Authenticating the Sender on CAN Bus using Inimitable Physical Characteristics of the Transmitter and Channel

by

Muhammad Tayyab

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering
(Computer Engineering)
in the University of Michigan-Dearborn
2018

Master's Thesis Committee:

Associate Professor Hafiz Malik, Chair
Professor Selim Awad
Assistant Professor Lu Wei

This thesis is dedicated to my beloved mother and family for their endless love, support, and encouragement

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACK	Acknowledgement
ADAS	Advanced Driver Assistant Systems
ANN	Artificial Neural Network
CAN	Controller Area Network
CAN-FD	Controller Area Network – Flexible Data Rate
DoS	Denial of Service
EOF	End of Frame
ECU	Electronic Control Unit
FOTA	Firmware Update Over the Air
GMLAN	General Motors Local Area Network
GPS	Global Positioning System
IDS	Intrusion Detection System
IP	Internet Protocol
LIN	Local Interconnect Network
OBD-II	On Board Diagnostics
PUF	Physical Unclonable Functions
REC	Receive Error Counter
RF	Radio Frequency
RFID	Radio Frequency Identification
RKE	Remote Keyless Entry

SOF Start of Frame

TCP/IP Transmission Control Protocol / Internet Protocol

TEC Transmit Error Counter

TMC Traffic Message Channel

V2I Vehicle to Infrastructure

V2V Vehicle to Vehicle

ABSTRACT

The Cybersecurity for the embedded systems has become a serious challenge in the recent times. Given that the embedded applications are being connected with each other and over the public internet while running the relatively fragile low-density code, they are prone to a wide range of attacks. These attack surfaces are inherent to most of the embedded applications. One such example is a modern automobile. A modern vehicle consists of a network of small electronic computers known as Electronic Control Units (ECUs), which makes possible the state-of-the art features. Because of the power of these tiny computers and the artificial intelligence, autonomous vehicles will be on the road for public use in near future. These vehicles will be connected over the internet and hence susceptible to the broad range of attacks. The problem gets worse in the automotive applications because of the presence of very weak internal networking protocols.

The ECUs are connected via each other over Controller Area Network (CAN) Bus which lacks the basic security features. It does not provide the authenticity of the message sender and the payload integrity is absent as well. In this paper, we have proposed a novel idea to solve both of these problems based on the physical fingerprinting the transmitter of the message packet. Electrical devices are unique in terms of the physical fingerprints, they leave in the transmitted messages due to the material's microstructure. This uniqueness exists in the time domain as well as the frequency domain of the signals. We have proposed various techniques to capture this uniqueness using the signal processing techniques at the message receiver side which will be able to link the received packet to the original transmitter. We have applied the Neural Network based Classifier in order

to realize an Intrusion Detection System proof of concept. Our proposed idea, realized with different techniques, has been proven to be more efficient than the state-of-the art intrusion detection systems.

We have analyzed the weaknesses in one of the advanced security techniques based on fingerprinting the clock behaviors of the message sender. We were able to launch the successful attack to bypass the intrusion detection system based on fingerprinting the clock behavior of the sender. Our work demonstrates the wide range of attacks: the external attacks by exploiting the in-vehicle infotainment system, internal attacks and a possible defense mechanism as well. We have summarized the possible attack vectors on our proposed idea as well with the challenges being faced for the real-world implementation.

CHAPTER 1: Introduction

Today cybersecurity is one of the biggest challenges being faced globally. The challenges are even daunting when it comes to the world of embedded systems, controlling the simple but critical operations of daily life, from pacemakers to the microwave ovens. A minor tampering with these devices may lead to the complete malfunction which may risk the thousands of costly human lives along with the huge financial loss. Modern automobiles fit this description of these threats perfectly. We will be using modern automobile's embedded networks as proof of concept for our work in this paper. However, the presented work is theoretically valid for other applications of embedded systems and embedded networks.

Automotive industry is heading towards the realization of autonomous cars. Many OEMs have already started the testing of commercial autonomous vehicles on the public roads and will be available for the domestic customers in near future. These state-of-the-art vehicles will be connected over the internet providing numerous features.

Nowadays, most of the features in the modern vehicles are based on the embedded systems. A typical vehicle is equipped with 70-100 such tiny computing devices known as Electronic Control Units (ECUs) [1]. These ECUs communicate with each other via different kind of networks. Some of these networks are very simple, robust and highly reliable while some can be complex. Controller Area Network (CAN) is relatively a simple but widely deployed networking protocol and act as the backbone of the in-vehicle communication. The simplicity and robustness in the

design of CAN protocol makes it difficult to provide the security and contains numerous design weaknesses. These inherent design-based weaknesses were mostly exploitable if the attacker has given the physical access and the vehicles were relatively securer as long as the attacker did not get the physical access. Now the concept of connected car is becoming popular and most of the newly manufactured vehicles are equipped with the internet connectivity. This aggravates the security because of increased attack surfaces and the attackers have the capability to compromise the vehicles remotely. This is the point where the security becomes a challenge for most of the internet of things concept i.e. connected embedded applications. When the internet connectivity meets the insecure inside networks, the inherent weaknesses in the legacy protocols become exploitable leading to the catastrophic outcomes.

