Electronic Control Unit Discrimination Using Wired Signal Distinct Native Attributes

Rahn Lassiter, Scott Graham, Timothy Carbino, Stephen Dunlap

Chapter 6

ELECTRONIC CONTROL UNIT DISCRIMINATION USING WIRED SIGNAL DISTINCT NATIVE ATTRIBUTES

Rahn Lassiter, Scott Graham, Timothy Carbino and Stephen Dunlap

Abstract A controller area network bus is a communications system used in modern automobiles to connect the electronic control units that implement normal vehicular operations as well as advanced autonomous safety and driver comfort features. However, these advancements come at the expense of vehicle security – researchers have shown that automobiles can be hacked by compromising electronic control units or by connecting unauthorized devices to the controller area network bus.

Physical layer device fingerprinting is a promising approach for implementing vehicle security. This chapter presents a fingerprinting method and classification algorithm for electronic control unit discrimination. Cross-lot discrimination is assessed using four Toyota Avalon electronic control units with different lot numbers as authorized devices, and a BeagleBoard, Arduino and CANable as rogue devices. The experiments yielded perfect rejection rates for rogue devices with false credentials and access denial rates exceeding 98% for authorized electronic control units with false credentials. Additionally, an average correct classification of approximately 99% was obtained for authorized devices.

Keywords: CAN bus, electronic unit discrimination, rogue device detection

1. Introduction

As automobiles become more technologically advanced and connected, they are more susceptible to hacking. Research funded by the U.S. Defense Advanced Research Projects Agency (DARPA) exposed several security vulnerabilities [9, 10]. In particular, using a laptop with wireless connectivity, researchers were able to attack vehicles as they were being driven on highways – remotely turn off the engines, activate the windshield wipers and wiper fluid releases, and

even disable the brakes at speeds below 15 mph. These threats are not limited to automobiles. Heavy vehicles, ships and aircraft are also vulnerable because they have electronic control systems connected in on-board networks.

The technology needed to perform attacks on vehicles is more accessible. In 2014, security researchers developed the CAN Hacking Tool targeting the controller area network (CAN) bus in modern vehicles – the tool costs less than $20 to build; it is the size of an iPhone and can be hooked up to a vehicle within five minutes [15]. Developmental boards such as Arduino and BeagleBoard can be programmed to emulate automobile electronic control units (ECUs) that provide gateways for hackers to compromise CAN bus systems. Although these "hobbyist" experiments may seem harmless, the same technology can be used to carry out serious hacking attacks on vulnerable vehicles.

This research demonstrates that wired signal distinct native attributes (WSDNA) can be leveraged to detect rogue devices such as the CAN Hacking Tool. The approach uses wired signal distinct native attribute fingerprinting and multiple discriminant analysis with maximum likelihood to identify (classify) and authenticate (verify) devices based on their unique signal variations. The experiments conducted during this research yielded perfect (100%) rejection rates for rogue devices with false credentials and access denial rates exceeding 98% for authorized electronic control units with false credentials. Additionally, an average correct classification of approximately 99% was obtained for authorized devices.

2. CAN Bus

CAN bus is a lightweight, broadcast communications system created in the 1980s by Bosch as a replacement for the older wiring systems used in automobiles [8]. The CAN bus system comprises multiple networked electronic control units that transmit, receive and process critical data such as vehicle speed, engine RPM and even the angle of the steering wheel. The latest CAN 2.0 version used in modern vehicles transmits data at speeds up to 1 Mbps. The CAN bus has two message formats: (i) base frame format; and (ii) extended frame format. This work focuses on devices that transmit data in the base frame format, which is specified in Table 1.

CAN signals are transmitted as non-return-to-zero (NRZ) encoded differential voltages. A differential voltage is the difference between the twisted pair CAN-Hi and CAN-Lo signals [6]. The bits in the base frame format are formed from the differences between the CAN-Hi and CAN-Lo signals [20]. A dominant bit (0) is transmitted when the difference between CAN-Hi and CAN-Lo is approximately 2 volts and a recessive bit (1) is transmitted when the difference between CAN-Hi and CAN-Lo is approximately 0 volts, as shown in Figure 1.

The CAN bus is a broadcast network where electronic control units transmit freely to all devices that are listening, or to devices that request information. An electronic control unit that needs to send data attempts to do so when the CAN bus is in an idle state. If multiple electronic control units transmit

Table 1. Typical base frame format [8].

Bits	Name/Field	Description
1	Start of Frame	Always dominant (0)
11	Identifier	Varies for each electronic control unit; also determines priority
1	Remote Transmission Request	Dominant for data frame (0)
1	Identifier Extension Bit	Difference between base frame and extended frame; dominant for base (0)
1	Reserved Bit	Must be dominant (0)
4	Data Length Code	Determines data length (bytes)
0-64	Data	Transmitted data
15	CRC	Checksum
1	CRC Delimiter	Recessive (1)
1	Acknowledgement Bit	Recessive (1)
1	Acknowledgement Delimiter	Transmitter sends recessive (1)
7	End of Frame	All recessive; end of transmission
7	Interframe Spacing	All recessive; time required to process message

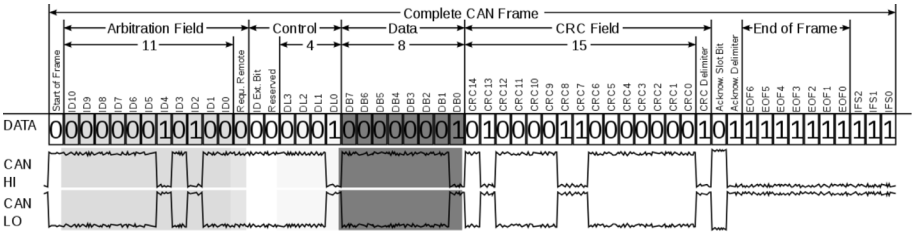


Figure 1. Base frame format [20].

messages at the same time, then the transmissions are synchronized at their start of frame bits and an arbitration occurs in the network.

