USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

WIP: Intrusion Detection and Localization for CAN by Extracting Propagation Delay Features from Message Intervals

Zhaozhou Tang, *Georgia Institute of Technology;* Khaled Serag, *Qatar Computing Research Institute;* Saman Zonouz, *Georgia Institute of Technology;* Z. Berkay Celik and Dongyan Xu, *Purdue University;* Raheem Beyah, *Georgia Institute of Technology*

https://www.usenix.org/conference/vehiclesec25/presentation/tang

This paper is included in the Proceedings of the 3rd USENIX Symposium on Vehicle Security and Privacy.

August 11–12, 2025 • Seattle, WA, USA

978-1-939133-49-6

Open access to the Proceedings of the 3rd USENIX Symposium on Vehicle Security and Privacy is sponsored by USENIX.

WIP: Intrusion Detection and Localization for CAN by Extracting Propagation Delay Features from Message Intervals

Zhaozhou Tang
Georgia Institute of Technology

Khaled Serag
Qatar Computing Research Institute

Saman Zonouz
Georgia Institute of Technology

Z. Berkay Celik
Purdue University

Dongyan Xu
Purdue University

Raheem Beyah
Georgia Institute of Technology

Abstract

The Controller Area Network (CAN bus) is a critical communication protocol used in vehicles. Its lack of built-in security allows an attacker with bus access to launch various impersonation attacks, such as spoofing and replay. Researchers proposed defense approaches to counter these attacks using various features, such as message frequencies, voltages, signal asymmetries, and more recently, timing. In this paper, we propose a new timing feature which we call "transmit signatures" (TS). TS strongly depends on the physical distances and propagation delays between ECUs, allowing us to detect and localize impersonation attacks. Unlike prior approaches, we extract TS from the natural time intervals between messages, without installing additional wiring or modifying ECUs' software and traffic. We formulate a hypothesis about TS' distance dependency. We then conduct experiments to validate and refine our hypothesis. Using the refined theory, we introduce and evaluate a TS modeling approach and propose attack detection and localization methods.

1 Introduction

The Controller Area Network (CAN) is the most widely used in-vehicle network protocol today. Introduced in the 1980s, it defines the communication rules of a broadcast-based bus, where various electronic control units (ECUs) exchange messages to control different aspects of a vehicle's operation. Researchers demonstrated that CAN is vulnerable to a plethora of impersonation attacks, where an attacker with bus access injects, forges, or replays messages of another ECU, potentially causing severe safety consequences [15, 18, 21, 22]. Impersonation could be enhanced by disabling the impersonated ECU using other vulnerabilities of CAN [1, 3, 31, 32].

Researchers have proposed many defense approaches against such attacks. Initially, they attempted to adopt cryptographic measures such as MACs to provide message authentication [9, 23, 24], but were hindered by challenges such as key management, high computational overhead, and stringent real-time requirements of ECUs. Consequently, intrusion detection systems (IDSs) gained more traction. These IDSs use a trusted node to detect traffic anomalies using diverse features, such as message frequencies [26, 35], clock skews [4], signal asymmetries [29, 33], and voltage [5, 8, 13, 14].

The relatively less explored timing-based IDS approaches have shown great promise recently. Some approaches measure signals' time difference of arrival (TDoA) which depends on ECUs' physical locations [20, 25, 28]. They offer accurate single-message attack detection and localization with very low false positive rates over prolonged operations. However, they require excessive additional wiring, limiting their practical adoptions. Others add secret delays between messages and inspect them to detect anomalies [10, 11, 30]. Although they offer equally good detection performance without requiring additional wiring, they forgo the attacker localization capability and incur other implementation overhead including modifying ECUs' software and rescheduling their traffic.

We aim to combine the benefits of existing timing-based IDSs and resolve their common limitations. Based on the observation that location-dependent features like TDoA support robust detection and localization of impersonation attacks, we propose a new timing feature of this type called "transmit signatures" (TS). TS is extracted from the natural time intervals between consecutive messages. It strongly depends on the physical distances and propagation delays between ECUs and could offer comparable attack detection and localization performance. Meanwhile, since TS is extracted from natural message intervals, it requires no additional wiring, software modifications, or traffic rescheduling, eliminating all implementation overhead of existing timing-based IDSs.

In this WIP paper, we lay the groundwork for building an IDS using TS. We first introduce TS and its extraction, and formulate a hypothesis about its distance dependency (Sec. 5). We then validate the hypothesis experimentally and refine it based on new findings (Sec.6). Using the refined theory, we introduce and evaluate a TS modeling approach, and propose how to use TS to detect and localize attacks (Sec. 7). Finally, we outline future research directions (Sec. 8).

