals and applications
- CAN FD technical specifications and implementations
- Automotive security vulnerabilities and attack demonstrations
- Cryptographic and non-cryptographic defense mechanisms
- Intrusion detection and anomaly detection approaches
- Secure automotive architectures and protocols
- Standards, regulations, and best practices

D. Paper Organization

The remainder of this survey is organized as follows:

Section 2 presents CAN protocol fundamentals including architecture, frame formats, arbitration, error handling, and physical layer characteristics.

Section 3 examines CAN FD protocol enhancements, technical specifications, performance improvements, and migration considerations.

Section 4 analyses automotive cybersecurity threats, attack taxonomies, vulnerability assessment, and documented real-world exploits.

Section 5 reviews defence mechanisms including authentication protocols, encryption schemes, intrusion detection systems, and secure gateway architectures.

Section 6 discusses automotive security standards, regulatory frameworks, AUTOSAR security modules, and compliance requirements.

Section 7 identifies open research challenges, future directions, and recommendations for secure automotive system design.

Section 8 concludes the survey with key findings and insights.

2. CAN PROTOCOL FUNDAMENTALS

A. Historical Development and Standardization

The Controller Area Network protocol was developed by Robert Bosch GmbH in 1983 to address the growing complexity of automotive wiring harnesses. Before CAN adoption, point-to-point wiring between sensors, switches, actuators, and controllers resulted in heavy cable bundles, increased vehicle weight, installation complexity, and maintenance challenges [8].

CAN introduced a multi-master broadcast serial bus enabling multiple ECUs to communicate over a shared two-wire differential bus without a central host controller. The protocol specification was officially released in 1986 and first implemented in the Mercedes-Benz W140 S-Class in 1991 [9]. The International Organization for Standardization (ISO) formalized CAN as ISO 11898 in 1993, covering the data link layer and physical layer specifications [10]. Subsequent revisions and extensions include:

- **ISO 11898-1:** Data link layer and logical link control
- **ISO 11898-2:** High-speed CAN physical layer (up to 1 Mbps)
- **ISO 11898-3:** Low-speed fault-tolerant CAN physical layer
- **ISO 11898-5:** High-speed CAN with low-power mode
- **ISO 11898-6:** Selective wake-up functionality

CAN became the de facto standard for automotive in-vehicle networks and is also widely used in industrial automation, medical equipment, maritime systems, and aerospace applications.

B. CAN Architecture and Topology

CAN implements a bus topology where all nodes are connected to a common two-wire differential bus (CAN_H and CAN_L). The differential signaling improves noise immunity and electromagnetic compatibility [11].

Key architectural characteristics:

1. Multi-master broadcast: Any node can initiate transmission when the bus is idle; all nodes receive transmitted frames.
2. Message-oriented communication: Frames are identified by message identifiers rather than node addresses.

3. Priority-based arbitration: Non-destructive bitwise arbitration resolves simultaneous transmission attempts.
4. Deterministic behavior: Highest-priority messages guaranteed to transmit with bounded latency.
5. Error detection and signaling: Multiple error detection mechanisms with automatic retransmission.

Physical layer: CAN uses differential voltage signaling between CAN_H and CAN_L lines:

- Dominant bit (logic 0): CAN_H $\approx$ 3.5V, CAN_L $\approx$ 1.5V, differential voltage $\approx$ 2V
- Recessive bit (logic 1): CAN_H $\approx$ 2.5V, CAN_L $\approx$ 2.5V, differential voltage $\approx$ 0V

The bus must be terminated with 120 Ω resistors at both physical ends to prevent signal reflections and ensure proper bus biasing.

Network topology constraints:

- Maximum bus length depends on bit rate: at 1 Mbps, maximum $\sim$ 40 meters
- Maximum number of nodes: typically 30-110 depending on transceiver specifications
- Star, linear, and tree topologies supported with appropriate stub length limitations

C. CAN Frame Structure

CAN supports four frame types:

1. Data Frame

The data frame carries application data from a transmitting node to receiving nodes.

Table -1: Standard CAN (11-bit identifier) frame structure

Field	Length	Description
Start of Frame (SOF)	1 bit	Dominant bit indicating frame start
Identifier	11 bits	Message priority and identification
Remote Transmission Request (RTR)	1 bit	Data frame (dominant) or remote frame (recessive)
Identifier Extension (IDE)	1 bit	Standard (dominant) or extended (recessive) format
Reserved bit (r0)	1 bit	Reserved, must be dominant
Data Length Code (DLC)	4 bits	Number of data bytes (0-8)
Data Field	0-64 bits	Actual payload data
CRC Sequence	15 bits	Cyclic redundancy check
CRC Delimiter	1 bit	Recessive bit
ACK Slot	1 bit	Transmitter sends recessive; receivers send dominant if correct
ACK Delimiter	1 bit	Recessive bit
End of Frame (EOF)	7 bits	Seven recessive bits

Intermission	3 bits	Three recessive bits (minimum idle time)
--------------	--------	--

Extended CAN (29-bit identifier) frame structure:

Adds an 18-bit identifier extension after the Substitute Remote Request (SRR) and IDE bits, providing 29-bit addressing for larger networks.

2. Remote Frame

A remote frame requests data from another node. It has the same structure as a data frame except:

- RTR bit is recessive
- No data field (DLC can specify expected data length)

3. Error Frame

Transmitted when a node detects an error. Consists of:

- Error flag: 6 dominant bits (violates bit-stuffing rule)
- Error delimiter: 8 recessive bits

4. Overload Frame

Indicates a node is not ready to receive additional frames. Similar structure to error frame but transmitted during interframe space.

D. Arbitration and Medium Access Control

CAN uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) with non-destructive bitwise arbitration.

Arbitration process:

1. Node waits for bus idle (recessive state)
2. Node starts transmitting identifier bit-by-bit
3. Each transmitting node monitors the bus while transmitting
4. If a recessive bit is transmitted but dominant is monitored, the node stops (loses arbitration)
5. The message with the lowest numerical identifier (highest priority) wins arbitration
6. Losing nodes become receivers and retry transmission when the bus becomes idle

Key advantages:

- No time wasted due to collisions
- Deterministic priority-based scheduling
- Highest-priority messages experience minimal latency
- Automatic retry for lower-priority messages

Priority inversion: Lower-priority frames may experience unbounded delay if higher-priority traffic continuously occupies the bus.