During synchronization, each identical bit is coherently combined to produce a waveform that has the same voltage for a one or a zero. The device with the lowest identifier number, which indicates higher priority, wins the arbitration and continues to transmit while the device that loses the arbitration stops transmitting as shown in Figure 2. Because of the potential for multiple electronic control units to transmit simultaneously in the arbitration field, Choi et al. [6] determined that the identifier may not be the best region to use to calculate statistical features for fingerprints in a typical CAN bus environment. Instead, they employed the extended identifier in the extended frame format used by electronic control units.

This work focuses on electronic control unit discrimination in a collision-free environment and proposes the use of a region of interest (ROI) after the arbitration field to address the issue of CAN bus collisions. The arbitration field

	SOF	Identifier						
ECU 1	0	0	0	0	1	0	1	0
ECU 2	0	0	0	1	stops transmitting			
CAN Bus	0	0	0	0	1	0	1	0

Figure 2. CAN bus arbitration.

in the base frame comprises identifier bits and the remote transmission request bit whereas the control field comprises the identifier extension bit, reserved bit and four data length code bits as shown in Figure 1. These bits are utilized as the region of interest for fingerprint generation.

3. Device Fingerprinting

This section discusses related work in the area of device fingerprinting and the radio frequency distinct native attribute (RF-DNA) methodology for device classification and discrimination.

3.1 Related Work

Several fingerprinting methods have been proposed for intrusion detection and security controls in CAN bus systems. Early attempts at electronic control unit discrimination employed a mean-squared error and convolution approach, achieving classification rates ranging from 90% to 100% [14]. Device identification was attempted using the identifier field in the base frame format used by electronic control units, but this was deemed to be unreliable [6].

Cho and Shin [5] developed a CAN bus simulation using multiple Arduino Unos with CAN shields; electronic control unit signals were acquired from real vehicles. Their fingerprinting approach leveraged the internal clocks of electronic control units to identify the transmitting devices. The fingerprints were generated based on the clock offset, clock frequency and clock skew. A recursive least-squares algorithm was used for electronic control unit detection and verification, achieving about 97% success in device detection.

The majority of fingerprinting methods employ statistical properties of signals and machine learning or neural net classifiers to identify unique attributes in the extracted features [1, 6, 11]. Avatefipour et al. [1] used a CAN transceiver and development board setup to simulate the CAN bus and electronic control units. Choi et al. [6] employed CAN boards connected in a physical network to simulate the CAN bus and various electronic control units. Jaynes et al. [11] plugged a device directly into the on-board diagnostics port (OBD-II) in a vehicle for electronic control unit signal acquisition. The three methods used different signal collection methods but similar fingerprint generation techniques and neural network classifiers, yielding correct classifications up to 98.6% in the case of Avatefipour et al. [1], 96.5% in the case of Choi et al. [6] and 86% in the case of Jaynes et al. [11].

3.2 RF-DNA Methodology

The radio frequency distinct native attribute (RF-DNA) methodology was developed to perform tasks such as detecting rogue devices, identifying aging devices and augmenting bit-level security [4, 16, 18, 21]. Radio frequency emissions are captured from devices and distinct native attributes of the emissions are generated based on the statistical features of signal amplitude, frequency and phase [4, 7, 13, 16, 18, 22].

Time-Domain Fingerprinting. Time-domain (TD) radio frequency fingerprints are generated from the instantaneous responses of signals, which include the instantaneous amplitude, instantaneous frequency and instantaneous phase. A discrete real-valued signal $s(k)$ is broken up into I-Q samples using the Hilbert transform [4]:

$$s(k) = s_I(k) + s_Q(k) \tag{1}$$

where the amplitude $a(k)$, frequency $f(k)$ and phase $\phi(k)$ are computed as:

$$a(k) = \sqrt{s^2(k)} \tag{2}$$

$$\phi(k) = \tan^{-1}[\frac{s_Q(k)}{s_I(k)}] \tag{3}$$

$$f(k) = \frac{1}{2\pi}[\frac{d\phi(k)}{dk}] \tag{4}$$

Features are typically centered and normalized using the mean and maximum values of the respective time-domain responses [21]. An invariant region, such as the preamble, mid-amble or post-amble, is identified as the region of interest. The region of interest is divided into N_R equal subregions. Usually, the entire region of interest is included as a subregion to produce $N_R + 1$ subregions for statistical feature extraction.

Typical features that are extracted include the standard deviation σ, variance σ^2, skewness γ and kurtosis κ. These statistics are computed for a subregion to generate the fingerprint F_{RF_i}. The fingerprints corresponding to a region are concatenated to form the composite fingerprint F_{RF}:

The fingerprints are expressed by the following equations:

$$F_{RF_i}^{RF} = [\sigma_{R_i}, \sigma_{R_i}^2, \gamma_{R_i}, \kappa_{R_i}]_{1\times 4} \tag{5}$$

$$F_{a,\phi,f}^{RF} = [F_{R_1}^{RF} : F_{R_2}^{RF} : F_{R_3}^{RF} : \cdots : F_{R_{N+1}}^{RF}]_{1\times[4(N_R+1)]} \tag{6}$$

$$F_C^{RF} = [F_a^{RF} : F_\phi^{RF} : F_f^{RF}] \tag{7}$$

The features included in an RF-DNA fingerprint comprise the number of responses N_{resp}, number of statistical features N_{stat} and number of subregions N_R. For example, if $N_{resp} = 4$, $N_{stat} = 3$ and $N_R = 9$, then the number of features $N_{feat} = 4 \times 3 \times 9 = 108$ [4].