Table 1: Comparing TS with other approaches.

Approach	Feature		Cost				
	Detection	Localization	Modify Software	Excessive Hardware	Reschedule Messages	Disrupt Traffic	Processing Overhead
Secret Delay [10, 11, 30]	✓	X	✓	X	✓	X	○
TDoA [20, 25, 28]	✓	✓	X	✓	X	X	○
CANARY [12]	✓	✓	X	✓	X	✓	○
CAN-LOC [16]	✓	✓	X	X	X	X	●
TS	✓	✓	X	X	X	X	○

In this paper, we make the following contributions:

- We propose a new timing feature (transmit signatures) extracted from the natural time interval between consecutive messages to detect and localize impersonation attacks.
- We formulate a hypothesis about TS' distance dependency. We validate and refine it with testbed experiments.
- Using the refined TS theory, we propose and evaluate a TS modeling approach and achieve good accuracy.

2 Background

CAN Basics. CAN uses differential voltages between two wires (CANH and CANL) to encode bits. A positive voltage represents a 0 (dominant bit) and a zero voltage represents a 1 (recessive bit). Nodes use a CAN transceiver to convert between digital signals and differential voltages.

Message Format. The bus state is recessive when it is idle. A transmitter starts a message by sending a dominant start of frame (SOF) bit. An identifier (ID) follows the SOF, which is uniquely assigned to nodes. At the end of a message, receivers send a dominant acknowledgment bit (ACK) to indicate they have received the message. Two consecutive messages are separated by at least three Inter-Frame Space (IFS) bits.

Bit Timing. Nodes agree on a common bit rate for communication. Bit durations are defined by the number of a minimum timing unit called the time quanta (t_q). A bit contains four segments: synchronization (Sync_Seg), propagation time (Prop_Seg), phase buffer 1 (Phase_Seg1), and 2 (Phase_Seg2) segments. The Sync_Seg is 1 t_q long. The other segments have configurable lengths based on the number of t_q.

CAN Bus Transmission Delays. Signals from a transmitter take some time to propagate to a receiver. This delay mainly arises from two sources. First, CAN transceivers incur delays when converting between digital signals and differential voltages. Second, the signals take time to propagate through wires. This propagation delay increases with distances between nodes, and also depends on wire types and impedance of electrical components connected along the wires.

Hard Synchronization. CAN nodes may have different clock frequencies (clock skews). Their actual bit durations may differ and they may accumulate a phase shift during the bus idle period. When a transmitter's SOF edge arrives, it may not align with the start of the receiver's bit time, due to this phase shift and transmisison delays. The hard synchronization accounts for this. When receiving the recessive to dominant SOF edge, nodes restart their bit time with Sync_Seg completed.

Impersonation Attacks. CAN does not offer built-in authentication. Therefore, an attacker can impersonate as another ECU by sending messages using another its ID to control the vehicle functions it is in charge of.

3 Related Work

IDS Approaches. Some IDSs inspect traffic features such as message frequencies [26, 35] and clock skews [4] to detect anomalies. Others use features dependent on ECUs' hardware characteristics, such as voltage [5, 8, 13, 14] and signal asymmetries [29, 33]. However, some IDSs are evadable [2, 27]. Many others, despite good detection performance, only detect attacks that inject flows of messages or incur non-negligible false positives over prolonged operations.

Timing-Based Approaches. Some approaches measure signals' time difference of arrival (TDoA) at two ends of the bus, which depends on propagation delays and ECUs' physical locations. TIDAL-CAN demonstrates TDoA's high accuracy in identifying message senders and detecting attacks [20]. It can also infer the attacker's physical locations, if the attacker has compromised in-vehicle ECUs or connected additional ECUs to the bus. EdgeTDC shows TDoA's strong security against various types of attacker manipulations [25]. SPARTA further shows TDoA's lightweight by implementing it on resource-constrained hardware [28]. However, measuring TDoA requires attaching wires from both ends of a CAN bus to a measuring unit in the middle. This doubles wire lengths and limits the implementation feasibility of TDoA-based approaches.