E. Error Detection Mechanisms

CAN implements five error detection methods:

1. Cyclic Redundancy Check (CRC)

15-bit CRC polynomial applied to SOF, arbitration field, control field, and data field. Detects:

- All single-bit errors
- All double-bit errors
- All odd number of bit errors
- Burst errors up to 15 bits
- Most longer burst errors

2. Frame Check

Verifies frame format compliance (bit positions, field lengths, fixed bit values).

3. ACK Check

Transmitter verifies that at least one receiver acknowledged correct reception.

4. Bit Monitoring

Transmitting node monitors the bus and compares transmitted and monitored bit values (except during arbitration and ACK slot).

5. Bit Stuffing

After five consecutive identical bits, a complementary bit is inserted. Receiving nodes remove stuffed bits. Violation indicates error.

Error handling:

- Detected errors trigger error frame transmission
- Automatic retransmission of corrupted frames
- Error counters track transmission and reception errors
- Fault confinement mechanisms prevent defective nodes from disrupting communication

F. Fault Confinement

CAN implements sophisticated fault confinement to isolate defective nodes:

Error states:

1. Error-Active: Node can transmit active error frames (6 dominant bits)
2. Error-Passive: Node can only transmit passive error frames (6 recessive bits)
3. Bus-Off: Node is disconnected from the bus

Error counters:

- Transmit Error Counter (TEC): Incremented on transmission errors
- Receive Error Counter (REC): Incremented on reception errors

State transitions:

- Error-Active ↔ Error-Passive: Threshold at 128 errors
- Error-Passive → Bus-Off: TEC exceeds 255
- Bus-Off → Error-Active: After monitoring 128×11 recessive bits

This mechanism prevents "babbling nodes" (continuously transmitting corrupted frames) from disrupting network operation.

G. Higher-Layer Protocols

Several higher-layer protocols have been developed on top of CAN:

1. CANopen

Developed by CAN in Automation (CiA) for industrial automation and embedded control. Features:

- Device profiles for motors, encoders, I/O modules
- Object dictionary for standardized data access
- Network management services
- Service Data Objects (SDO) for configuration
- Process Data Objects (PDO) for real-time data exchange

2. J1939

SAE standard for commercial vehicles and heavy-duty trucks. Characteristics:

- 29-bit extended identifier
- Parameter Group Number (PGN) for message identification
- Multi-packet transport protocol for messages >8 bytes

- Standardized Suspect Parameter Numbers (SPN)
- Diagnostic trouble codes (DTC)

3. DeviceNet

Industrial automation protocol using CAN physical layer. Provides:

- Master-slave and peer-to-peer communication
- Explicit messaging for configuration
- I/O messaging for real-time control
- Vendor-specific and standard device profiles

4. NMEA 2000

Maritime communication protocol based on CAN. Used for:

- Navigation systems
- Instrumentation
- Sensor networks on boats and ships

5. CANaerospace

Aerospace and aviation protocol with:

- Defined message structures
- Data types for sensor values
- Standardized node identifiers

H. CAN Performance Characteristics

Advantages:

1. High reliability: Error detection probability > 99.9%
2. Real-time performance: Deterministic arbitration ensures bounded latency for high-priority messages
3. Low cost: Mature technology with inexpensive transceivers and controllers
4. Robustness: Differential signaling, error handling, fault confinement
5. Simplicity: Straightforward implementation and debugging
6. Multicast: Efficient broadcast reduces redundant transmissions

Limitations:

1. Limited bandwidth: Maximum 1 Mbps
2. Small payload: Only 8 bytes per frame
3. No security: No authentication, encryption, or access control
4. Priority inversion: Low-priority messages can be delayed indefinitely
5. Bus length limitations: High bit rates require short bus lengths
6. No support for large data: Multi-frame protocols needed for software updates

3. CAN FD PROTOCOL EVOLUTION

A. Motivation for CAN FD Development

As automotive systems became more sophisticated, classical CAN limitations became increasingly problematic:

1. High-resolution sensors: Camera systems, radar, lidar generate large data volumes
2. Flash programming: Software updates and calibration data require megabytes of transfer
3. Diagnostic data: Detailed fault logs and event recording exceed 8-byte capacity
4. Network congestion: Increasing ECU count and communication frequency saturate 1 Mbps bandwidth

5. Development efficiency: Small payloads increase protocol overhead and complicate application development

CAN FD (CAN with Flexible Data-Rate) was developed by Bosch in 2011 and standardized in ISO 11898-1:2015 to address these limitations while maintaining CAN protocol compatibility [12].

B. CAN FD Technical Enhancements

1. Increased Payload Capacity

CAN FD supports payloads from 0 to 64 bytes, compared to classical CAN's 0 to 8 bytes.

Table -2: DLC encoding for CAN FD

DLC	Data Bytes	DLC	Data Bytes
0-8	0-8	9	12
10	16	11	20
12	24	13	32
14	48	15	64

2. Dual Bit-Rate Transmission

CAN FD introduces two bit-rate phases:

- Arbitration phase: Uses classical CAN bit rate (typically 500 kbps) for compatibility
- Data phase: Switches to faster bit rate (up to 8 Mbps) for data field transmission

Table -3: Frame structure differences

Phase	CAN	CAN FD
Arbitration	Up to 1 Mbps	Typically 500 kbps
Data	Same as arbitration	Up to 8 Mbps

3. Improved CRC Protection

CAN FD uses longer CRC sequences for enhanced error detection:

- 17-bit CRC: For payloads ≤ 16 bytes
- 21-bit CRC: For payloads > 16 bytes

Additional CRC improvements include stuff bit counting to detect systematic errors.