The wired signal distinct native attribute (WS-DNA) fingerprinting approach, which is based on the RF-DNA process, is adopted in the WS-DNA methodology used in this research. The composite WS-DNA fingerprints are given by:

$$F_C^{WS} = [F_a^{WS} : F_\phi^{WS} : F_f^{WS}] \tag{8}$$

WS-DNA signals are acquired directly from the wire of a transmitting device instead of over-the-air captures of radio frequency emissions from the device as in the case of the RF-DNA methodology [2–4, 12, 17–19].

Multiple Discriminant Analysis Maximum Likelihood. Device fingerprints are compared using a multiple discriminant analysis maximum likelihood (MDA/ML) classifier. Multiple discriminant analysis is a dimensionality reduction algorithm that takes the extracted features or fingerprints and reduces them to $N-1$ classes, where N is the number of devices. The maximum likelihood classifier assumes that the data has a Gaussian distribution, equal priors and uniform costs. The classifier establishes thresholds based on training fingerprints and assigns each test fingerprint to a class using Bayesian decision criteria [22].

Additionally, K-fold cross-validation is used to increase reliability [4]. Cross-validation is accomplished by: (i) dividing the training fingerprints into K equal blocks; (ii) holding one block out and conducting training with the remaining $K-1$ blocks; (iii) conducting testing using the block that was held out; and (iv) repeating the process until all the blocks have been held out. The iteration that produces the highest score is used for model development [3, 21].

Device discrimination is a two-step process comprising classification and verification. Classification is a one-vs-many assessment that determines which training fingerprint best matches a testing fingerprint. Verification is a one-vs-one assessment that determines how similar the identity of a claimed fingerprint is to the identity of the actual fingerprint [3, 16].

4. Experimental Methodology

This section discusses the experimental setup and collection as well as the parameters used in the WS-DNA fingerprinting methodology.

4.1 Device Under Test and Signal Collection

The device under test (DUT) was a steering angle sensor (SAS) from a Toyota Avalon. This electronic control unit transmits a data frame or burst in the base frame format approximately every 260 μs as shown in Figure 3. The

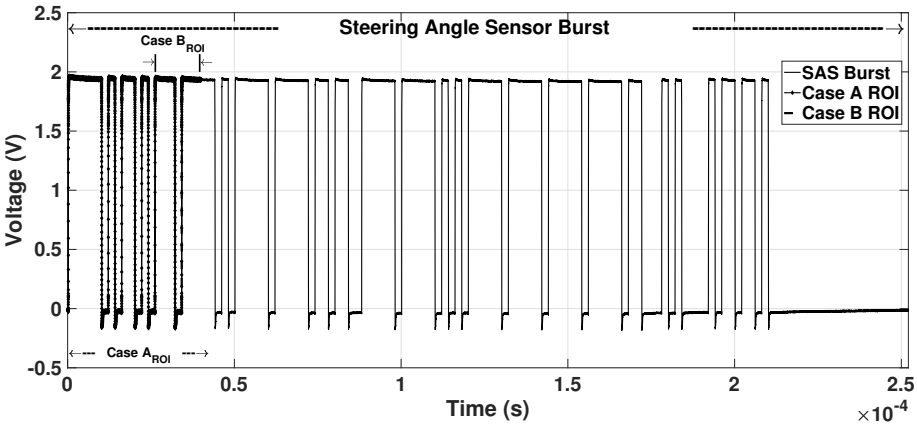


Figure 3. Data frame or burst from a Toyota SAS.

steering angle sensor was chosen for the experiments because it has a relatively high priority on the CAN bus of a Toyota Avalon and because it continuously transmits data with or without user input.

Table 2. Devices under test (four-class cross-lot discrimination).

Device	Device ID	Lot	Average $\mathbf{SNR_C}$
1	SAS 1 (A1)	503G	42.9 dB
2	SAS 2 (A2)	823F	42.4 dB
3	SAS 3 (A3)	826I	43.5 dB
4	SAS 4 (A4)	523E	43.4 dB

Four devices ($N_C = 4$ classes), each from a different lot, were used to assess the cross-lot discrimination (Table 2).

Table 3. Rogue devices used for authentication testing.

Rogue Device ID	Description
R1	BeagleBoard; ISO 1050 CAN transceiver
R2	Arduino Uno with CAN shield
R3	CANable

Additionally, three rogue devices ($N_{rg} = 3$) were created to present false credentials during attempts to access the CAN bus as authorized devices. Table 3 provides information about the rogue devices.

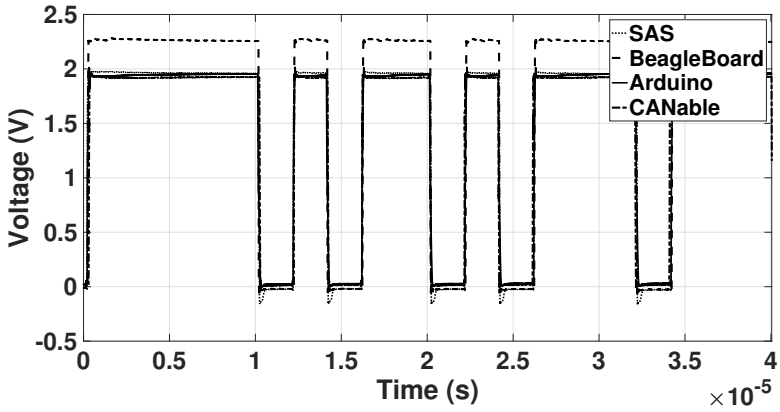


Figure 4. Average region of interest responses for all the devices.

Figure 4 shows the average differential voltage waveform of the region of interest of the steering angle sensors compared with those of the rogue devices. All the devices transmit the same bit-level data and should be accepted as authorized devices on the CAN bus. On average, rogue device R1 has a maximum differential voltage that is 0.2 V greater than those of the other rogue devices as well as the steering angle sensors as shown in Figure 4.