Others add secret delays between messages for authentication without requiring extra wiring. INCANTA and CANTO add delays to periodic messages' arrival times but do not support aperiodic messages [10, 11]. ZBCAN instead adds discretized delays in all messages' IFS [30], which is calculated

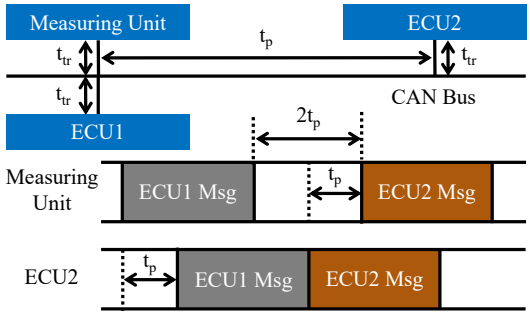

Figure 1: Illustrating the origin of TS.

deterministically using a secret key and a lightweight pseudo random function. Thus, ZBCAN provides accurate single-message detection, zero false positive rates, and low computational overhead. Nevertheless, secret delay approaches have two main limitations. First, all ECUs' software must be modified, posing significant implementation overhead. Adding delays also modifies traffic patterns and may increase bus load or cause priority inversions. Second, without using location-dependent features, they cannot localize the attacker like TDoA, which is crucial for post-incident management such as finding and replacing a compromised ECU.

Attacker Localization Approaches. Other attacker localization approaches exist besides TDoA. CANARY uses relays to partition the bus [12]. It disconnects parts of the bus and analyzes the traffic in each part to narrow down the attacker's location. However, it is expensive to implement and disrupts communication until the attacker is located. CAN-LOC collects voltage signals and trains a neural network to locate the attacker [16]. Despite its high accuracy, CAN-LOC requires a high sampling rate and is very computationally expensive.

Comparison with Our Approach. Table 1 compares qualitatively timing-based IDSs and attacker localization approaches with our proposed approach (TS). IDSs using secret delays only provide detection and not localization. They all require software modifications and message rescheduling. IDSs using TDoA offer both features, but require excessive hardware changes by doubling wire lengths. CANARY requires hardware changes by installing relays, and disrupts communication while performing attacker localization. CAN-LOC incurs the most computational overhead. TS can provide both detection and localization. It eliminates all implementation overhead of other approaches and is computationally efficient.

4 Threat Model

In line with the scope of other timing-based approaches [25, 30], we focus on detecting and localizing impersonation attacks. Attacks such as transmitting authorized messages with fake content are not considered. Other works have addressed such attacks and they can be easily integrated with our approach to provide protection [17, 34]. We assume the attack can originate from any source, including an in-vehicle ECU compromised by an attacker remotely, or an additional ECU connected by an attacker with physical access. The attacker can inject messages using any techniques, such as using the ECU's CAN controller or bypassing it to use another ECU peripheral, similar to prior works [6, 33].

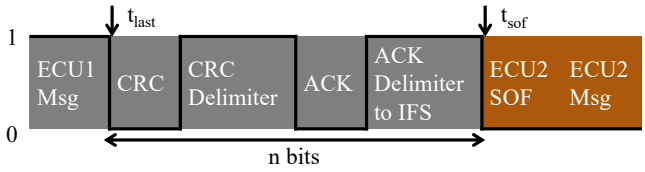

Figure 2: Quantities used for calculating TS.

5 Transmit Signature Hypothesis

In this section, we introduce TS. We formulate a hypothesis about its distance dependency and explain how to extract it.

Distance Dependency. Consider an example CAN bus layout in Fig. 1. A measuring unit and ECU1 are located in the same position. ECU2 is located some distance away from them. We assume there is a propagation delay of t_p between ECU1 and ECU2 and no propagation delay between ECU1 and the measuring unit. For simplicity, we assume all transceiver delays equal to t_{tr} and investigate impacts of unequal transceiver delays in the future (Sec. 8). If ECU1 sends a message at time t_0, the measuring unit receives it at $t_0 + 2t_{tr}$ and ECU2 receives it at $t_0 + t_p + 2t_{tr}$. If the duration of ECU1's message (including three IFS bits) is t_{msg}, the measuring unit and ECU2 each see the message end at $t_0 + t_{msg} + 2t_{tr}$ and $t_0 + t_p + t_{msg} + 2t_{tr}$. If ECU2 sends a message t_r afterwards, it arrives at the measuring unit at $t_0 + t_p + t_{msg} + t_p + t_r + 4t_{tr}$ due to propagation delays. The measuring unit sees an additional delay of $2t_p + 2t_{tr}$ between the end of ECU1's message and the start of ECU2's message. We define this additional delay as ECU2's TS. As this example shows, TS depends on the propagation delay between senders of the current and previous message and their relative distances.

