2018 1st International Conference on Data Intelligence and Security

CAN Authorization Using Message Priority Bit-Level Access Control

Adam J. Brown*, Todd R. Andel*, Mark Yampolskiy*, and J. Todd McDonald*
* School of Computing
University of South Alabama
Mobile, Alabama 36688
Email: abrown@jagmail.southalabama.edu

***Abstract*—The controller area network (CAN) is widely used to interconnect electronic components of cyber-physical systems, such as automobiles. It was designed to suffice near-real-time requirements, which makes it an attractive choice for the transportation industry. However, CAN lacks means to authenticate transmitted messages and as such is vulnerable to a variety of attacks. In this paper, we consider a scenario in which one node is compromised and is sending illicit messages across the bus. We propose a solution that leverages the existing CAN specifications by employing the standards error handling and fault confinement properties to enforce authorization. The proposed solution limits the efficacy of an adversarial node on the network, without negatively affecting bit time during normal operations or compliance with the base standard. Our analysis shows that a malicious node can be identified and denied access to the bus after 37 consecutive messages. For even the slowest bus speeds of 10kbps, our solution reduces the downtime experienced from a denial of service attack to be less than one second.**

***Index Terms*—controller area network, CAN, authorization, access control, security, robustness, resilience, automobile**

I. Introduction

Within the transportation industry, interconnected components of automobiles and similar vehicles communicate using a controller area network (CAN). Features of the CAN standard enable the protocol to ensure near-real-time communications, and robust means for error detection ensure high fidelity conveyances of information [1]. The standard is attractive to automobile manufacturers not only for its near-real-time capabilities but also for its simplification of network topology and for its reduction of wiring required through use of a broadcast paradigm over a shared bus. To minimize delays in the transmission of a message, CAN nodes have equal access to the bus and determine priority in the event of a collision through a non-destructive arbitration process. Arbitration, however, does not occur based on the addressing specifics of the message but on the content of the message itself. This feature permits CAN frames to be more lightweight by not requiring the inclusion of the addresses of the source or destination.

While the lack of direct addressing allows CAN overall faster transmissions, it prevents the standard from having a means to authenticate the source of a broadcasted message [2]–[6]. Furthermore, automobiles have been shown to be vulnerable to an adversary compromising a device using both wired and wireless access [7], [8]. Universal master-level access to a bus creates a network built on trust and not authentication, potentially rendering the CAN-enabled system exposed to adversarial manipulations. Each node, if compromised and commandeered by an adversary, may be used at the application layer to transmit any messages. This style of attack resembles a masquerade-style attack. In motor vehicles, nodes that can be remotely accessed over wireless networks raise the overall risk of the networked system by lowering access barriers for an attack. Once an adversary has access to any node, messages can be broadcasted freely across the bus, and to the extent that the system can physically act upon those messages, system operability deteriorates [7]–[9].

In vehicles, any solution to the outlined problem is restricted by its impact on the communication timing, because timely and correct communication is required to maintain functionality. Further, due to the wide-spread adoption of CAN protocol, a solution is needed that can be retrofitted into systems in use; this call for a solution that would not modify the structure of the CAN frame. In this paper, we propose a solution that satisfies these constraints for a scenario of a single malicious node on a CAN bus that sends illicit messages.

The remainder of the paper is structured as follows: Section 2 discusses existing literature into the vulnerability of CAN, the capabilities of attackers, and proposed measures for redress; Section 3 provides background into CAN and details properties of the standard relevant to the solution in this paper; Section 4 frames the existing threat, proposed solutions, and the assumptions and limitations of this solution; Section 5 details the proposal with its associated gains and costs; Section 6 discusses design decisions; and Section 7 states the conclusion and intended future work.

II. Related Works

The need for protection from messages sent by adversarial sources through a compromised node has been recognized and actively discussed by the research community. Hoppe et al. [2] employed a CERT taxonomy to analyze the CAN bus system. Classifying attack scenarios aimed at automobiles, the authors identified the lack of authentication as a means for exploitation by malicious actors. Similarly, Kleberger et al. [3] constructed a taxonomy of CAN for automobiles and identified security flaws and features within the standard. Among the flaws, the authors cited the ability to use the lack of authentication for

978-1-5386-5762-1/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDIS.2018.00008

an adversary to reprogram CAN nodes. Larson et al. [9] cited the potential for an adversary with illicit access to a node to effectuate a large range of attacks.