A Keysight InfiniiVision MSOX3054T 5.0 GHz oscilloscope operating at f_s = 1 GSPS was used to collect and store the baseband signals from the Toyota steering angle sensors. A total of 260 ms of signals were collected, which comprised $N_{bursts} \approx 1,000$ bursts. To reduce environmental and collection bias, a random permutation of five collections of $N_{bursts} \approx 200$ bursts for each device were taken over a one-week period at various times and various temperatures. To further reduce experimental variability, each steering angle sensor was locked into the same position so that all the devices transmitted the same 64-bit message and all the devices used the same power supply.

MATLAB was used to process the unfiltered signals and generate WS-DNA fingerprints. Each burst or data frame was extracted by cross-correlating the collected signal with an ideal preamble reference signal and each burst was aligned at the same starting index in a fingerprint generation matrix. Prior to fingerprint generation, a fourth-order baseband Butterworth filter was used to reduce noise. The estimated average collected signal-to-noise ratio (SNR) was computed by taking the ratio of the average power of the region of interest to the average power of the noise region before the start of frame, yielding a signal-to-noise ratio $SNR_C \approx 43.1$ dB.

4.2 Signal-to-Noise Ratio Scaling

Multiple noise realizations were required for fingerprint generation. Although every effort was taken to reduce the effects of environmental noise,

additive white Gaussian noise (AWGN) was assumed to be present in signals from the power supply, oscilloscope and collection probes. However, because this noise does not demonstrate the effects of different channel conditions, different iterations of like-filtered, power-scaled independent additive white Gaussian noise were added during post-processing to simulate different channel conditions.

In the experiments, noise was added to produce $-46\,\text{dB} < SNR_{\Delta} < 0\,\text{dB}$ in 2 dB increments, where SNR_{Δ} denotes the reduction in the signal-to-noise ratio under the collected conditions as the power of the additive white Gaussian noise was increased. In this work, SNR_{col} (collected conditions) denotes the signal-to-noise ratio where the classification performance is statistically equal to the classification performance at SNR_C. To be clear, the signal-to-noise ratio was never improved. Instead, additive white Gaussian noise was added to each burst until the average correct classification $\%\text{C} \approx 1/N_C$ was obtained.

4.3 Fingerprint Generation

Fingerprints were generated for the ideal, collision-free environment to: (i) assess the WS-DNA classification and verification performance using an entire invariant region of interest; and (ii) use a comparable amount of bits as in [6] to provide a performance estimate for the WS-DNA implementation for electronic control units using the extended frame format. This set of fingerprints does not represent a realistic CAN bus scenario because collisions occur frequently, but the fingerprints could be used to establish a baseline for the electronic control units prior to installation in a vehicle. A second set of fingerprints was generated to address the best region of interest for the WS-DNA implementation for electronic control units using the base frame format on the CAN bus in a realistic environment.

- **Case A (Ideal Collision Free Environment):** Time-domain WS-DNA fingerprints were generated using the steering angle sensor preamble with the region of interest comprising the start of frame, arbitration field and control field. Additionally, $N_{samp} = 210$ samples were included before the start of frame bit, resulting in a region of interest with $N_{samp} \approx 40,000$ samples. The region of interest was further divided into $N_R = 54$ contiguous subregions each containing $N_{samp} \approx 740$ samples (Figure 5).

  The total number of features N_{feats} included in the WS-DNA fingerprints is equal to $N_{resp} \times N_{stats} \times N_R + 1$. Thus, $N_{feats} = 3 \times 4 \times 55 = 660$ features. Fingerprints for the $N_{rg} = 3$ rogue devices were generated along with the authorized devices using the same fingerprint generation method.

- **Case B (Realistic CAN Bus Environment):** Time-domain WS-DNA fingerprints were generated to address a typical collision environment for electronic control units using the base frame format, but excluding the start of frame and arbitration field. The region of interest for this scenario

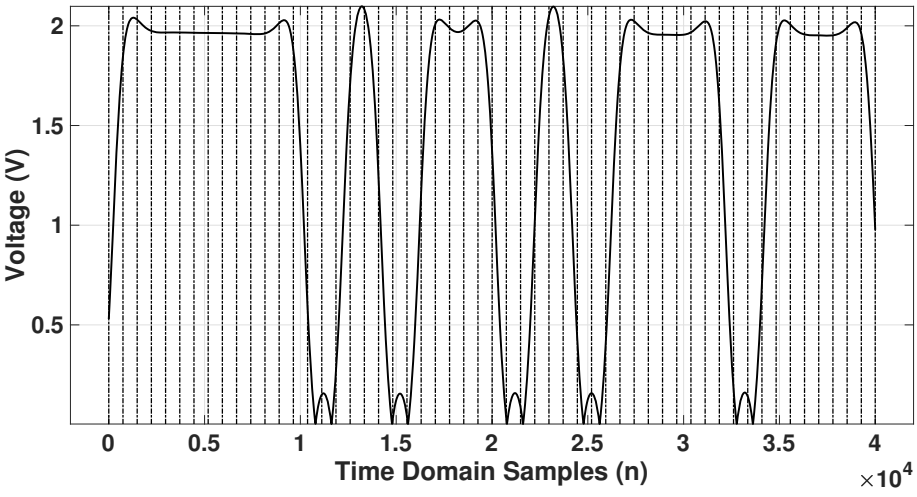


Figure 5. Region of interest divided into $N_R = 54$ subregions.

included the remote transmission request bit, identifier extension bit, reserved bit and the four data length code bits. The region of interest was divided into $N_R = 45$ subregions, each containing $N_{samp} \approx 306$ samples. The total number of features N_{feats} included in the WS-DNA fingerprints is equal to $N_{resp} \times N_{stats} \times N_R + 1$. Thus, $N_{feats} = 3 \times 4 \times 46 = 552$ features.