4. Frame Format Modifications

CAN FD frame differences from classical CAN:

- FDF bit (Flexible Data Format): Distinguishes CAN FD frames from classical CAN
- BRS bit (Bit Rate Switch): Indicates bit-rate switching for data phase
- ESI bit (Error State Indicator): Indicates error state of transmitting node
- Extended Data Length Code (DLC): Supports encoding for up to 64 bytes
- No stuffing in CRC field: Reduces overhead and improves efficiency

5. Performance Improvements

Bandwidth comparison:

For 64-byte payload transmission:

- Classical CAN (1 Mbps): Requires eight 8-byte frames ≈ 880 bits ≈ 880 μ s

- CAN FD (1/8 Mbps): Single 64-byte frame ≈ 544 bits ≈ 135 μ s (arbitration at 1 Mbps, data at 8 Mbps)

Effective throughput improvement: $6.5\times$ faster for large data transfers.

C. CAN FD Frame Structure

Arbitration phase (compatible with classical CAN):

- Start of Frame (SOF)
- Identifier (11 or 29 bits)
- RTR/RRS bits
- IDE bit
- FDF bit (recessive for CAN FD)
- Reserved bit

Control field:

- BRS bit: Indicates bit-rate switch
- ESI bit: Error state indicator
- DLC (4 bits): Data length code

Data phase (high-speed):

- Data field (0-64 bytes)
- CRC field (17 or 21 bits + CRC delimiter)
- Stuff count (modulo 8 gray code)

Acknowledgment and end (classical speed):

- ACK slot and delimiter
- End of Frame (7 bits)

D. CAN FD Implementation Considerations

1. Hardware Requirements

- CAN FD controllers: Support dual bit-rate and extended payload
- Transceivers: Must handle higher bit rates and switching delays
- Bus topology: Critical for high-speed data phase (shorter stubs, controlled impedance)
- Termination: Precise 120Ω termination essential at high bit rates

2. Software Stack

- Driver adaptation: Updated drivers for CAN FD frame handling
- Higher-layer protocols: J1939, AUTOSAR COM stack modifications
- Application layer: Utilize larger payloads efficiently

3. Backward Compatibility

CAN FD controllers can operate in three modes:

1. Classical CAN only: Transmit/receive only classical CAN frames
2. CAN FD mode: Transmit/receive CAN FD frames
3. Mixed mode: Coexistence handling (both types supported)

Compatibility challenges:

- Classical CAN nodes see CAN FD frames as error frames (FDF bit is recessive in reserved position)
- Network segmentation may be required during migration
- Gateway ECUs can translate between CAN and CAN FD segments

4. Network Design

- Bit-rate selection: Balance between speed and bus length
- Message prioritization: Maintain real-time guarantees

- Bus loading analysis: Ensure sufficient bandwidth margin
- Error handling: Adapted error management strategies

E. CAN FD Performance Analysis

Advantages over classical CAN:

1. Higher throughput: Up to 10× improvement for large payloads
2. Reduced latency: Fewer frames needed for equivalent data volume
3. Lower bus loading: Fewer frame transmissions reduce arbitration overhead
4. Protocol efficiency: Reduced overhead percentage for large data transfers
5. Improved diagnostics: Larger frames support detailed diagnostic messages
6. Faster programming: Flash updates 3-5× faster

Remaining limitations:

1. Still no security: No authentication or encryption mechanisms
2. Physical layer constraints: High bit rates require careful bus design
3. Increased complexity: Dual bit-rate timing requires precise configuration
4. Not suitable for long buses: High-speed data phase limits bus length
5. Compatibility management: Migration complexity in existing systems

F. CAN FD Adoption and Industry Trends

Current adoption status:

- Automotive OEMs: Major manufacturers deploying CAN FD in new vehicle platforms
- Supplier ecosystem: Semiconductor vendors, ECU suppliers support CAN FD
- Tool vendors: Diagnostic tools, development environments updated for CAN FD
- Standards bodies: AUTOSAR, SAE updated specifications for CAN FD integration

Application domains:

1. Powertrain ECUs: Engine control, transmission management
2. Gateway modules: Central communication hubs
3. ADAS systems: Camera, radar, sensor fusion ECUs
4. Body control: Lighting, climate, comfort functions
5. Diagnostic systems: Enhanced diagnostic capabilities
6. Software updates: Over-the-air (OTA) update infrastructure

4. AUTOMOTIVE CYBERSECURITY THREATS**A. Threat Landscape Overview**

Modern vehicles are increasingly exposed to cybersecurity threats due to:

1. Connectivity proliferation: Cellular, Wi-Fi, Bluetooth, V2X, OBD-II dongles
2. Software complexity: Millions of lines of code with potential vulnerabilities

3. Supply chain risks: Third-party components may contain weaknesses
4. Long vehicle lifecycle: 10-15 year operational period with evolving threats
5. Safety-critical systems: Attacks can directly impact passenger safety

Threat actors:

- Hackers/researchers: Demonstrate vulnerabilities for awareness
- Criminals: Vehicle theft, ransomware, data theft
- Nation-states: Espionage, sabotage, infrastructure disruption
- Insiders: Malicious employees, aftermarket modifications

B. Attack Surface Analysis

Entry points for automotive cyber-attacks:

1. Wireless Interfaces

- Cellular (4G/5G): Telematics control units, remote services
- Wi-Fi: Infotainment, diagnostics, software updates
- Bluetooth: Hands-free, streaming, phone pairing
- Keyless entry/RFID: Relay attacks, signal amplification
- Tire pressure monitoring (TPMS): Unencrypted RF signals
- Remote start systems: Aftermarket installations

2. Physical Access Points

- OBD-II port: Direct CAN bus access, diagnostic tools
- USB ports: Malicious firmware, infected media
- SD cards: Infected map updates, malware injection
- Service connectors: Diagnostic interfaces, programming ports

3. Supply Chain

- Third-party modules: Vulnerable ECU firmware
- Software dependencies: Compromised libraries, toolchains

4. V2X Communication

- Vehicle-to-Vehicle (V2V): Message spoofing, false warnings
- Vehicle-to-Infrastructure (V2I): Traffic light manipulation, fake signals
- Vehicle-to-Cloud (V2C): Server compromises, data breaches

C. CAN Bus Attack Vectors**1. Message Injection**

Description: Attacker injects malicious CAN frames onto the bus.

Attack scenarios:

- Send spoofed sensor values (speed, RPM, temperature)
- Trigger unintended actuator commands (unlock doors, disable brakes)
- Flood bus with high-priority frames (denial of service)

Example: Inject frames commanding steering angle changes or brake pressure reduction.