Extracting TS. To measure TS, we place a measuring unit at one end of a CAN bus and measure the timestamps of all $1 \rightarrow 0$ edges in messages. Fig. 2 shows ECU1 and ECU2's messages and the quantities used to calculate ECU2's TS:

- t_{last}: Timestamp of the last $1 \rightarrow 0$ edge before the ACK bit in ECU1's message
- t_{sof}: Timestamp of the SOF bit in ECU2's message
- n: The number of bits between t_{last} and t_{sof}
- T: ECU2's actual bit duration

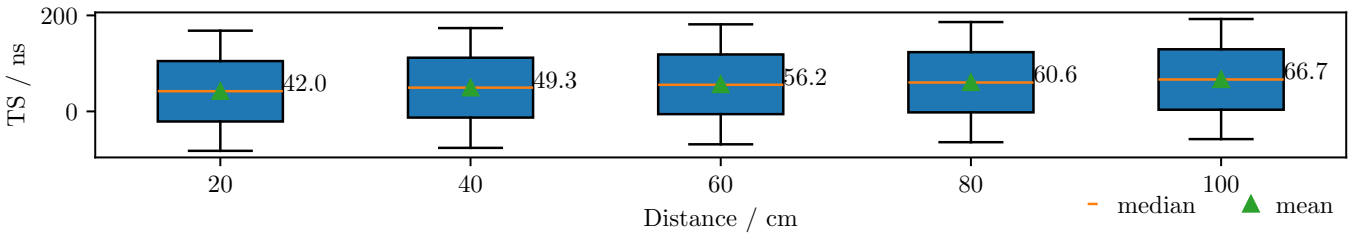

Figure 3: Boxplot of ECU2's TS when placed at different distances from ECU1.

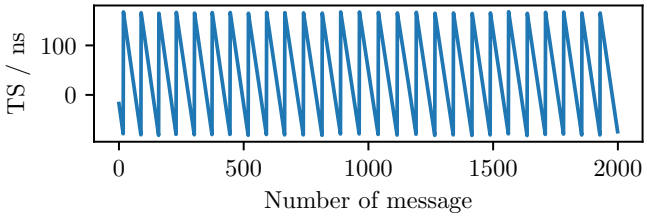

Figure 4: ECU2's TS for each successive message.

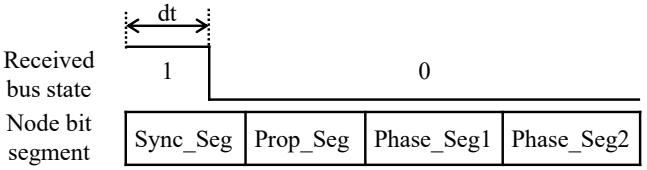

Figure 5: Hard synchronization.

t_{last} is when the measuring unit sees the end of ECU1's message. We use the last edge before ACK since the ACK bit is not transmitted by ECU1 but by ECU2. t_{sof} is when the measuring unit sees the start of ECU2's message. If ECU2's message is sent some time after ECU1, the expected delay between t_{last} and t_{sof} had there been no propagation delays can be represented using ECU2's bit duration ($n \times T$). T is measured using ECU2's messages to account for deviations between ECU2's actual and ideal bit duration. Following prior works [29, 33], we calculate it using the time $t \downarrow\downarrow$ between two consecutive $1 \rightarrow 0$ edges with $n \downarrow\downarrow$ bits in between:

$$T = \frac{t_{\downarrow\downarrow}}{n_{\downarrow\downarrow}} \tag{1}$$

We subtract the expected delay from the actual to obtain ECU2's TS, which is a fraction of one bit duration:

$$TS = t_{sof} - t_{last} - nT \tag{2}$$

6 Experimental Validation

Experiment Setup. To validate the TS hypothesis in Sec. 5, we set up a testbed as Fig. 1. All ECUs use the Arduino Due board and its on-chip CAN controller, and TJA1051 CAN transceiver. We let ECU1 transmit fixed-period messages and ECU2 transmit a message immediately after it receives a message from ECU1. We measure message edge timestamps using a time-to-digital converter and calculate ECU2s' TS using Equation 2. We vary the distance between ECU1 and ECU2 and repeat the measurements.

Experiment Findings. Fig. 3 shows a boxplot of ECU2's TS for five different distances: 20cm, 40cm, 60cm, 80cm, and 100cm, with the means labeled. The distributions are distance-dependent. The mean, minimum, and maximum TS increases with distance, although they do not increase linearly, possibly due to non-uniform wire propagation delays. We perform a one-way Analysis of Variance (ANOVA) to test if the distributions for any two distances are the same. We obtain a p-value of 1.77e−29, supporting the distance dependency of TS. However, the distributions have significant overlaps.