Checkoway et al. [7], following a threat analysis of an automobile, explored a wide variety of internal and external attack vectors to gain illicit access to the in-vehicle CAN. Having successfully accessed a node, the authors proceeded to demonstrate full control over the system as a whole. Koscher et al. [8] built upon that research and effectuated additional attacks against automobiles. In a test on a runway, the authors demonstrated full control over vehicular operations in a multitude of their injection attacks. Denial of service attacks were demonstrated to be capable of rendering an automobile unresponsive to the extent such that the vehicle could be prevented from turning on or off. Foster and Koscher [10] continued their previous work and identified additional capabilities of an attacker having compromised a node.

Carsten et al. [11] identified attack methods that exploit the lack of direct addressing information in CAN frames. The listed adversarial actions ranged from message falsification to system degradation and override. The authors then assessed a variety of proposed alternatives addressing the vulnerability of a CAN-enabled system acting on illicit messages originating from a compromised node. Lin et al. [4] modeled attacks effective against CAN and identified properties of the standard which pose challenges when addressing the vectors.

Addressing one of these concerns, Carsten et al. [12] constructed a hashed message authenticator partially comprised from a timing value using a universal clock of their devising. The alternative would fortify an automobile against falsified and replayed CAN frames, but the authors did not explore the specific timing delays which would be introduced in the adoption of the system and acknowledged that the system may not be feasible in practice with existing bus speeds.

Ueda et al. [3] proposed the calculation of a message authentication code from an encryption key and incorporating that value into the existing data field of a CAN frame. A special purpose controller node would authenticate the message based on the source and the code within the data frame. Messages failing authentication would raise an error flag. The proposed solution, while relatively lightweight and conforming to the frame size limits of the CAN standard, increased bit delay as the number of connected nodes increased and would likely be unsuitable for larger networks.

Lin et al. [13] proposed an integrated mixed integer linear programming algorithm to address design-level vulnerabilities in CAN. The optimization formulation proposed while more efficient than a greedy heuristic in a previous study nonetheless introduced timing delays for each broadcast. Groza and Murvay [6] encountered similar delays as well as minor synchronization errors in their implementation of a modified protocol with authentication. Patsakis et al. [14] constructed an altered CAN protocol equipped with mutual authentication based on a ticketing system. While affording a more distributed security architecture, the system, if implemented, would nonetheless experience delays in broadcasts across the bus due to the additional authentication requirements.

III. Background: Controller Area Network

A. CAN Frame

While the specific format of the CAN frame, or message, differs depending on the physical layer, the ordering of the field classes within the message conform to that of the Classical Base Frame. Following a dominant bit denoting the start of the frame, a message contains an arbitration field with an identifier, a control field with meta information, a data field with the payload, a cyclic redundancy check field to detect certain errors, and an acknowledgement field used to verify receipt of an uncorrupted frame. By default, bits within fields of length greater than one are ordered in big endian fashion.

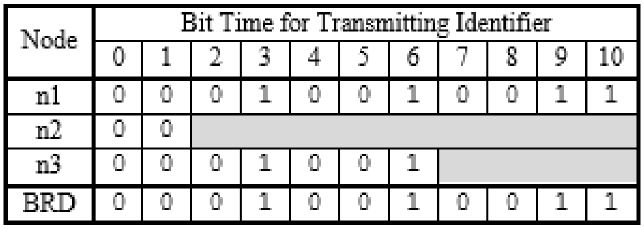
The arbitration field of a CAN frame is comprised of at least two pieces of information: an identifier of the type of message being sent and a bit that designates whether the packet is soliciting information from another node on the network. CAN messages do not have space allocated for source and destination information. Broadcasting is employed over direct addressing to increase overall communication speed and to reduce the network traffic footprint caused by each frame sent. Additionally, having each node receive every broadcasted frame allows for more robust error detection.

B. CAN Properties

1) Carrier Sense Multiple Access: The media access control protocol for the CAN standard is a carrier-sense, multiple access protocol with collision detection and arbitration based on message priority. Carrier-sense multiple access means that, for each node attempting to communicate across the shared bus, the node must first verify that there is no other traffic. Once the bus is confirmed idle, the node can attempt to broadcast its message. Simultaneous attempts to access the bus result in traffic collisions which trigger content-based arbitration. A node that wins arbitration remains a transmitter until either the bus becomes idle again or until the node loses arbitration.

2) Collision Detection: Because each node on the CAN bus operates as a bus master, it is possible that more than one node will begin transmitting a frame simultaneously. To enhance the performance of the protocol, in the event of a collision, any node in conflict with the node with a message of higher priority on the bus terminates transmission to wait for a bus idle state before attempting to transmit again. Because the message of greater priority persists, there is a shorter overall wait time before a defeated node can reattempt to transmit.