Requirements:

- Physical or remote access to CAN bus
- Knowledge of message identifiers and format

Impact:

- Safety-critical function manipulation
- Vehicle behavior alteration
- Driver confusion or panic

2. Replay Attacks

Description: Capture legitimate CAN frames and retransmit them later.

Attack scenarios:

- Record door-unlock sequence and replay to gain access
- Capture authentication handshake and replay
- Store control commands for later execution

Requirements:

- CAN bus monitoring capability
- Frame storage and retransmission capability

Impact:

- Unauthorized vehicle access
- Bypass of authentication mechanisms
- Replay of privileged commands

3. Denial of Service (DoS)

Description: Prevent legitimate CAN communication through bus saturation or error injection.

Attack methods:

- Bus flooding: Transmit continuous high-priority frames
- Error frame injection: Force bus-off state in target ECUs
- Bit manipulation: Corrupt frames to trigger retransmissions

Impact:

- ECU communication failure
- Safety system unavailability
- Vehicle operational disruption

4. ECU Spoofing

Description: Impersonate legitimate ECU by sending messages with its identifier.

Attack scenarios:

- Masquerade as engine ECU to alter performance
- Impersonate body control module for access control
- Fake sensor ECU to provide false environmental data

Requirements:

- Knowledge of ECU message identifiers
- Ability to inject messages during ECU silence

Impact:

- False sensor readings propagated through system
- Unintended actuator activation
- Safety system compromise

5. Firmware Manipulation

Description: Modify ECU firmware to alter behavior or create persistent backdoors.

Attack methods:

- Exploit diagnostic protocols (UDS, KWP2000)
- Use insecure bootloaders for unauthorized reflashing
- Inject malicious code during software updates

Impact:

- Persistent compromise surviving power cycles
- Difficult-to-detect malicious behavior

- Complete ECU control

6. Time-Based Attacks

Description: Exploit timing characteristics of CAN communication.

Attack scenarios:

- Timing analysis: Infer vehicle state from message timing patterns
- Synchronization attacks: Disrupt time-critical control loops
- Suspension attacks: Delay critical safety messages

D. Documented Real-World Exploits**1. Charlie Miller and Chris Valasek Research (2013-2015)**

Target vehicles: Toyota Prius, Ford Escape, Jeep Cherokee

Attack method:

- Exploited cellular connectivity in Uconnect infotainment system
- Remote code execution on infotainment ECU
- Pivoted from infotainment to CAN bus
- Sent CAN messages to control steering, braking, acceleration

Impact:

- First demonstrated remote wireless vehicle control
- Led to 1.4 million vehicle recall by Fiat Chrysler
- Highlighted insufficient network segmentation

Reference: [13][14]

2. Keen Security Lab Tesla Model S Research (2016)**Attack vector:**

- Exploited vulnerability in browser-based infotainment system
- Achieved remote root access
- Manipulated vehicle functions via CAN messages

Demonstrated capabilities:

- Remote door lock/unlock
- Display manipulation
- Brake control during low-speed operation

Manufacturer response:

- Over-the-air security patch deployed
- Bug bounty program enhancement

Reference: [15]

3. Relay Attacks on Keyless Entry Systems**Attack technique:**

- Use two devices to relay signals between key fob and vehicle
- Amplify weak signals to extend effective range
- Bypass proximity-based authentication

Affected systems:

- Most keyless entry/start systems vulnerable
- Attack demonstrated on multiple manufacturers

Impact:

- Vehicle theft without physical key
- Insurance and law enforcement challenges

Reference: [16]

4. TPMS Sensor Exploits**Vulnerability:**

- Tire Pressure Monitoring System sensors transmit unencrypted data
- Fixed or predictable sensor IDs

Attack demonstrations:

- False pressure warnings injection
- Sensor ID tracking for vehicle surveillance
- Denial-of-service through flooding

Reference: [17]

5. OBD-II Malware**Attack vector:**

- Malicious dongle inserted into OBD-II port
- Executes arbitrary CAN commands
- Can disable security systems, steal vehicle data

Demonstrated capabilities:

- ECU firmware extraction
- CAN bus fuzzing and manipulation
- Covert data exfiltration

E. Attack Taxonomy and Classification**By attack vector:**

1. Physical access attacks: OBD-II, USB, physical ECU tampering
2. Short-range wireless: Bluetooth, Wi-Fi, NFC, RFID
3. Long-range wireless: Cellular, satellite, V2X
4. Supply chain: Pre-installed malware, compromised components

By target layer:

1. Physical layer: Signal jamming, bus manipulation
2. Data link layer: Frame injection, arbitration manipulation
3. Application layer: Protocol exploitation, logic flaws
4. Cross-layer: Combined attacks across multiple layers

By objective:

1. Information gathering: Eavesdropping, traffic analysis, privacy invasion
2. Disruption: Denial of service, ECU disabling
3. Control: Steering, braking, acceleration manipulation
4. Theft: Vehicle stealing, component theft, data theft

F. Threat Modeling Frameworks**1. STRIDE Model Application**

- Spoofing: ECU impersonation, message source falsification
- Tampering: Firmware modification, data corruption
- Repudiation: Inability to trace malicious actions
- Information Disclosure: Eavesdropping on CAN traffic
- Denial of Service: Bus flooding, error injection
- Elevation of Privilege: Gaining unauthorized ECU access

2. CVSS Scoring for Automotive

Adaptation of Common Vulnerability Scoring System for automotive context considering:

- Safety impact (injury, fatality risk)
- Attack complexity and required access
- Affected population size
- Exploit public availability

3. HEAVENS Methodology

Heuristics for Automotive Security Assessment:

- Identify assets (ECUs, data, functions)

- Define security goals (confidentiality, integrity, availability)
- Threat analysis using attack trees
- Risk assessment and prioritization
- Security requirements derivation

5. DEFENSE MECHANISMS AND COUNTER MEASURES**A. Cryptographic Authentication****1. Message Authentication Codes (MAC)**

Concept: Append cryptographic tag to CAN messages for integrity and authenticity verification.