When ECU2's TS are plotted over time for each successive message, they show a common pattern. Fig. 4 shows an example when the distance is 20cm. ECU2's TS oscillates periodically. When it reaches its maximum, it decreases linearly until reaching a minimum value and then increases to the maximum in the next message. Furthermore, the difference between the maximum and minimum TS is 1 t_q (Sec. 2).

Refined TS Theories. These results are explained by CAN's hard synchronization mechanism (Sec. 2). The CAN standard states that when nodes perform hard synchronization "the bit time shall be restarted by each bit-timing logic unit with Sync_Seg completed" [7]. Fig. 5 shows the exact behavior that we deduce based on our experiment and confirm in open-source CAN controller designs [19]. When a node receives a $1 \rightarrow 0$ SOF edge, it does not start Prop_Seg right after the edge, but instead waits until the current t_q has elapsed, and then starts Prop_Seg. The time dt from the start of the t_q before Prop_Seg to the edge location depends on the accumulated phase shift between the transmitter and receiver and changes over time. This explains the observations in Fig. 4.

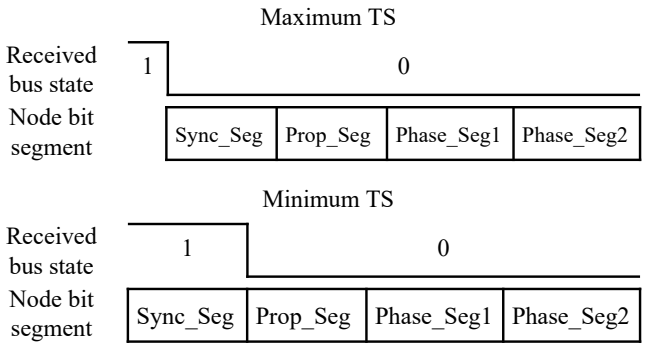

Figure 6: When maximum and minimum TS occurs.

(1) Maximum and Minimum: Fig. 6 shows when the maximum and minimum TS occurs. We assume at t_0, ECU1 sends a message containing n_{msg} bits (including three IFS bits) each with duration T_1. The maximum TS is observed if ECU1's SOF edge arrives at ECU2 right after the start of a t_q ($dt \approx 0$). ECU2 waits until the end of the t_q and starts Prop_Seg. From ECU2's perspective, the first bit in ECU1's message starts at $t_0+t_p+2t_{tr}$ and the message finishes at $t_0+t_p+2t_{tr}+n_{msg}T_1$. Then ECU2 sends its message and the measuring unit receives at $t_0+2t_p+4t_{tr}+n_{msg}T_1$, observing a TS of $2t_p+2t_{tr}$.

Contrastingly, the minimum TS occurs if ECU1's SOF edge arrives close to the end of a t_q ($dt \approx t_q$) and ECU2 immediately starts Prop_Seg. From ECU2's perspective, the first bit in ECU1's message starts at $t_0+t_p+2t_{tr}-t_q$ and the message finishes at $t_0+t_p+2t_{tr}+n_{msg}T_1-t_q$. Then ECU2 sends its message and the measuring unit receives it at $t_0+2t_p+4t_{tr}+n_{msg}T_1-t_q$, observing a TS of $2t_p+2t_{tr}-t_q$. Its difference from the maximum TS is 1 t_q.

(2) Periodic Oscillation: This is caused by changes in the phase shift between ECU1 and ECU2 over time. Suppose their actual bit durations are T and $T-\Delta T$, respectively. After every n bit time, dt increases by:

$$nT - n(T-\Delta T) = n\Delta T \tag{3}$$

which is linear over time. Relating this to Fig. 6, initially dt is small. For each successive message, dt increases and ECU2's TS decreases as it becomes closer to the minimum TS condition. At the minimum condition when dt increases further in the next message, ECU1's SOF edge is at the start of a t_q. The maximum TS condition is satisfied causing a large TS. This process repeats throughout the ECUs' operation.

7 Applying TS Theories

We apply the refined theories in Sec. 6 to propose and evaluate a TS modeling method. We then propose how to use TS to detect and localize attacks and conduct a security analysis.