4.4 MDA/ML Classification and Verification

This section discusses the use of multiple discriminant analysis with maximum likelihood for device classification and device verification:

- **Device Classification:** A total of $N_{NZ} = 5$ noise realizations were used per signal-to-noise ratio to generate a total of $N_{prints} \approx 5,000$ fingerprints. $N_{trng} = N_{test} \approx 2,500$ interleaved training and testing fingerprints per device were used for classification. Additionally, $K = 5$ was used for cross-validation, which is consistent with previous RF-DNA work [18, 21]. Decision thresholds were established during the training phase and testing fingerprints were classified based on the decision region they fell in during the testing phase.

- **Device Verification:** Device verification was implemented using the Euclidean distance as the measure of similarity and an equal error rate (EER) of 10% as the measure of success. In the experiments, the equal error rate (device dependent metric) was chosen such that true verification rate (TVR) was equal to the rogue rejection rate (RRR). The true verification rate corresponds to the number of attempts by an authorized device that are correctly accepted divided by the total number

of attempts. The rogue rejection rate corresponds to the total number of rogue attempts that are correctly rejected divided by the number of attempts.

A probability mass function was generated during training. Device-dependent thresholds $t_V(d)$ were established based on the desired true verification rate and false verification rate (FVR) for authorized device verification and established based on the desired true verification rate and rogue acceptance rate (RAR) for rogue device verification. Note that FVR = 1 – TVR and RAR = 1 – RRR.

During testing, the verification test statistic Z_V was generated from the fingerprint of each unknown device and compared against the threshold t_V. Devices were either granted access or denied access (correctly or incorrectly) depending on how Z_V compared against t_V [3].

Receiver operating characteristic (ROC) curves were generated to present the verification performance using the established verification and acceptance rates. Stem plots were generated to present the results for each of the $N_{test} \approx 5,000$ rogue attempts (for rogue devices R1, R2 and R3) to pass as authorized devices (A1, A2, A3 and A4) [3]. Rogue device acceptance and rejection rates were established using the BeagleBoard, Arduino, CANable and an arbitrarily-chosen fourth device (R4) as rogue devices. The unauthorized rogue devices were excluded from the training so that the classifier would be presented with true rogue devices during the verification phase.

5. Experimental Results

This section presents the results for multiple discriminant analysis with maximum likelihood classification and verification. The classification results are presented using %C versus SNR_Δ plots and confusion matrices. The verification results are presented using ROC curves and stem plots.

5.1 Device Classification

Figures 6 and 7 show the classification results. The confusion matrix results at SNR_{col} are presented in Table 4. All the classification results presented are based on 95% confidence intervals, which are omitted in Figures 6 and 7 for visual clarity because the confidence intervals fall in the vertical extent of the markers.

The results reveal that device SAS 3 has a statistically-significant increase in correct classification over all the other devices from $SNR_\Delta \geq -39\,\text{dB}$ to $SNR_\Delta = -14\,\text{dB}$. Upon further inspection, SAS 3 was verified to have the newest internal components. The classification results for device SAS 2 are statistically equal to the cross-class average. Devices SAS 1 and SAS 4 were incorrectly classified as each other more often than with the other two devices.

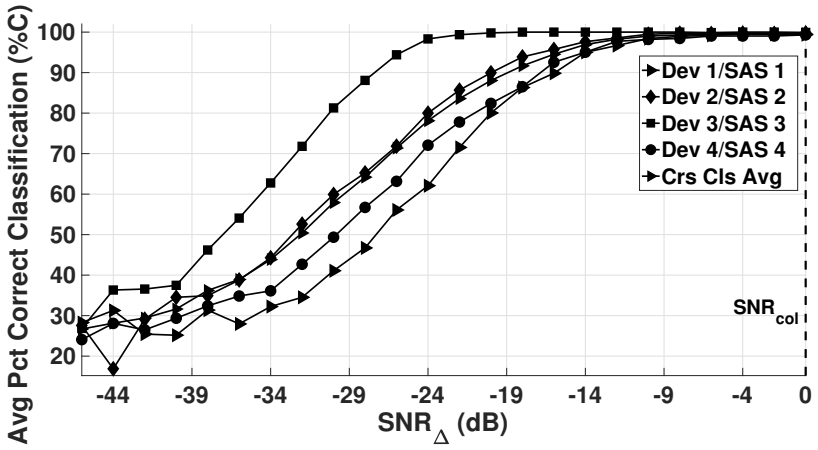


Figure 6. Classification results for $N_C = 4$ classes using the ECU preamble.

These devices were both obtained from used vehicles that were manufactured during the same year.

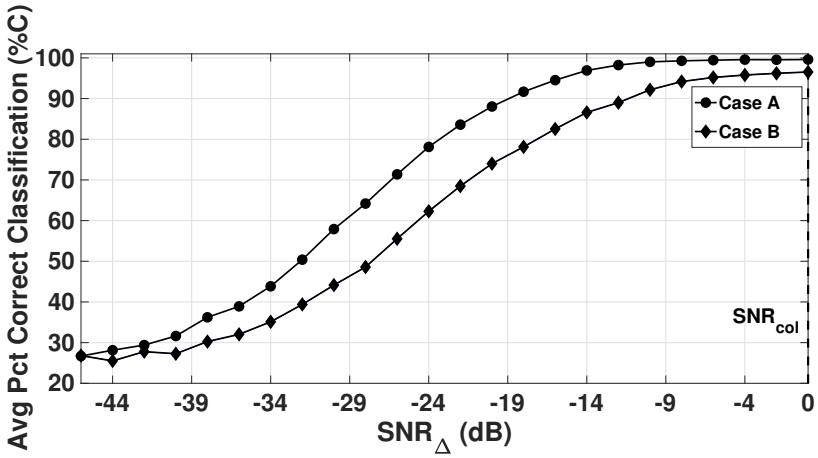


Figure 7. Classification results for cross-class average for Case A and Case B ROIs.