3) Bitwise Arbitration: In order for the CAN standard to be utilized in real-time environments, detected collisions are resolved in a manner that satisfies the timing requirements of the system. A standard with multi-master access and no form of direct addressing, however, lacks the qualified means to arbitrate based on the priority of the source. Therefore, to avoid circumstances in which system critical communications can be delayed for indeterminate lengths of time, the CAN standard employs bitwise arbitration based on message priority. Rather than assign priority to nodes sending messages, the CAN bus arbitrates based on the content of the frame sent.



Node	Bit Time for Transmitting Identifier										
	0	1	2	3	4	5	6	7	8	9	10
n1	0	0	0	1	0	0	1	0	0	1	1
n2	0	0									
n3	0	0	0	1	0	0	1				
BRD	0	0	0	1	0	0	1	0	0	1	1

Fig. 1: CAN Arbitration Example

Arbitration in CAN is non-destructive; a node that wins arbitration experiences no disruption in transmission. A node which loses arbitration by sending a logical low before any other simultaneously transmitting node ceases to transmit at that point and becomes a receiver of the resultant broadcast. The broadcast, labeled BRD in Figure 1, carries only the highest priority message. In the example, the second and third nodes lost priority at the third and eighth most significant bits respectively. The broadcast transmitted the dominant bit (i.e., ’0’) when there was a conflict thus sacrificing no time despite the collision. That the non-destructive arbitration process occurs without delaying the transmission of the identifier in nominal bit time is a prominent feature as to the suitability of the CAN standard for real-time system communications.

4) Error Detection: Despite timing capabilities, the CAN protocol would not be appropriate for real-time systems if it lacked the ability to ensure the integrity of messages. To accommodate, the standard provides for five non-mutually exclusive error types detectable by a variety of mechanisms. Of those five, a bit error can be detected by a source node at the bit time at which that node detects a different bit broadcasted than what had been sent, and a form error is triggered when a field contains illegal bits for which there is no tolerance.

5) Error Signaling: A node detecting an error, other than a CRC error, starts an error flag at the next bit. In the event of a CRC error, the error flag is sent following the acknowledgement delimiter in the classical frame. An error frame consists of the error flag followed by the error delimiter. Comprised of six consecutive bits of the same type, the error flag is active if using dominant bits and passive if using recessive bits. In response to an error flag, the bus enters either an error-active or an error-passive state depending on the flag raised. The error delimiter is a sequence of eight recessive bits.

6) Fault Confinement: To reduce bus errors over time, the CAN standard employs a confinement strategy to identify and isolate allegedly faulty nodes. Fault confinement not only promotes message integrity over time but also reduces delays during error flagging and retransmission. Each node possesses an accompanying counter for transmit errors and for receive errors which increment in response to errors. When a node properly sends or receives a frame, the respective counter decrements. Error counters of both kinds increase by either one or eight depending on the form and the circumstance of the detected error. Though the specific values by which the counters decrease change depending on bus configuration, error counters are representative of a relative error rate for any given node.

A node for which the transmit error counter exceeds a value of 127 enters an error-passive state. Other nodes along the bus are notified of the event by sending an active error flag by the node changing error state. Nodes become error-active again when both error counters reach 127 or less. If the transmit error counter of a node exceeds 255, however, the bus supervisor requests the node enter a bus-off state. A node in this state cannot send frames even in response to data requests. Functionality can be restored to the node pursuant to a restart request followed by a wait period during which the node must monitor 128 occurrences of 11 consecutive recessive bits on the bus before its error counter is reset to zero.

IV. Security Needs

A. Threat Model

The combination of design features that enable the CAN standard to accommodate real-time application requirements also exposes the implemented networks to internal and external threats. Because message headers lack addressing fields, nodes communicating over a CAN bus must trust that structurally valid frames are also legitimate frames. Any bus node can send any structurally valid message. As diagramed in Figure 2, a validly constructed message is broadcasted and acted upon regardless of the message’s original source or intended purpose. Each node therefore becomes a potential point of entry, and an attacker can use the compromised node to perform the actions outlined in Figure 3.

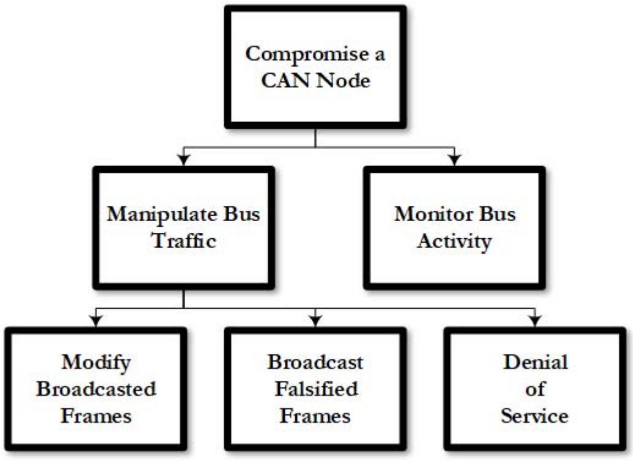


Fig. 3: CAN Threat Taxonomy

Functionality of CAN applications in cyber-physical systems, such as in automobiles, rely on timely and correct information. An attacker having gained access to a CAN node may influence either. Of particular danger are attacks that would