Algorithm 1 Find expected TS based on the refined theories

Input: The expected minimum TS (min)
the expected maximum TS (max)
the slope of change in TS (a)
Output: The expected TS

```
1: procedure EXPECTEDTS(min, max, a)
2:     TS = TS_last − a
3:     if TS < min then
4:         TS = max − min + TS
5:     end if
6:     return TS
7: end procedure
```

Algorithm 2 Difference between expected and actual TS

Input: Measured TS (TS),
predicted TS (TS_{pred})
length of 1 t_q (t_q)
Output: Difference between measurement and prediction

```
1: procedure COMPUTEDIFF(TS, TS_pred,)
2:     diff = |TS − TS_pred|
3:     if diff > 0.5t_q then
4:         if TS < TS_pred then
5:             diff = ||TS − min| + |max − TS_pred||
6:         else
7:             diff = ||max − TS| + |TS_pred − min||
8:         end if
9:     end if
10:    return diff
11: end procedure
```

Modeling TS. Based on Fig. 4, we model ECU2's expected TS using a linear model with the following parameters:

- min: The expected minimum TS
- max: The expected maximum TS
- a: The slope of the linearly decreasing TS in a unit of nanoseconds per message

We learn them from some reference messages. We choose min and max as the minimum and maximum observed TS. To find a, we fit a linear regression model to messages in a time window where their TS decreases linearly. We compute the expected TS for a message with Algorithm 1. It first subtracts a from the last measured TS (line 2). If the result is smaller than min, it is adjusted to a value close to max (lines 3-5).

Attack Detection. For each new message we calculate its expected TS assuming it is from the authorized sender and compare it with the measured TS. We cannot simply compute their absolute difference. Small prediction inaccuracies when TS is close to min or max may cause Algorithm 1 to incorrectly adjust the prediction close to max when it should not, or vice versa, resulting in a large absolute difference. Instead,

Table 2: Mean squared errors (MSE) of TS modeling.

Distance / cm	20	40	60	80	100
MSE / ns^2	0.077	0.059	0.054	0.071	0.117

we use Algorithm 2 to additionally check for large absolute differences and compute the difference as the spread of the measured and predicted values around *min* and *max* (lines 3-9). We choose a threshold ε and detect an attack if:

$$computeDiff(expectedTS, TS) > \varepsilon \tag{4}$$

Attacker Localization. When an attack is detected, we first check if it is launched by a compromised ECU. For every ECU e we compute the difference between the expected and measured TS, assuming it is the attacker:

$$error_e = computeDiff(expectedTS, TS) \tag{5}$$

We then choose the ECU with the smallest difference:

$$e_{best} = \arg\min_e(error_e) \tag{6}$$

If $error_{e_{best}}$ is smaller than ε, we have identified a compromised ECU. Otherwise, the message's TS does not match any existing ECUs, and it must be from an additional ECU installed by the attacker. We cannot locate it using a single message since its TS may be influenced by the unknown phase shift of this ECU. Instead, we take the maximum TS observed in a sequence of malicious messages, which purely depends on distance. We can compare the attacker's maximum TS with all legitimate ECUs' maximum TS to find between which two ECUs the attacker is located. Using its neighbors' TS and distances, we may further use linear interpolation to give a more precise estimate of the attacker's location.

Security Analysis. To evade detection, an attacker must adjust message transmission timings to emulate the victim's TS. If the attacker injects messages using a CAN controller, he can only adjust message timings at the granularity of bit durations. His messages' TS do not change as Equation 2 already considers the number of bits between messages when calculating TS. Thus, the attacker cannot evade detection and similarly can be accurately localized, whether he compromises an ECU or installs an additional ECU. If the attacker bypasses the CAN controller, he can obtain finer control of message timings and change his TS. However, he must know the victim's expected TS by accurately measuring the propagation delays between ECUs down to nanosecond precision. This is not possible without specialized hardware and he cannot evade detection if he compromises or installs a standard ECU. However, his location may be estimated less accurately.

Evaluating TS Modeling. Using the testbed in Sec. 6, we model and predict ECU2's TS for five different distances with Algorithm 1. We compute the prediction errors using Algorithm 2 and show the mean squared error (MSE) in Table 2. Compared to the difference between TS at different distances (mean TS differs by 4.4ns at minimum in Fig. 3), the errors are small by orders of magnitude. This demonstrates the robustness of TS features and their good potential to offer accurate intrusion detection and localization.

8 Discussion and Future Research

Networks With More ECUs. We have validated the TS theories and modeling on a two-ECU testbed. Real networks contain more ECUs which we need to account for. Since TS depends on the distance between senders of the current and previous message, we will investigate building separate models for every ECU pair or including the sender of the previous message as an additional model parameter.

Longer Response Times. In our preliminary experiments, ECU2 responds immediately after ECU1's messages. We will test longer response times between messages.