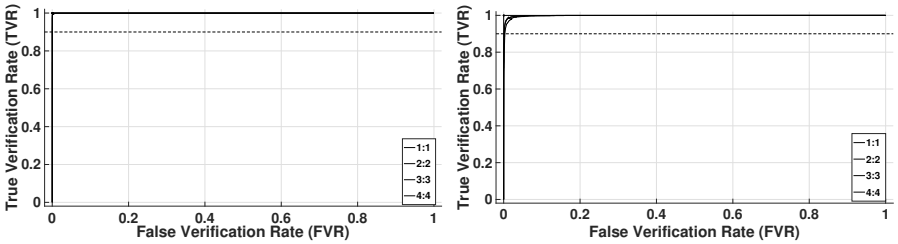
Figure 7 provides a direct comparison of the classification results using the fingerprints generated with the regions of interests used in Case A and Case B. The results are statistically equal at $SNR_\Delta \leq -40\,\text{dB}$. Moreover, using the preamble as the region of interest yielded statistically better classification for $SNR_\Delta > -40\,\text{dB}$. This is arguably the result of having more bits and more bit transitions in the region of interest, which provide more useful time-domain discrimination information.

Table 4. Cross-lot discrimination confusion matrix (%) for $N_C = 4$ classes.

	SAS 1	**SAS 2**	**SAS 3**	**SAS 4**
SAS 1	**99.6**/93.76	**0**/1.88	**0**/0	**0.4**/4.36
SAS 2	**0.04**/1	**99.6**/97.8	**0**/0.04	**0**/1.16
SAS 3	**0**/0.56	**0.04**/1.04	**99.92**/97.88	**0.04**/0.52
SAS 4	**0.28**/2.76	**0.2**/1.92	**0**/0.12	**99.52**/95.2

Table 4 shows the cross-lot discrimination confusion matrix for $N_C = 4$ classes; the results are displayed as %C Case A/%C Case B. The bold values in the table correspond to the classification results for Case A and the non-bold values correspond to the classification results for Case B. The classification performance degraded when the arbitration field was excluded from the region of interest. The classification performance values of devices SAS 1 and SAS 4 were reduced by approximately 5% and the classification performance values of devices SAS 2 and SAS 3 were reduced by approximately 2%. SAS 1 and SAS 4 were confused with each other more often than with the other devices; these devices were obtained from used vehicles manufactured during the same year. As the signal-to-noise ratio was degraded, SAS 1 and SAS 4 were incorrectly classified as each other more often than other devices, which may indicate that these devices look more similar to each other as they age.

Greater than 90% correct identification of similar components was achieved using WS-DNA fingerprints generated in Case B. Moreover, correct classification (%C) greater than 90% in realistic implementations was obtained even when the signal-to-noise ratio was degraded by 10 dB.



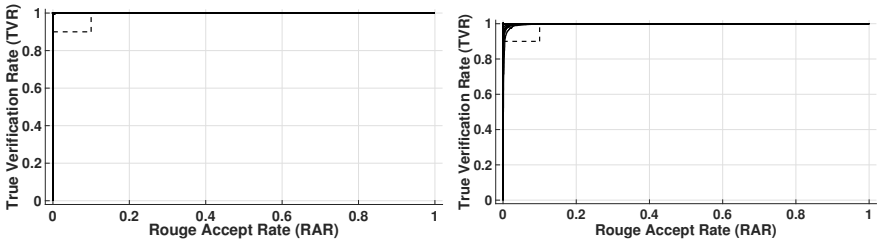
(a) Case A: ROC curve.

(b) Case B: ROC curve.

Figure 8. Authorized device verification ROC curve at SNR_{col}.

5.2 Device Verification

Figure 8 shows the results for authorized device verification. Note that the Euclidean distance was used as the measure of similarity and success was defined as a true verification rate greater than 0.9 and a false verification rate less



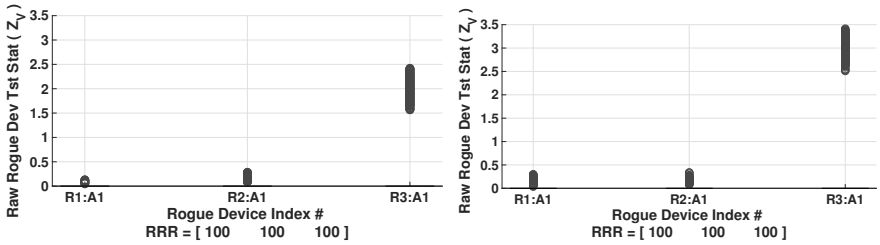
(a) Case A: Rogue ROC curve. (b) Case B: Rogue ROC curve.

Figure 9. Rogue device verification ROC curve at SNR_{col}.

than 0.1. The horizontal black dashed lines correspond to the true verification rate benchmark of 0.9, which is consistent with previous RF-DNA work [4, 7, 16, 18, 21]. The solid ROC curves for Case A and Case B indicate that all four devices satisfy the true verification benchmark at the average collected signal-to-noise ratio.

In the rogue device verification scenario, rogue devices presented false credentials and were either accepted or rejected as the device they claimed to be based on the threshold established by the probability mass function generated during training.

Figure 9 shows the results for rogue device verification. The dashed black boxes represent the areas where the true verification rate is greater than 0.9 and the rogue acceptance rate is less than 0.1. The black stars on each line denote the device-dependent equal error rate and the solid curves denote devices that met the success criteria. Consistent with the authorized device verification results, all the devices successfully met the equal error rate success criteria for Case A and Case B.



(a) Case A: Rogue rejection rate. (b) Case B: Rogue rejection rate.

Figure 10. Rejection rates of rogue devices using valid credentials at SNR_{col}.