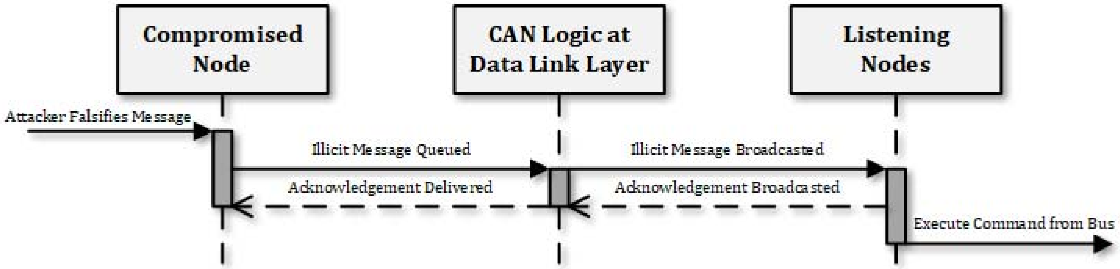

Compromised Node
CAN Logic at Data Link Layer
Listening Nodes
Attacker Falsifies Message
Illicit Message Queued
Acknowledgement Delivered
Illicit Message Broadcasted
Acknowledgement Broadcasted
Execute Command from Bus

Fig. 2: Attack Sequence Against CAN

negatively impact the overall safety of the vehicle. For example, many models of smart cars are equipped with network-capable infotainment centers, which have been demonstrated as vulnerable access points for attackers to directly affect other nodes or to even update supervisor firmware [7], [8].

B. *Threat Implementation*

Merely adding source and destination fields to the CAN frame to allow for direct addressing, however, will not deter adversaries who could readily spoof other nodes. Such a precaution would only add a trivial obstacle to be overcome. Traditional information systems authenticate using MACs, or HMACs if verifying data integrity as well. These measures increase processing time and computational resources, which in automobiles may impair the ability of the system to comply with hard real-time requirements. Timing of communications in vehicles should ideally be minimal and deterministic. Many cryptographic solutions addressing the lack of authentication in CAN have been proposed [4]–[6], but many module manufacturers have not adopted them. This is perhaps due to uncertainty as to the systemic effects of introducing the necessary timing delays to incorporate a given proposed solution into the CAN standard. Security solutions, which are not integrated into the design of a system, stack, rather than share, requirements. Building security into the CAN standard should minimize design tradeoffs. Rather than applying externally developed measures to function parallel to CAN bus operations, we propose a method of authorization that utilizes existing properties of the standard to determine access control and preserves the performance gains afforded by the initial design decisions.

V. Proposed Solution: Real-Time Bitwise Authorization

A. *Data Link Layer Access Control*

Instead of proposing a means of authentication for arguably more comprehensive security, we propose that a means of authorization through access control yields a similar net effect without negatively impacting network broadcast time during normal operations. We propose an access control check at the data link sublayer which invalidates an outgoing message at broadcast time if the message exceeds a predetermined authorization.

Because CAN is already equipped with bitwise content-based arbitration, identifiers are inherently ordered by priority. In an automobile, the priority of a message directly correlates to vehicular operations and safety. By using the predetermined priority levels as proxies for security levels, the access controller only needs to count the sequence of dominant bits and ensure that it is not greater than the expected number. Evaluating the outgoing message at the MAC layer or lower ensures that any external user cannot circumvent the authorization check. At the designated sublayer, the access controller would function utilizing a preset value representing the highest priority of message allowed for the connected device. If higher in the stack than the MAC sublayer, the tolerance value could readily be device-specific whereas if performing the logic close to the physical layer, it could be connector-specific. To preserve non-destructive properties of identifier transmission the device does not take action if the attached node violates the preset security level. Instead, if an illicit message is sent and not interrupted by an error flag indicating that the unauthorized identifier resulted from a benign error, the access controller logic flips the acknowledgement delimiter to a dominant parity which triggers a bit error by the source node and a form error by the listening nodes.