Messages With Variable Periods. As shown in Equation 3, ECUs' phase shift depends on the time between messages. Consequently, the change in TS per message depends on the messages' periods. In our experiment, we use fixed-period messages. Therefore, the change in TS per message is fixed and we can fit a linear model to TS using the number of messages as the independent variable. To account for variable-period messages, we will incorporate the time elapsed since the last message in the TS model.

Unequal Transceiver Delays. In the current TS theories we assume all ECUs' transceiver delays are equal, but in reality they may have small differences. The increase in TS with distance as we observe in Sec. 6 suggests that differences in propagation delays may be the dominant component in TS. We will empirically measure transceiver delay differences across different transceiver models and compare with wire propagation delays to validate if this generally holds true.

Environmental Influences. We will investigate if TS is influenced by environmental conditions such as temperature, and account for them if it is. We will also perform a security analysis of our attack detection and localization approach against poisoning attacks similar to [2, 25].

9 Conclusion

We propose a new timing feature (transmit signatures) to secure the CAN bus. Extracted from natural message intervals, TS strongly depends on distances and propagation delays between ECUs. It can potentially offer robust detection and localization of impersonation attacks and incurs no implementation overhead such as ECU software or traffic modifications. We formulate TS theories and conduct experiments to validate and refine them. We then propose modeling, attack detection, and localization approaches using TS, and conduct initial experiments to demonstrate the accuracy of TS modeling.

Acknowledgments

We thank the anonymous reviewers and shepherd for their invaluable feedback. This work was supported in part by the National Science Foundation (NSF) under the Secure and Trustworthy Cyberspace (SaTC) program and Grant CNS-2144645, as well as the Office of Naval Research (ONR) under Grants N00014-22-1-2671 and N00014-18-1-2674. Any opinions, findings, and conclusions in this paper are those of the authors and do not necessarily reflect the views of our sponsors.

References

[1] Khaled Serag Alsharif. PROACTIVE VULNERABILITY IDENTIFICATION AND DEFENSE CONSTRUCTION – THE CASE FOR CAN. 2023.

[2] Rohit Bhatia, Vireshwar Kumar, Khaled Serag, Z Berkay Celik, Mathias Payer, and Dongyan Xu. Evading voltage-based intrusion detection on automotive can. In *Network and Distributed System Security Symposium (NDSS)*, 2021.

[3] Kyong-Tak Cho and Kang G. Shin. Error handling of in-vehicle networks makes them vulnerable. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016.

[4] Kyong-Tak Cho and Kang G. Shin. Fingerprinting electronic control units for vehicle intrusion detection. In *USENIX Security Symposium*, 2016.

[5] Kyong-Tak Cho and Kang G. Shin. Viden: Attacker identification on in-vehicle networks. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017.

[6] Alvise de Faveri Tron, Stefano Longari, Michele Carminati, Mario Polino, and Stefano Zanero. Canflict: Exploiting peripheral conflicts for data-link layer attacks on automotive networks. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.

[7] International Organization for Standardization (ISO). Road vehicles — Controller area network (CAN). Part 1: Data link layer and physical coding sublayer, ISO 11898-1:2024. 2024.

[8] Mahsa Foruhandeh, Yanmao Man, Ryan M. Gerdes, Ming Li, and Thidapat Chantem. SIMPLE: single-frame based physical layer identification for intrusion detection and prevention on in-vehicle networks. In *Annual Computer Security Applications Conference (ACSAC)*, 2019.

[9] Bogdan Groza, Stefan Murvay, Anthony Van Herrewege, and Ingrid Verbauwhede. Libra-can: a lightweight broadcast authentication protocol for controller area networks. In *International Conference on Cryptology and Network Security*, 2012.

[10] Bogdan Groza, Lucian Popa, and Pal-Stefan Murvay. INCANTA - intrusion detection in controller area networks with time-covert authentication. In *Security and Safety Interplay of Intelligent Software Systems*, 2018.

[11] Bogdan Groza, Lucian Popa, and Pal-Stefan Murvay. CANTO - covert authentication with timing channels over optimized traffic flows for CAN. *IEEE Transactions on Information Forensics and Security*, 2021.

[12] Bogdan Groza, Lucian Popa, Pal-Stefan Murvay, Yuval Elovici, and Asaf Shabtai. CANARY - a reactive defense mechanism for controller area networks based on active relays. In *USENIX Security Symposium*, 2021.

[13] Marcel Kneib and Christopher Huth. Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018.