HMAC-Based Approaches:

- HMAC-SHA256: Strong security but computational overhead
- HMAC-SHA1: Moderate security, faster computation
- Truncated MAC: Reduced tag size fits within CAN constraints

Implementation challenges:

- Payload reduction: MAC consumes data bytes (e.g., 4-8 bytes)
- Key management: Secure storage and distribution of symmetric keys
- Computational overhead: Real-time constraints on resource-limited ECUs
- Backward compatibility: Integration with legacy systems

Example:

For 8-byte CAN frame with 4-byte MAC:

- Available payload: 4 bytes
- MAC computed over identifier + data
- Receiver verifies MAC before processing message

Research contributions:

- CANAuth: Lightweight authentication protocol for CAN [18]
- LiBrA-CAN: Hash-based authentication with minimal overhead [19]
- vatiCAN: HMAC with selective authentication [20]

2. Asymmetric Cryptography

Public-Key Infrastructure (PKI) for Automotive:

- Digital signatures: Verify software update authenticity
- Certificate-based authentication: ECU identity verification
- Key establishment: Secure session key exchange

Elliptic Curve Cryptography (ECC):

- Smaller key sizes than RSA (256-bit ECC $\approx$ 3072-bit RSA)
- Lower computational and storage requirements
- Suitable for resource-constrained ECUs

Applications:

- Secure boot: Verify firmware signature before execution
- Over-the-air updates: Authenticate update packages
- V2X communication: Message signing and verification

Challenges:

- Signature size exceeds single CAN frame

- Computational latency impacts real-time performance
- Complex key management infrastructure

3. Lightweight Cryptography

NIST Lightweight Cryptography Initiative:

Developing algorithms optimized for constrained devices:

- ASCON: Authenticated encryption, low resource usage
- TinyJAMBU: Efficient for 8-bit microcontrollers
- Grain-128AEAD: Stream cipher with authentication

Automotive-specific approaches:

- Hardware acceleration for crypto operations
- Co-processor integration in ECU architectures
- Optimized implementations for automotive MCUs

B. Intrusion Detection Systems

1. Anomaly-Based Detection

Concept: Establish normal CAN traffic patterns and detect deviations.

Detection methods:

a) Frequency-based detection:

- Monitor message transmission intervals
- Detect deviations from expected periodicity
- Identify injection attacks through timing anomalies

b) Sequence-based detection:

- Learn typical message sequences
- Detect out-of-sequence or unexpected messages
- Use state machines or Markov models

c) Value-based detection:

- Monitor sensor value ranges and correlations
- Detect physically impossible or inconsistent values
- Apply fuzzy logic or statistical methods

d) Entropy-based detection:

- Measure randomness in CAN traffic
- Detect flooding through entropy changes
- Identify abnormal traffic distributions

Machine learning approaches:

- Neural networks: Deep learning for complex pattern recognition
- Decision trees: Rule-based classification
- Support Vector Machines (SVM): Binary classification of normal/attack
- Clustering: Unsupervised learning of traffic patterns

Research contributions:

- INDRA: Neural network-based intrusion detection [21]
- OTIDS: One-time signature-based detection [22]
- LSTM-based IDS: Recurrent neural networks for sequence analysis [23]

2. Specification-Based Detection

Concept: Define formal specifications of correct CAN behavior and detect violations.

Specification methods:

- Finite state machines: Model ECU state transitions
- Temporal logic: Express timing and ordering constraints
- Protocol specifications: Define allowed message formats and sequences

Advantages:

- Low false-positive rate
- Explainable detection decisions
- No training phase required

Challenges:

- Requires complete and accurate specifications
- Manual specification effort
- Difficulty handling legitimate variations

3. Hybrid Detection Approaches

Combination of anomaly and specification-based methods:

- Use specifications for known attack patterns
- Apply anomaly detection for unknown threats
- Reduce false positives through correlation

C. Network Segmentation and Gateways

1. Security Gateway Architecture

Concept: Isolate critical and non-critical network segments through gateway ECUs with filtering and validation.

Gateway functions:

- Message filtering: Block unauthorized messages between segments
- Protocol translation: Convert between CAN, CAN FD, Ethernet
- Rate limiting: Prevent flooding attacks
- Authentication enforcement: Verify message sources
- Logging and monitoring: Record suspicious activities

Typical segmentation:

- Powertrain domain: Engine, transmission, critical control
- Chassis domain: Braking, steering, suspension
- Body domain: Lighting, climate, comfort
- Infotainment domain: Display, connectivity, entertainment
- ADAS domain: Cameras, radar, sensor fusion

2. Intrusion Prevention Systems (IPS)

Active defense mechanisms:

- Frame dropping: Discard suspicious messages
- Source isolation: Disconnect compromised ECU from bus
- Rate limiting: Throttle excessive traffic
- Challenge-response: Verify ECU identity on demand

D. Hardware Security Modules (HSM)

1. Secure Crypto Processors

Functionality:

- Dedicated cryptographic accelerators
- Secure key storage (tamper-resistant)
- Random number generation
- Secure boot and firmware verification

Automotive HSM features:

- ISO 26262 ASIL-D compliance for safety
- AES, SHA, ECC hardware acceleration
- Protected memory regions
- Side-channel attack resistance

Common implementations:

- Infineon AURIX: Automotive microcontrollers with integrated HSM
- NXP S32: Secure MCU family with crypto support

- Renesas RH850: Automotive MCUs with security features

2. Trusted Platform Module (TPM)

Applications:

- Secure storage of cryptographic keys
- Device identity and attestation
- Measured boot process
- Secure firmware updates

E. Secure Software Update Mechanisms

1. Over-the-Air (OTA) Update Security

Security requirements:

- Authentication: Verify update package origin
- Integrity: Detect unauthorized modifications
- Confidentiality: Protect proprietary software
- Rollback protection: Prevent downgrade attacks
- Atomicity: Ensure complete or no update (no partial state)

Implementation approach:

1. Digital signatures: Sign update packages with manufacturer private key
2. Encryption: Encrypt update to protect IP
3. Secure boot: Verify firmware signature before execution
4. Version management: Enforce monotonic version numbers
5. Fail-safe mechanisms: Maintain backup image for recovery

2. AUTOSAR Secure Onboard Communication

SecOC (Secure Onboard Communication) Module:

- Provides message authentication for AUTOSAR COM stack
- Supports MAC generation and verification
- Integrates with Crypto Service Manager (CSM)
- Freshness value management to prevent replay attacks

Freshness mechanisms:

- Counter-based: Monotonically increasing counter appended to messages
- Timestamp-based: System time included in MAC calculation
- Challenge-response: Random nonce for dynamic authentication

F. Physical Security Measures

1. Secure Boot

Process:

1. Boot ROM: Immutable first-stage bootloader in MCU
2. Signature verification: Check bootloader integrity
3. Chain of trust: Each stage verifies the next
4. Execution prevention: Halt if verification fails

Benefits:

- Prevents unauthorized firmware execution
- Establishes trusted execution environment
- Foundation for runtime security

2. Debug Port Protection

Measures:

- Disable JTAG/SWD interfaces in production
- Password-protected debug access

- Debug port fuses for permanent disabling
- Secure debug authentication

3. Tamper Detection

Mechanisms:

- Physical tamper switches
- Voltage glitch detection
- Clock frequency monitoring
- Temperature sensors for attack detection

Response actions:

- Erase cryptographic keys
- Enter safe mode
- Log tamper event
- Notify remote security operations center

6. STANDARDS AND REGULATORY FRAMEWORKS

A. ISO Standards

1. ISO 11898 Series

ISO 11898-1: CAN data link layer and logical link control

ISO 11898-2: High-speed CAN physical layer

ISO 11898-3: Low-speed fault-tolerant physical layer

2. ISO/SAE 21434 - Automotive Cybersecurity

Published: 2021

Scope: Road vehicles cybersecurity engineering

Key requirements:

a) Organizational cybersecurity management:

- Cybersecurity governance structure
- Risk management framework
- Continuous monitoring and improvement

b) Project-dependent cybersecurity activities:

- Threat analysis and risk assessment (TARA)
- Security by design principles
- Cybersecurity verification and validation
- Incident response procedures

c) Cybersecurity lifecycle phases:

- Concept phase: Security goals definition
- Development phase: Security architecture and implementation
- Production phase: Secure manufacturing and configuration
- Operations and maintenance: Update management, monitoring
- Decommissioning: Secure disposal and data erasure

d) Risk assessment methodology:

- Asset identification
- Threat scenario analysis
- Impact rating (safety, financial, operational, privacy)
- Attack feasibility assessment
- Risk determination and treatment

3. ISO 26262 - Functional Safety

Safety-security interaction:

- Security vulnerabilities can impact functional safety
- Coordinated safety and security analysis required
- ASIL (Automotive Safety Integrity Level) considerations for security components

B. SAE Standards

1. SAE J1939

High-layer protocol for commercial vehicles and heavy-duty trucks on CAN bus.

2. SAE J3061 - Cybersecurity Guidebook

Published: 2016 (superseded by ISO/SAE 21434)

Contributions:

- Cybersecurity process framework
- Threat modeling guidance
- Verification and validation methods

3. SAE J2945/1 - DSRC Security

Dedicated Short-Range Communication (DSRC) security:

- Message authentication for V2V communication
- Certificate management
- Privacy preservation through pseudonym certificates

C. AUTOSAR Security Architecture**1. Classic AUTOSAR Security Modules**

Crypto Service Manager (CSM):

- Unified interface for cryptographic operations
- Hardware abstraction for HSM/TPM
- Support for encryption, signing, hashing, RNG

Secure Onboard Communication (SecOC):

- Message authentication for inter-ECU communication
- Freshness value management
- Integration with AUTOSAR COM stack

Crypto Stack:

- Cryptographic primitives implementation
- Interface to hardware acceleration
- Key management services

2. Adaptive AUTOSAR Security

Identity and Access Management (IAM):

- Certificate-based ECU authentication
- Access control for services and resources

Transport Layer Security (TLS):

- Secure communication over Ethernet
- Mutual authentication
- Encrypted data transmission

Secure Software Update:

- Authenticated update packages
- Rollback protection
- Atomic installation

3. Security Best Practices

Secure coding guidelines:

- Input validation and sanitization
- Memory safety (buffer overflows prevention)
- Integer overflow protection
- Secure random number generation

Vulnerability management:

- Security testing and code review
- Penetration testing
- Fuzzing and automated analysis

D. Regulatory Requirements**1. UNECE WP.29 Regulations**

UN R155 - Cybersecurity:

Effective: July 2024 (for new vehicle types)

Requirements:

- Cybersecurity management system (CSMS)
- Risk assessment and mitigation

- Cybersecurity by design
- Testing and verification
- Software update management
- Monitoring and incident response

UN R156 - Software Update:

Requirements:

- Software Update Management System (SUMS)
- Secure update processes
- Type approval for update procedures

2. Regional Regulations

European Union:

- General Data Protection Regulation (GDPR): Vehicle data privacy
- Type approval requirements including cybersecurity
- Market surveillance and enforcement

United States:

- NHTSA guidelines for automotive cybersecurity
- FTC Act: Consumer protection and data security
- State-level legislation (e.g., California SB-327)

China:

- GB standards for automotive cybersecurity
- Multi-Level Protection Scheme (MLPS)
- Data localization requirements

E. Industry Initiatives**1. Auto-ISAC (Automotive Information Sharing and Analysis Center)**

Mission: Facilitate threat intelligence sharing among automotive manufacturers and suppliers.

Activities:

- Cybersecurity incident reporting
- Threat analysis and advisories
- Best practices development
- Collaboration with law enforcement

2. EVITA (E-Safety Vehicle Intrusion Protected Applications)

European research project defining security architecture for automotive systems.

Contributions:

- Hardware security modules specification
- Security levels for ECUs (Full/Medium/Light)
- Secure communication protocols

3. PRESERVE

European project focusing on:

- Security and privacy in V2X
- Pseudonym certificate management
- Misbehavior detection

7. RESEARCH CHALLENGES AND FUTURE DIRECTIONS**A. Technical Challenges****1. Real-Time Security**

Problem: Balancing cryptographic security with hard real-time constraints.