Figure 10 shows the rejection rates for unauthorized rogue devices (R1, R2 and R3) using the valid credentials (i.e., ID) of the authorized device A1 at SNR_{col}. The verification results are based on burst-by-burst grant/deny access criteria [4]. Note that the `O` symbols denote access correctly denied and the `X` symbols denote access incorrectly granted. The horizontal black lines correspond to the device-dependent equal error rate thresholds.

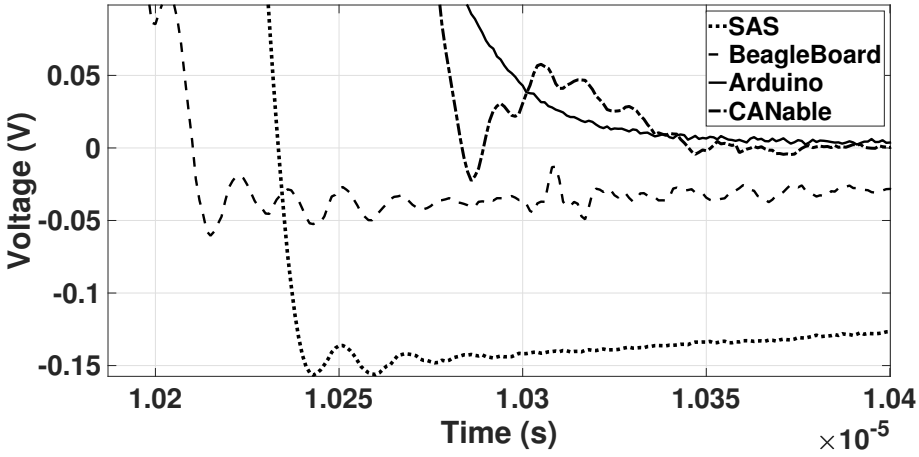
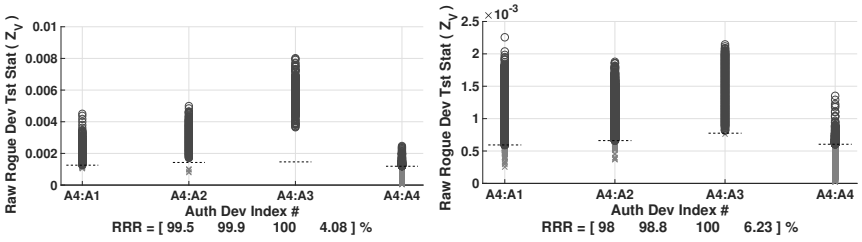


Figure 11. Zoomed-in view of bit transitions in Figure 4.

The rogue devices were rejected 100% of the time in Case A and Case B. The results also indicate that using the smaller region of interest yields rogue device fingerprints that are more similar to the fingerprints of the authorized devices, except for rogue device R3 based on the same vertical and horizontal axes in both figures. Although the rogue rejection rates were perfect for rogue devices R1 and R2, the verification test statistics Z_V generated for these devices were closer to the threshold t_V, indicating a greater similarity in Case B than in Case A.

Rogue device R3 looks less like device SAS 1 in Case B, which is likely due to the symbol and transition misalignment seen in Figure 11, a zoomed-in view of the bit transitions in Figure 4. All the rogue device transitions are slightly misaligned and do not accurately replicate the authorized device transitions.



(a) Case A: Rogue rejection rate.

(b) Case B: Rogue rejection rate.

Figure 12. Rejection rates of a device (A4) using the credentials of other devices.

Figure 12 shows the rejection rates when the compromised device SAS 4 (or A4) presented false credentials belonging to the other three authorized devices (A1, A2 and A3). Note that the `O` symbols denote access correctly denied and the `X` symbols denote access incorrectly granted. The black dashed lines

correspond to the device-dependent equal error rate thresholds. Excluding the results for device A4 presenting its own credentials, the average rogue rejection rate is still approximately 100% when an authorized electronic control unit attempts to present false credentials in Case A. In Case B, the average rogue rejection rate dropped approximately 1%, resulting in a rogue rejection rate of approximately 99%.

Overall, the rogue rejection rates are high for unauthorized devices because the devices were unable to accurately match the authorized electronic control unit symbol rate, resulting in drastic differences in the transition regions as shown in Figure 11. The figure also shows that, although rogue device R1 has a higher average amplitude than the other devices, the bit transitions are more aligned with the authorized devices than the rogue device R3. This results in a greater degree of similarity.

6. Conclusions

Electronic control units in modern automobiles implement normal vehicular operations as well as advanced autonomous safety and driver comfort features. However, the automobiles can be hacked by compromising the electronic control units or by connecting unauthorized devices to the controller area network bus.

The WS-DNA methodology described in this chapter is a viable solution for electronic control unit classification and verification. Although development boards such as Arduino and BeagleBoard can be used to create rogue electronic control units, the differences in their signal transition regions and amplitudes provide enough information to reject these devices when they are compared against authorized electronic control units. When only the message preamble of an electronic control unit was used, 100% of the CAN bus access attempts by three rogue devices were detected. Using an authorized steering angle sensor as a compromised device yielded a rogue device rejection rate greater than 99%, even when a region of interest smaller than the preamble was used. Additionally, the average correct classification of the four authorized devices was greater than 99% at SNR_{col}. As expected, when only seven bits were used as the region of interest in Case B, the classification performance was statistically worse than in Case A. Specifically, in Case B, the average correct classification was approximately 96% at SNR_{col} and the average detection rate for compromised devices was slightly lower than in Case A. Despite the decreased performance, the unauthorized rogue rejection rate was still 100% for Case B, indicating that the WS-DNA methodology is suitable for authenticating base frame format electronic control units. The results are also promising for extended frame format electronic control unit based on the results in Case A.