The sequence of bus activity in relation to our proposed access control logic is diagramed in Figure 4. In response to an illicit message, the access control logic monitors the frame with an identifier field of a higher priority than allowed. The source node broadcasts normally, but the acknowledgement delimiter is changed and the message is invalidated before it reaches the bus. Once the source node detects the altered bit, the transmit error counter increases by eight, an error flag is raised, and the source node attempts to retransmit the message. To prevent the rebroadcast from being invalidated again and entering into a loop until the source node enters a bus-off state, the source node must not detect another error, but to prevent the other nodes from taking action on subsequent messages, the broadcast across the bus cannot be the original illicit message. To accommodate these requirements and to maintain compliance with existing systems, the access control logic introduces message disparity during the retransmission. When an illicit message is retransmitted after having been invalidated by the access control logic, the message is echoed to the source node while a null message is broadcasted to the other nodes on

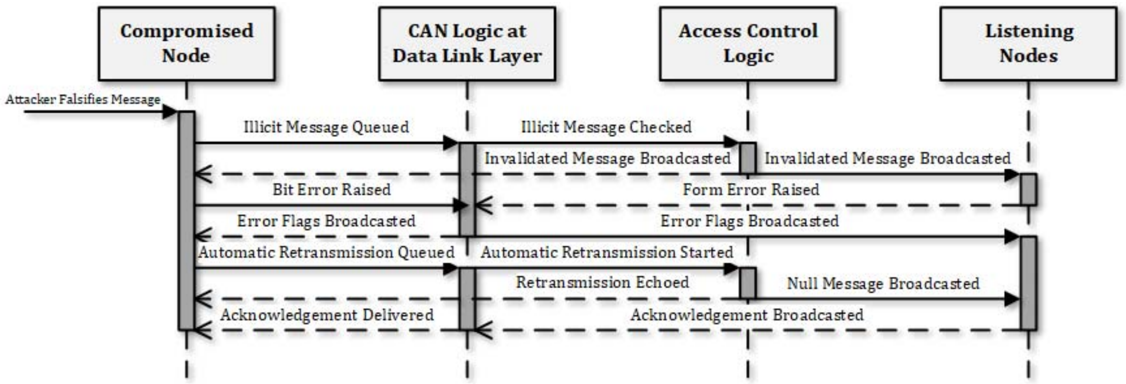

Compromised Node
CAN Logic at Data Link Layer
Access Control Logic
Listening Nodes
Attacker Falsifies Message
Illicit Message Queued
Illicit Message Checked
Invalidated Message Broadcasted
Invalidated Message Broadcasted
Bit Error Raised
Form Error Raised
Error Flags Broadcasted
Error Flags Broadcasted
Automatic Retransmission Queued
Automatic Retransmission Started
Retransmission Echoed
Null Message Broadcasted
Acknowledgement Delivered
Acknowledgement Broadcasted

Fig. 4: Attack Sequence Against CAN with Access Control

the bus. The source node detects no discrepancies or errors in the echoed message, decreasing the transmit error counter by one, and the other nodes take no action on the bus, allowing normal operations to resume.

Access controller functions, as listed in Table I, depend on the bus activity, as diagramed in Figure 5. During standard bus operations, when no messages beyond authorization have been broadcasted, the access controller serves as a repeater and merely forwards broadcasts bidirectionally. Once an illicit message has been detected, the access controller invalidates the message, prompting errors and ensuring that no bus action will be taken. When the source node attempts to resend the illicit message, the access controller broadcasts a null message to all other nodes while echoing the unmodified illicit message to the source node. Additional error handling is not necessary on the part of the access controller because errors from the base CAN standard are triggered. Fault confinement allows CAN nodes to quarantine and mute the effect of a compromised node. Once a node has entered a bus-off state, while it can no longer broadcast on the bus under a reset command is issued by the supervisor, it can still listen and function normally unless a module manufacturer disabled this capacity in a specific implementation.

TABLE I: Access Controller Functions

Operation	Bus State	Function
Repeater	Normal Operations	Bidirectional forwarding of broadcasts over bus
Access Controller	Message Exceeds Authorization	Invalidation of detected illicit message
Echoer	Message Retransmission	Echoes original illicit message to source node
Null Messenger	Message Retransmission	Broadcast of null message to other nodes

Any messages constructed or replayed by the attacker for the purpose of affecting the function or safety of the vehicle are invalidated by the access controller at non-safety critical nodes. Each message prompts the automatic retransmission by the source upon detection of the error during which other nodes detect a null message broadcast and take no action. Following the retransmission, the transmit error counter of the compromised node will have gone up by eight and down by one for a net increase of seven. Normal bus operation then resumes. In a denial of service attack, after a maximum of 37 consecutive messages, the transmit error counter of the compromised node will exceed a value of 256, triggering the standard's fault confinement property to prompt the device to switch to a bus-off state, terminating the attack.