Research directions:

- Hardware-accelerated cryptography optimized for automotive workloads
- Deterministic cryptographic operations with bounded execution time

- Selective authentication based on message criticality
- Pre-computation and caching strategies

2. Lightweight Cryptography

Problem: Resource-constrained ECUs with limited CPU, memory, and power.

Research directions:

- ASCON and other NIST lightweight finalists evaluation for automotive
- Specialized crypto co-processors for low-end ECUs
- Energy-efficient cryptographic implementations
- Trade-offs between security strength and resource usage

3. Key Management at Scale

Problem: Managing cryptographic keys across hundreds of ECUs, multiple suppliers, vehicle lifetime.

Research directions:

- Hierarchical key derivation schemes
- Distributed key management without central authority
- Key rotation strategies minimizing downtime
- Post-quantum key establishment protocols

4. Intrusion Detection Accuracy

Problem: High false-positive rates disrupt vehicle operation; false-negatives allow attacks.

Research directions:

- Explainable AI for intrusion detection
- Federated learning for cross-vehicle threat intelligence
- Context-aware detection considering driving scenarios
- Adaptive thresholds based on operational mode

B. Standardization Gaps

1. Unified Security Architecture

Current state: Fragmented approaches across OEMs and regions.

Needed:

- Harmonized security architecture reference model
- Interoperability standards for security components
- Common security API for ECU suppliers

2. Vulnerability Disclosure

Current state: Inconsistent vulnerability handling processes.

Needed:

- Standardized CVE-like system for automotive
- Coordinated disclosure timelines
- Patch management best practices
- Cross-industry vulnerability database

3. Security Testing and Certification

Current state: Limited standardized testing methodologies.

Needed:

- Automotive penetration testing frameworks
- Security certification schemes
- Third-party security assessment criteria
- Continuous monitoring and re-certification

C. Privacy and Data Protection

1. Connected Vehicle Data

Challenges:

- Location tracking and movement pattern analysis

- Driver behavior profiling
- In-cabin monitoring (camera, microphone)
- V2X broadcast message correlation

Research directions:

- Differential privacy for trajectory data
- Anonymous credential systems for V2X
- On-device data processing to minimize collection
- User-controlled data sharing preferences

2. GDPR and Regional Compliance

Challenges:

- Right to be forgotten vs. safety-critical logging
- Data minimization vs. diagnostic requirements
- Cross-border data transfer restrictions
- Consent management in shared vehicles

D. Human Factors

1. Usable Security

Challenges:

- Complex security indicators confuse drivers
- Security updates require user action
- Alert fatigue from false warnings

Research directions:

- Intuitive security status visualization
- Automated security update with minimal user interaction
- Adaptive alerting based on severity and context

2. Insider Threats

Challenges:

- Service technicians with legitimate access
- Aftermarket modifications bypassing security
- Employee sabotage or negligence

Research directions:

- Role-based access control with audit logging
- Anomaly detection for service procedures
- Tamper-evident seals and monitoring
- Employee security awareness training

E. Economic and Business Challenges

1. Cost-Benefit Analysis

Challenge: Justifying security investments against unclear threat probabilities.

Research directions:

- Risk quantification models for automotive cybersecurity
- Insurance-based approaches to cyber risk transfer
- Cost-effective security architecture optimization

2. Supply Chain Complexity

Challenge: Ensuring security across multi-tier global supply chains.

Research directions:

- Hardware root of trust for component authentication
- Blockchain-based supply chain transparency
- Third-party security audits and certification
- Contractual security requirements enforcement

F. Recommended Research Priorities

Based on the comprehensive analysis, the following research areas are prioritized:

High Priority

1. Lightweight authenticated encryption for resource-constrained ECUs
2. Machine learning-based intrusion detection with low false-positive rates
3. Secure OTA update mechanisms with fail-safe and rollback protection
4. V2X security and privacy preserving protocols
5. Security validation and testing methodologies
Medium Priority
6. Quantum-resistant cryptography migration path for automotive
7. Hardware security module standardization and cost reduction
8. Secure Ethernet/TSN integration with legacy CAN networks
9. Incident response and forensics capabilities
10. Privacy-preserving analytics for vehicle data Long-Term Research
11. Autonomous vehicle security against sensor and AI attacks
12. Blockchain applications for vehicle lifecycle management
13. Homomorphic encryption for privacy-preserving diagnostics
14. Secure multi-party computation for collaborative ADAS
15. Bio-inspired security mechanisms for adaptive defense

8.CONCLUSION

This comprehensive survey has systematically examined the evolution, capabilities, limitations, and security challenges of CAN and CAN FD protocols in automotive systems. The analysis reveals that while these communication technologies have successfully supported the automotive industry for decades, the transition to connected, autonomous, and software-defined vehicles introduces unprecedented cybersecurity challenges requiring multi-layered defense strategies.

A. Key Findings

1. CAN Protocol Maturity and Limitations

Classical CAN protocol provides robust, deterministic, and cost-effective communication suitable for distributed automotive control systems. However, its fundamental limitations—particularly the 8-byte payload restriction, 1 Mbps maximum bandwidth, and complete absence of security features—increasingly constrain modern vehicle architectures requiring high-bandwidth sensor data exchange and secure communication.

2. CAN FD as Bandwidth Solution

CAN FD successfully addresses payload and bandwidth limitations through 64-byte frames and dual bit-rate transmission achieving up to 8 Mbps during the data phase. This evolution enables more efficient flash programming, diagnostic data transfer, and reduced frame overhead. However, CAN FD does not inherently resolve security vulnerabilities and still lacks native authentication, encryption, or access control mechanisms.

3. Expanding Threat Landscape

Real-world demonstrations by security researchers have conclusively shown that modern vehicles are vulnerable to remote exploitation, message injection, ECU spoofing, and safety-critical function manipulation. The increasing connectivity through cellular, Wi-Fi, Bluetooth, and V2X interfaces exponentially expands the attack surface beyond traditional physical access vectors. Nation-state actors, cybercriminals, and sophisticated attackers now view vehicles as viable targets for sabotage, theft, surveillance, and ransomware.