Security can be established on the CAN bus using the WS-DNA methodology with fingerprints generated from the region of interest used in Case B. A device capable of monitoring and collecting signals could be installed on the CAN bus, programmed with authorized electronic control unit WS-DNA fingerprints as well as an multiple discriminant analysis maximum likelihood classifier. CAN

bus traffic could then be collected and analyzed in real-time to detect the presence of compromised or rogue devices in the network.

The WS-DNA methodology can be applied to a range of CAN bus and electronic control unit discrimination problems. Investigating electronic control unit discrimination for the extended frame format could validate the claims made in Case A. Like model discrimination – differentiating between electronic control units from the same manufacturer and with the same lot number – should also be examined, although it is a more difficult aspect of RF-DNA discrimination [3]. Additionally, discriminating between vehicle electronic control units with different functions such as a steering angle sensor, engine control module and telematic control unit would be beneficial. Finally, discriminating between CAN transceivers and evaluating the temperature effects on fingerprinting and discrimination are also promising topics for future research.

The views expressed in this chapter are those of the authors, and do not reflect the official policy or position of the U.S. Air Force, U.S. Department of Defense or U.S. Government. This document has been approved for public release, distribution unlimited (Case #88ABW-2019-0050).

References

[1] O. Avatefipour, A. Hafeez, M. Tayyab and H. Malik, Linking received packets to the transmitter through physical-fingerprinting of controller area network, *Proceedings of the IEEE Workshop on Information Forensics and Security*, 2017.

[2] T. Carbino, Exploitation of Unintentional Ethernet Cable Emissions Using Constellation Based-Distinct Native Attribute (CB-DNA) Fingerprints to Enhance Network Security, Ph.D. Dissertation, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2015.

[3] T. Carbino, M. Temple and J. Lopez, A comparison of PHY-based fingerprinting methods used to enhance network access control, in *ICT Systems Security and Privacy Protection*, H. Federrath and D. Gollmann (Eds.), Springer, Cham, Switzerland, pp. 204–217, 2015.

[4] T. Carbino, M. Temple and J. Lopez, Conditional constellation based distinct native attribute (CB-DNA) fingerprinting for network device authentication, *Proceedings of the IEEE International Conference on Communications*, 2016.

[5] K. Cho and K. Shin, Fingerprinting electronic control units for vehicle intrusion detection, *Proceedings of the Twenty-Fifth USENIX Security Symposium*, pp. 911–927, 2016.

[6] 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(6), pp. 4757–4770, 2018.

[7] W. Cobb, E. Garcia, M. Temple, R. Baldwin and Y. Kim, Physical layer identification of embedded devices using RF-DNA fingerprinting, *Proceedings of the Military Communications Conference*, pp. 2168–2173, 2010.

[8] S. Corrigan, Introduction to the Controller Area Network (CAN), Application Report SLOA101, Texas Instruments, Dallas, Texas, 2002.

[9] R. Currie, Developments in Car Hacking, Information Security Reading Room, SANS Institute, North Bethesda, Maryland, 2015.

[10] A. Greenberg, Hackers remotely kill a Jeep on the highway – With me in it, *Wired*, July 21, 2015.

[11] M. Jaynes, R. Dantu, R. Varriale and N. Evans, Automating ECU identification for vehicle security, *Proceedings of the Fifteenth IEEE International Conference on Machine Learning and Applications*, pp. 632–635, 2016.

[12] J. Lopez, N. Liefer, C. Busho and M. Temple, Enhancing critical infrastructure and key resources (CIKR) level-0 physical process security using field device distinct native attribute features, *IEEE Transactions on Information Forensics and Security*, vol. 13(5), pp. 1215–1229, 2018.

[13] M. Lukacs, P. Collins and M. Temple, Device identification using active noise interrogation and RF-DNA "fingerprinting" for non-destructive amplifier acceptance testing, *Proceedings of the Seventeenth Annual IEEE Wireless and Microwave Technology Conference*, 2016.

[14] P. Murvay and B. Groza, Source identification using signal characteristics in controller area networks, *IEEE Signal Processing Letters*, vol. 21(4), pp. 395–399, 2014.

[15] P. Paganini, CAN hacking tools, 20 USD to hack a car remotely, *Security Affairs*, February 9, 2014.

[16] D. Reising, M. Temple and J. Jackson, Authorized and rogue device discrimination using dimensionally-reduced RF-DNA fingerprints, *IEEE Transactions on Information Forensics and Security*, vol. 10(6), pp. 1180–1192, 2015.

[17] B. Ross, T. Carbino and S. Stone, Physical-layer discrimination of power line communications, *Proceedings of the International Conference on Computing, Networking and Communications*, pp. 341–345, 2017.

[18] B. Ross, T. Carbino and M. Temple, Home automation simulcasted power line communications network (SPN) discrimination using wired signal distinct native attribute (WS-DNA), *Proceedings of the Twelfth International Conference on Cyber Warfare and Security*, pp. 313–322, 2017.

[19] B. Ross, T. Carbino and M. Temple, Simulcasted power line communications network (SPN) configuration validation for home automation applications using wired signal distinct native attribute (WS-DNA) fingerprinting, *Journal of Information Warfare*, vol. 16(3), pp. 95–118, 2017.

[20] Wikipedia Contributors, CAN-Bus-Frame in Base Format without Stuffbits, *Wikipedia Commons* (`commons.wikimedia.org/wiki/File:CAN-Bus-frame_in_base_format_without_stuffbits.svg`), 2017.

[21] M. Williams, S. Munns, M. Temple and M. Mendenhall, RF-DNA fingerprinting for airport WiMax communications security, *Proceedings of the Fourth International Conference on Network and System Security*, pp. 32–39, 2010.

[22] M. Williams, M. Temple and D. Reising, Augmenting bit-level network security using physical layer RF-DNA fingerprinting, *Proceedings of the IEEE Global Telecommunications Conference*, 2010.