B. *Real-Time Considerations*

Incorporating the logic for the access controller with the existing standard does not impact the timing of bus activity. The controller uses conditional logic to prepare an action later in time at which point it pushes a known value, a dominant bit, as the acknowledgement delimiter. No time is lost during normal operations, and in the event of an illicit message, the bus is delayed for the duration it takes to transmit an error flag and the null message.

Whenever a node broadcasts an illicit message, the only effect to bus operations is the delay time equal to the size of the null frame due to automatic retransmission. If the transmission is anomalous, this delay will be infrequent and isolated. If the transmission is the result of a concerted effort to subvert system operations, the access controller will force the compromised node to accumulate broadcast errors until the attack ceases or until the compromised node reaches a bus-off state as a result of the standard's fault confinement strategy.

By relying on the self-policing strategy of the CAN standard, the access control logic does not require additional capabilities with respect to quarantining a node determined to be a security risk. Though a node in a bus-off state cannot initiate broadcasts, it can listen on the bus and can function as normal. It is only hindered in its ability to influence the bus. Furthermore, no additional vulnerability has been introduced. Without the access control logic, an attacker can trivially force a compromised node into a bus-off state by repeatedly sending invalid messages across the bus.

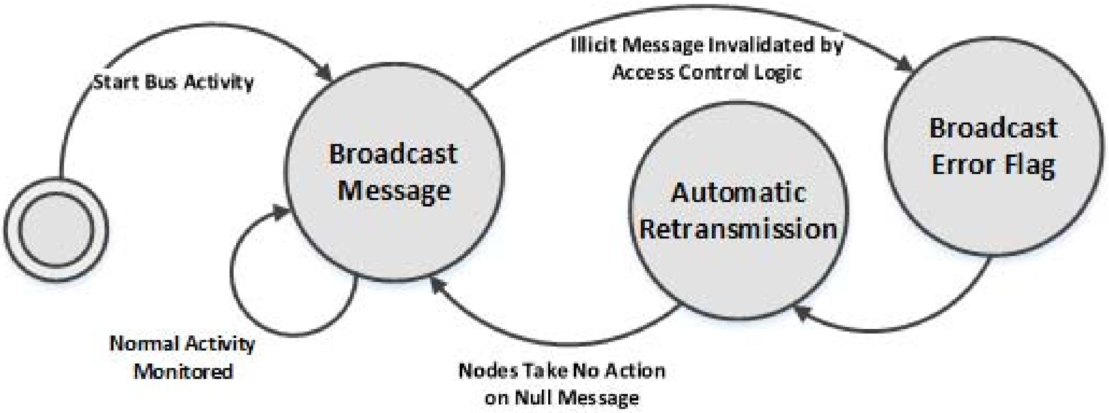

Start Bus Activity
Broadcast Message
Illicit Message Invalidated by Access Control Logic
Automatic Retransmission
Broadcast Error Flag
Normal Activity Monitored
Nodes Take No Action on Null Message

Fig. 5: CAN Bus Activity During Attack with Access Control

C. Resilience Capacity

As the focus of access control is to handle authorization, the goal is to prevent the compromising of low priority nodes to influence bus operations using high priority messages. By assigning a clearance value to each node, compromising a remotely accessible node should have limited accessibility to device itself, or from affecting lower priority nodes. Thus, replay attacks are not a concern. Without access control, an adversary can intercept a valid message, and replay it to replicate the effect. Adding access control, however, limits the effect of a replay attack to only those messages of equal or lower priority of the compromised node. Because nodes with wireless receivers do not produce high priority commands in modern vehicles, the access controller effectively eliminates the ability of a replay attack to affect safety-critical operations.

To illustrate the application of the proposed system, suppose an automobile in which the infotainment center is only authorized to transmit the lowest priority messages. Suppose an adversary gains access to the infotainment center of an automobile. If the attacker uses the node to send a command as if it were the node for the steering column, the message transmitted from the infotainment center is of a higher priority than allowed by the preconfigured access controller. The access controller alters the acknowledgement delimiter, prompting both a bit error and a form error. The infotainment center automatically retransmits the message sent by the attacker, but the access controller broadcasts a null message across the bus while echoing the original illicit message back to the infotainment center. The bus would then proceed as normal with the transmit error counter for the infotainment center having increased by a net of seven.

In addition to preventing command and control attacks that target safety-critical operations, the access control logic defeats denial of service attacks against safety-critical components. If a node is compromised in a system employing access control, that node cannot persistently send high priority messages because the node will be relegated to a bus-off state after 37 consecutive messages. Persistently sending lower priority messages will not affect valid high priority frames due to arbitration. Therefore, in order to perform a denial of service attack in spite of the access controller, a node allowed to send high priority messages would need to be directly compromised, and low priority devices would no longer be appropriate points of entry.