CAN serves as the backbone network for the in-vehicle communication between various Electronic Control units (ECUs). It is a legacy network developed in late 1980s and was adopted by the automotive industry quickly [2]. The main reason was the robustness and simplicity. But it lacked the basic security features. It does not provide any of the basic security functionalities known as Confidentiality, Integrity and Authenticity. (C.I.A). Though the integrity is provided through CRC field but that is easy to bypass and the purpose of this is to detect the transmission errors not the malicious tampering of the packets [3,4,5]. There is no information about the sender of the message in the header as well. The architecture of CAN is strictly the bus based so, any node on the bus is able to connect and transmit the message. This breaks the notion of the authenticity of at all.

In this paper, we will be addressing the problem of authenticity primarily and we will be able to provide Integrity as well. The confidentiality is not a problem as of now in the CAN network because the nature of the communication content is primarily related to the arbitrary functionality

of the vehicle and does not contain the sensitive information and can be read by plugging in to the OBD-II port of the vehicle. The problem arises when the malicious packets are injected and cannot be verified by the receivers.

1.1. Our Contribution:

We have proposed the idea of authenticating the sender of the message based on the physical fingerprints of the transmitters. We are exploiting the presence of natural randomness in the microstructure of the material, due to which the electrical devices become unique and naturally leave the fingerprints in the transmitted signal at the physical layer. If these fingerprints can be estimated accurately from the physical signals, the transmitter of the signals can be authenticated. We have analyzed the limitations of this approach and have shown the efficiency of this method. We have used CAN network as a proof of concept here however the proposed techniques can be extended to other embedded networks along with many other applications where the source of the electrical signal needs to be authenticated for example digital forensics etc.

Physical fingerprints of the message sending nodes are being used by different other researchers as well to authenticate the senders in networks e.g. Tak Cho and Kang G. Shin has proposed a Clock based Intrusion Detection System (CIDS) which fingerprints the unique attributes of the clock of the message transmitter [6]. We have analyzed its security and have proposed a successful attack.

To follow the paper, we are providing a brief overview of the CAN protocol in the next section. This can be skipped easily if the reader is already familiar with the architecture of CAN bus.

CHAPTER 2: CAN Overview

CAN is one of the widely used networking protocols for embedded systems-based applications. It was developed in 1983 by Robert BOSCH GmbH and was immediately adopted by the automotive industry [2]. The reason behind this popularity was the simplicity and the robustness of the protocol which made it easier to replace the peer to peer connection network topology in the automobiles at that time. The primary goal of CAN protocol was to replace the complex wiring scheme deployed in the vehicles at that time. CAN protocol is much simpler and easier to implement as compared to TCP/IP based networks which are too complex for real-time applications. Though now Ethernet based networking is becoming popular in automotive industry but still CAN is considered the primary channel for in-vehicle communication.

The description of CAN protocol can be divided into the physical layer implementation and the data link layer implementation according to the ISO 11898 protocol.

2.1. Physical Layer Implementation:

There are different variants of CAN protocol when it comes to the physical layer implementation.

1. Two wire CAN bus
2. Single wire CAN bus
3. Fault-tolerant CAN bus

However, out of these two wire CAN bus is considered to be the standard one and is deployed most. Other variants are relatively less deployed, so we will not be discussing the details of those variants. Hence forth, the CAN bus will imply the two-wire implementation of the CAN bus.

CAN bus is constructed based on the differential signal scheme and hence physically consists of two channels, CAN High and CAN Low [8,9]. These CAN channels are twisted together as well in order to have minimum errors in signal propagation in the noisy environments. In practical implementations, CAN Bus is terminated with a 130 Ohms resistor as well at the end to diminish the electromagnetic reflection.

In idle state, CAN High and CAN Low are driven at 2.5 Volts. This state is equivalent to transmitting 1. The bit '1' is also known as the Recessive bit. In order to transmit a '0' CAN High is driven to 3.75 volts while CAN Low is driven to 1.25 volts. Hence '0' is given the name dominant bit. These thresholds may vary a little depending upon the vendors.

As mentioned earlier, CAN is a bus-based networking topology, hence every node on the network receives the signal. It is a broadcast-based protocol which enables every connected node to transmit the data.