[14] Marcel Kneib, Oleg Schell, and Christopher Huth. EASI: edge-based sender identification on resource-constrained platforms for automotive networks. In *Network and Distributed System Security Symposium (NDSS)*, 2020.

[15] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, et al. Experimental security analysis of a modern automobile. In *IEEE Symposium on Security and Privacy (S&P)*, 2010.

[16] Efrat Levy, Asaf Shabtai, Bogdan Groza, Pal-Stefan Murvay, and Yuval Elovici. Can-loc: Spoofing detection and physical intrusion localization on an in-vehicle can bus based on deep features of voltage signals. *IEEE Transactions on Information Forensics and Security*, 2023.

[17] Stefano Longari, Carlo Alberto Pozzoli, Alessandro Nichelini, Michele Carminati, and Stefano Zanero. Candito: improving payload-based detection of attacks on controller area networks. In *International Symposium on Cyber Security, Cryptology, and Machine Learning*, 2023.

[18] Charlie Miller and Chris Valasek. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015.

[19] Igor Mohor. Can protocol controller. https://opencores.org/projects/can, 2017.

[20] Pal-Stefan Murvay and Bogdan Groza. TIDAL-CAN: differential timing based intrusion detection and localization for controller area network. *IEEE Access*, 2020.

[21] Sen Nie, Ling Liu, and Yuefeng Du. Free-fall: Hacking tesla from wireless to can bus. *Black Hat USA*, 2017.

[22] Sen Nie, Ling Liu, Yuefeng Du, and Wenkai Zhang. Over-the-air: How we remotely compromised the gateway, bcm, and autopilot ecus of tesla cars. *Black Hat USA*, 2018.

[23] Mert D Pesé, Jay W Schauer, Junhui Li, and Kang G. Shin. S2-can: Sufficiently secure controller area network. In *Annual Computer Security Applications Conference (ACSAC)*, 2021.

[24] Andreea-Ina Radu and Flavio D Garcia. Leia: A lightweight authentication protocol for can. In *European Symposium on Research in Computer Security (ESORICS)*, 2016.

[25] Marc Roeschlin, Giovanni Camurati, Pascal Brunner, Mridula Singh, and Srdjan Capkun. Edgetdc: On the security of time difference of arrival measurements in CAN bus systems. In *Network and Distributed System Security Symposium (NDSS)*, 2023.

[26] Matthew Rogers, Phillip Weigand, Jassim Happa, and Kasper Rasmussen. Detecting CAN attacks on J1939 and NMEA 2000 networks. *IEEE Transactions on Dependable and Secure Computing*, 2023.

[27] Sang Uk Sagong, Xuhang Ying, Andrew Clark, Linda Bushnell, and Radha Poovendran. Cloaking the clock: Emulating clock skew in controller area networks. In *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2018.

[28] Oleg Schell and Marcel Kneib. SPARTA: signal propagation-based attack recognition and threat avoidance for automotive networks. In *ACM Asia Conference on Computer and Communications Security*, 2023.

[29] Oleg Schell, Claudio Oechsler, and Marcel Kneib. Asymmetric symbol and skew sender identification for automotive networks. *IEEE Transactions on Information Forensics and Security*, 2022.

[30] Khaled Serag, Rohit Bhatia, Akram Faqih, Muslum Ozgur Ozmen, Vireshwar Kumar, Z. Berkay Celik, and Dongyan Xu. ZBCAN: A zero-byte CAN defense system. In *USENIX Security Symposium*, 2023.

[31] Khaled Serag, Rohit Bhatia, Vireshwar Kumar, Z. Berkay Celik, and Dongyan Xu. Exposing new vulnerabilities of error handling mechanism in CAN. In *USENIX Security Symposium*, 2021.

[32] Khaled Serag, Vireshwar Kumar, Z Berkay Celik, Rohit Bhatia, Mathias Payer, and Dongyan Xu. Attacks on can error handling mechanism. In *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2022.

[33] Zhaozhou Tang, Khaled Serag, Saman Zonouz, Z. Berkay Celik, Dongyan Xu, and Raheem Beyah. ERACAN: Defending Against an Emerging CAN Threat Model. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024.

[34] Armin Wasicek, Mert D Pesé, André Weimerskirch, Yelizaveta Burakova, and Karan Singh. Context-aware intrusion detection in automotive control systems. In *ESCAR USA*, 2017.

[35] Clinton Young, Habeeb Olufowobi, Gedare Bloom, and Joseph Zambreno. Automotive intrusion detection based on constant can message frequencies across vehicle driving modes. In *ACM Workshop on Automotive Cybersecurity*, 2019.