In a system lacking access control, a denial of service attack is effective if an adversary sends legitimately constructed messages of greater priority than competing messages. If low priority messages are broadcasted, higher priority tasks are not impeded. Therefore, attackers merely send the highest priority frame to completely disrupt communications. Access control shifts the focus from the priority of the message to the priority of the node. The adversary becomes limited by the point of entry. As long as the compromised node is not capable of sending high priority messages, the node cannot effectively be used to fully stop vehicular operations.

TABLE II: Impact of an Attack on Bus Time

Msg. Size	Attack Type	Bus Speed			
		10kbps	33 kbps	125 kbps	250 kbps
Min	Single Message	0.0044s	0.0013s	0.0003s	0.0002s
Max	Single Message	0.0108s	0.0033s	0.0009s	0.0004s
Min	Denial of Service	0.4736s	0.1435s	0.0379s	0.0189s
Max	Denial of Service	0.9427s	0.2870s	0.0758s	0.0379s

As can be seen in Table II, depending on the bus speed, implementation of the access control logic reduces the duration of otherwise successful safety-critical denial of service attacks of indeterminate length to less than a second of downtime. It should be noted that, because the invalidated message raises both bit and form errors, the source and listening nodes raise errors. As such, even after the source node enters the error-passive state, the denial of service attack will be limited to the same degree as when the node had been in the error-active state. If an implementation supports abort requests, the time added from broadcasting the null message can be subtracted from bus downtime, nearly halving the bus downtime.

VI. Design Decisions

A. Attribute Selection

1) Selection of the Data Link Layer: The logic for access control results from contemplation of a need for CAN standard compliance. By handling the logic below the application layer, the access control logic cannot be manipulated or circumvented by maliciously constructed packets, and relative vulnerability of a node is not vicariously transferred.

2) Selection of Acknolwedgement Delimiter: Altering the acknowledgement delimiter raises two types of errors which helps guarantee that an error will be raised. Raising errors by the source and listening nodes provides assurance that the message will be invalidated. Secondly, because a dominant bit in the acknowledgement delimiter slot is explicitly disallowed by the CAN standard, a message can be guaranteed to be invalidated at that point. Forcing the value for other bits may be an inadequate approach because either there is tolerance for either value despite what is expected or because the illicit message intended for that bit to be the forced value. Another approach would be to negate the first bit in the control field of an illicit message, thus triggering a bit error from the source node. However, it is possible that the first bit in the control field was incorrectly transmitted in the original message and that the negation actual repairs the illicit broadcast. In such an unlikely event, no bit error is detected, no error flag is raised, and the illicit message successfully impacts bus operations.

3) Zero Tolerance of Authorization Breaches: The logic of the access controller as proposed has no tolerance for illicit messages. While innocuous bit errors in the identifier would interrupt the original message with an error flag, some CAN modules could have a limited ability to send high priority messages, and if so, a counter can be configured for the tolerance level afforded a given node by the access controller.

B. The Null Message

1) Use of a Null Message: The access controller requires the ability to echo messages back to the transmitter to prevent a loop of automatic retransmissions. The CAN standard allows for an optional abort request capability from the LLC sublayer. If an implementation allows abort requests, instead of prompting an echo and a null message broadcast, the access control logic can instead terminate the retransmission following the invalidation of an illicit message. However, because the capability is optional in the CAN standard, the abort request method is not suggested due to anticipated problems when retrofitting existing systems.

2) Max Identifier of a Null Message: When the source node is retransmitting an illicit message, the access controller node broadcasts a null message of equal length using all zeroes in the identifier. If the identifier of the null operation broadcast was preset to be anything else, the retransmitted broadcast would potentially lose arbitration. If arbitration is lost at that point, the decision to inform the original source node that it lost arbitration would occur one bit time later due to the need for a comparison at actual bit time. If the remainder of the identifier of the illicit message are comprised of dominant bits, the source node could not be interrupted and would continue transmitting and could not listen to the actual broadcast. Potential synchronization issues aside, if the broadcasted message was a request for data from a compromised node, system operations would be directly impacted when the node does not receive the request. As such, the null operation frame needs to have the highest priority at the cost of the time it takes to transmit the message.

VII. Conclusion and Future Work

Networked components in automobiles and other vehicles which use CAN are vulnerable to attacks from outside actors. Because the communication standard does not provide the means to validate the message's source, an attacker having gained illicit access to a node can falsify, replay, or construct messages to impair the system and negatively affect operations and safety. Strict timing requirements limits the standard's ability to be substantially expanded with robust authentication measures, and application level precautions are costly to implement because they require adoption by all components and manufacturers.