4. Multi-Layered Defense Necessity

No single security technology provides comprehensive protection. Effective automotive cybersecurity requires coordinated deployment of:

- Cryptographic authentication to verify message sources and integrity
- Intrusion detection systems to identify anomalous behavior
- Network segmentation to isolate critical domains
- Secure gateways to enforce communication policies
- Hardware security modules to protect cryptographic keys
- Secure boot and update mechanisms to ensure firmware integrity

5. Standardization Progress

The publication of ISO/SAE 21434 and UNECE WP.29 regulations represents significant progress toward harmonized automotive cybersecurity practices. These frameworks mandate risk-based security engineering throughout the vehicle lifecycle, from concept through decommissioning. However, implementation heterogeneity across manufacturers and regions creates interoperability challenges and potential security gaps.

6. Emerging Technology Integration

The automotive industry is transitioning toward centralized compute architectures, Automotive Ethernet backbones, and software-defined vehicles. While these technologies offer improved performance and flexibility, they introduce new security challenges related to virtualization, containerization, over-the-air updates, and complex software supply chains. Successful integration requires security-by-design principles applied from initial architecture definition.

B. Critical Insights

1. Security-Safety Interaction: Now ultimately the Automotive cybersecurity cannot be addressed independently from functional safety. Security vulnerabilities directly impact safety-critical systems, and safety mechanisms can be defeated through cyber-attacks. ISO 26262 and ISO/SAE 21434 must be applied in coordinated fashion.
2. Legacy System Challenge: The 10-15 year vehicle operational lifetime creates a "security debt" where vehicles designed with minimal security remain vulnerable throughout their service life. Retrofit security solutions and secure gateway integration are needed to protect legacy fleets.

3. Resource Constraints: Unlike IT systems where performance overhead is acceptable, automotive embedded systems operate under strict real-time, cost, power, and environmental constraints. Security solutions must be optimized for these limitations without compromising protection effectiveness.
4. Privacy-Security Balance: Connected vehicles generate vast amounts of personal data. Security mechanisms must protect this data while respecting privacy regulations and user preferences. Transparent data governance and user control are essential for trust.
5. Continuous Evolution: The threat landscape evolves continuously as attackers develop new techniques. Automotive security cannot be a one-time design exercise but requires ongoing monitoring, threat intelligence sharing, vulnerability management, and defensive updates throughout the vehicle lifecycle.

REFERENCES

- [1] S. Checkoway et al., "Comprehensive experimental analyses of automotive attack surfaces," in *Proceedings of USENIX Security Symposium*, 2011, pp. 77–92.
- [2] R. Bosch GmbH, "CAN Specification Version 2.0," Robert Bosch GmbH, Stuttgart, Germany, 1991.
- [3] K. Etschberger, *Controller Area Network: Basics, Protocols, Chips and Applications*. Germany: IXXAT Press, 2001.
- [4] K. Koscher et al., "Experimental security analysis of a modern automobile," in *Proceedings of IEEE Symposium on Security and Privacy*, 2010, pp. 447–462.
- [5] F. Hartwich, "CAN with Flexible Data-Rate," in *Proceedings of International CAN Conference*, 2012.
- [6] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, 2015.
- [7] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *Black Hat USA*, 2014.
- [8] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 2nd ed. Springer, 2011.
- [9] N. Navet et al., "Trends in automotive communication systems," *Proceedings of IEEE*, vol. 93, no. 6, pp. 1204–1223, 2005.
- [10] ISO 11898-1:2015, "Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling," International Organization for Standardization.
- [11] W. Zimmermann and R. Schmidgall, *Bussysteme in der Fahrzeugtechnik*, 5th ed. Springer Vieweg, 2014.
- [12] M. Boroujeni et al., "CAN FD flexible data-rate CAN," *Robert Bosch GmbH*, 2012.
- [13] C. Miller and C. Valasek, "Adventures in automotive networks and control units," *DEF CON*, vol. 21, pp. 260–264, 2013.
- [14] A. Greenberg, "Hackers remotely kill a Jeep on the highway—with me in it," *Wired*, 2015.
- [15] Keen Security Lab, "Experimental security assessment of BMW cars," Tencent, 2018.
- [16] A. Francillon, B. Danev, and S. Capkun, "Relay attacks on passive keyless entry and start systems in modern cars," in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2011.
- [17] I. Rouf et al., "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *Proceedings of USENIX Security Symposium*, 2010, pp. 323–338.
- [18] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars (ESCAR)*, 2004.
- [19] B. Groza, S. Murvay, A. van Herrewege, and I. Verbauwhede, "LiBrA-CAN: Lightweight broadcast authentication for controller area networks," *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 3, 2017.
- [20] P. Mundhenk, A. Paverd, A. Mrowca, S. Steinhorst, M. Lukasiewicz, S. A. Fahmy, and S. Chakraborty, "Security in automotive networks: Lightweight authentication and authorization," *ACM Transactions on Design Automation of Electronic Systems*, vol. 22, no. 2, 2017.
- [21] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *Proceedings of IEEE International Conference on Cyber Security and Cloud Computing*, 2016, pp. 36–41.
- [22] M. Muter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2011, pp. 1110–1115.
- [23] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "CANet: An unsupervised intrusion detection system for high dimensional CAN bus data," *IEEE Access*, vol. 8, pp. 58194–58205, 2020.
- [24] ISO/SAE 21434:2021, "Road vehicles — Cybersecurity engineering," International Organization for Standardization and SAE International, 2021.
- [25] UNECE WP.29, "UN Regulation No. 155 – Cyber security and cyber security management system," United Nations Economic Commission for Europe, 2021.
- [26] AUTOSAR, "Specification of Secure Onboard Communication," AUTOSAR Release 4.4.0, 2018.
- [27] S. Woo et al., "A practical wireless attack on the connected car and security protocol for in-vehicle CAN," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993–1006, 2015.
- [28] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in *Proceedings of World Congress on Industrial Control Systems Security*, 2015, pp. 45–49.
- [29] W. Choi, H. Jo, S. Woo, J. Chun, J. Park, and D. Lee, "Identifying ECUs using inimitable characteristics of signals in controller area networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4757–4770, 2018.
- [30] Q. Wang et al., "VeCure: A practical security framework to protect the CAN bus of vehicles," in *Proceedings of International Conference on the Internet of Things (IoT)*, 2017, pp. 13–20.