We consider the scenario in which an adversary has succeeded in compromising a single node on a CAN network. We propose an authorization scheme that limits malicious activity after an attacker has compromised a vehicle component. If the compromised device is not safety-critical, which is likely if remotely accessible in a contemporary motor vehicle, bus access can be limited without affecting nominal bit time. By monitoring a message broadcasted across the bus, the identifier field can be used to determine if the source node has permission to send a given message. If the message is deemed illicit by the access control logic, the message is invalidated, and recipient nodes take no action, and in the context of a denial of service attack, the error counter will accumulate until the compromised node switches to bus-off pursuant to the CAN standard's fault confinement property.

The proposed approach has several advantages. The access control logic is fully compliant with the base CAN standard and does not require modification. Additionally, bus activity is only interrupted during the time of an attack and experiences no delays during normal operations. Implementations that allow abort requests minimize delays even during an attack. Incorporation of the proposed access control logic also limits denial of service attacks to be under one second. Though a node used for a concerted attack will relegate itself to a bus-off state, it will still be able to listen to the bus and to function normally, and because it was non-essential to automobile operations, there are no side effects to vehicle safety.

Future research in this area includes formally modeling the base CAN standard and the proposed extension to prove that the additional logic does not negatively impact nominal bit time during normal operations. The model will also demonstrate that the logic properly limits an adversary from issuing messages above the authorization of the compromised node. After establishing the proposal holds mathematically, the base CAN standard and the modified extended version with access control will be incorporated in a real-time network emulation model. The network emulator will be used to monitor traffic in real-time and to make diagnostic comparisons for evaluating proposed solutions.

References

[1] S. Corrigan, "Introduction to the controller area network (can)," Texas Instruments, Tech. Rep. Application Report SLOA101A, 2008.

[2] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive can networks — practical examples and selected short-term countermeasures," in *Proc. of the 27th international conference on Computer Safety, Reliability, and Security (SAFECOMP'08)*, 2008, pp. 235–248. [Online]. Available: https://dx.doi.org/10.1007/978-3-540-87698-4_21

[3] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *IEEE Symposium on Intelligent Vehicles (IV '11)*, 2011, pp. 528–533. [Online]. Available: https://dx.doi.org/10.1109/IVS.2011.5940525

[4] C.-W. Lin and A. Sangiovanni-Vincentelli, "Cyber-security for the controller area network (can) communication protocol," in *Proc. of the 2012 International Conference on Cyber Security (CYBERSECURITY '12)*, 2012, pp. 1–7. [Online]. Available: https://dx.doi.org/10.1109/CyberSecurity.2012.7

[5] H. Ueda, R. Kurachi, H. Takada, T. Mizutani, M. Inoue, and S. Horihata, "Security authentication system for in-vehicle network," *SEI Technical Review*, vol. 81, pp. 5–9, 2015.

[6] B. Groza and S. Murvay, "Efficient protocols for secure broadcast in controller area networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2034–2042, 2013. [Online]. Available: https://dx.doi.org/10.1109/TII.2013.2239301

[7] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. of the 20th USENIX conference on Security (SEC'11)*, 2011.

[8] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *Proc. of the 2010 IEEE Symposium on Security and Privacy (SP '10)*, 2010, pp. 447–462. [Online]. Available: https://dx.doi.org/10.1109/SP.2010.34

[9] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in *Proc. of the 2008 IEEE Symposium on Intelligent Vehicles (IV '08)*, 2008, pp. 220–225. [Online]. Available: https://dx.doi.org/10.1109/IVS.2008.4621263

[10] I. Foster and K. Koscher, "Exploring controller area networks," *;login:*.

[11] P. Carsten, T. R. Andel, M. Yampolskiy, and J. T. McDonald, "In-vehicle networks: Attacks, vulnerabilities, and proposed solutions," 2015. [Online]. Available: https://doi.org/10.1145/2746266.2746267

[12] P. Carsten, T. R. Andel, M. Yampolskiy, J. T. McDonald, and S. Russ, "A system to recognize intruders in controller area network (can)," in *Proc. of the 3rd International Symposium for ICS & SCADA Cyber Security Research (ICS-CSR '15)*, 2015, pp. 111–114. [Online]. Available: https://doi.org/10.14236/ewic/ICS2015.15

[13] C.-W. Lin, Q. Zhu, C. Phung, and A. Sangiovanni-Vincentelli, "Security-aware mapping for can-based real-time distributed automotive systems," in *Proc. of the international conference on Computer-Aided Design (ICCAD '13)*, 2013.

[14] C. Patsakis, K. Dellios, and M. Bouroche, "Towards a distributed secure in-vehicle communication architecture for modern vehicles," *Computer Security*, vol. 40, pp. 60–74, feb 2014. [Online]. Available: https://doi.org/10.1016/j.cose.2013.11.003