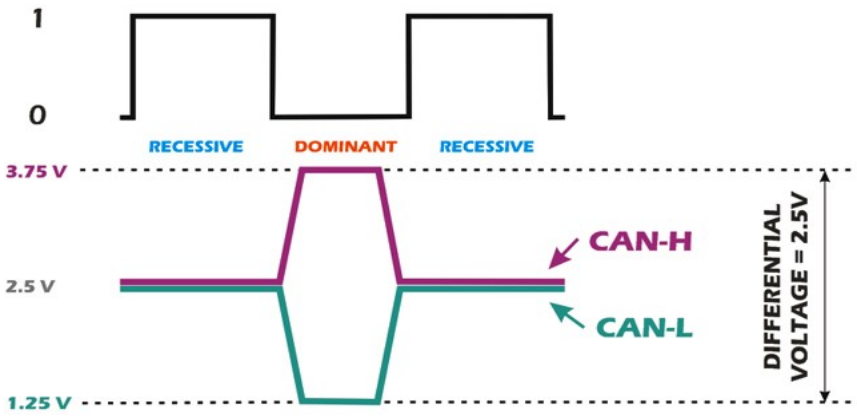


Figure 2.1.1. CAN Bus Signal [7]

CAN bus also offers different data rates on the bus, usually 125 kbps, 250 kbps, 500 kbps and 1000 kbps. This also enables the complex automotive architecture for the networking needs. This is the reason we find different CAN buses in a vehicle, usually in the name of High speed CAN bus, Low speed CAN bus and the Diagnostic CAN bus.

2.2. Packet Architecture:

Basically, there are three types of packets in CAN network.

- Data Frame
- Control Frame
- Error Frame
- Overload Frame

In this paper we will be focusing more on the data packets.

2.2.1. Data Frames:

Data packets are the most common types of the CAN protocol packets. As evident from name the data packets contain the data as a payload and transmit it. The basic structure of the Data frame is shown in Fig. 2.2.1.1. There are four key elements in the packets:

- Arbitration ID
- Data Length Code
- Data
- CRC Field

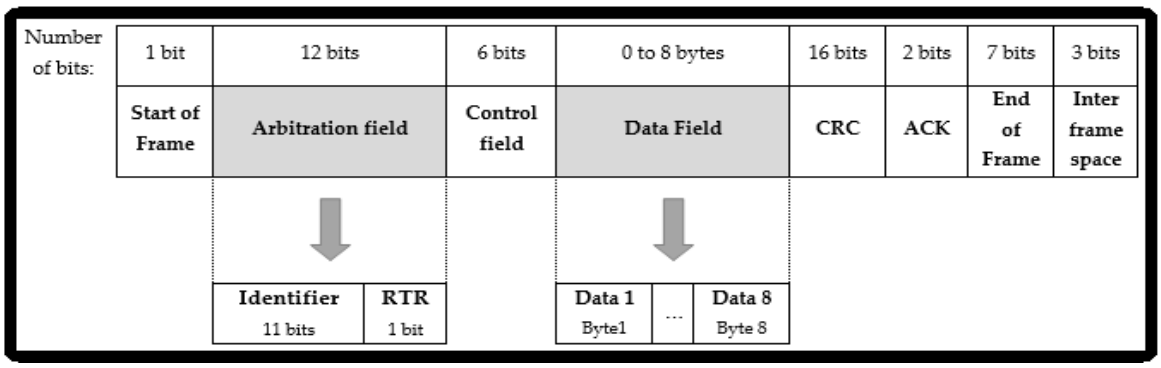


Figure 2.2.1.1. CAN Data Frame [10]

As stated earlier, in CAN protocol there is no field for sender and the receiver of the message, given the broadcast-based topology. Each node is capable of transmitting and receiving over the bus at the physical layer. The messages are identified based on the Arbitration IDs. This can be 11-bit field of 29-bit field if CAN extended is used. This field plays an important role in the arbitration in case two or more nodes happen to transmit the messages at the same time. In that case the messages with the lower ID will be transmitted i.e. the lower the ID the higher the priority is. We will explain the arbitration mechanism in coming paragraphs.

DLC field contains the information about the data payload length. This is the 3-bit field which means the maximum of 8 bytes can be sent in the standard CAN protocol in one packet. However, in the extended version, up to 64 bytes can be transmitted per packet.

Data field is a variable length and contains the payload to be transmitted. The length is determined by DLC field. For Standard CAN protocol this is limited to eight bytes while for extended CAN this can take up to 64 bytes.

For detection of propagation errors, CRC checksum is included in the packet. Upon receiving, the node re-computes the CRC checksum and compares it with value present in the packet. If the value

matches, the receive reports back to the transmitter by setting ACK bits. In case of an error, the error frame is generated which makes all the other nodes on the Bus to discard the received packet.

CRC is strictly not related to the security of the packet. An attacker can update the contents of the packet without the detection or can re-compute the CRC field. This is entirely different from the secure Message Authentication Codes (MACs).

2.2.2. Arbitration:

Whenever two or more nodes on the same bus try to transmit at the same time contention takes place. As stated earlier, the message ID with lower value gets the bus and gets transmitted. This is handled by CAN transmitter automatically. To do so, the CAN transceiver compares the value being transmitted and the value on the bus. The value on the bus is ANDED bit wise with the transmitted value. So, when zero is transmitted and the value on the bus is 1, the result of AND is zero and this way zero is transmitted resulting in the lower value node winning the arbitration. The node which sends one and see zero on the bus, stops transmitting.

The scenario is explained in Figure 2.2.2.1 where Node 3 transmits due to lower ID.

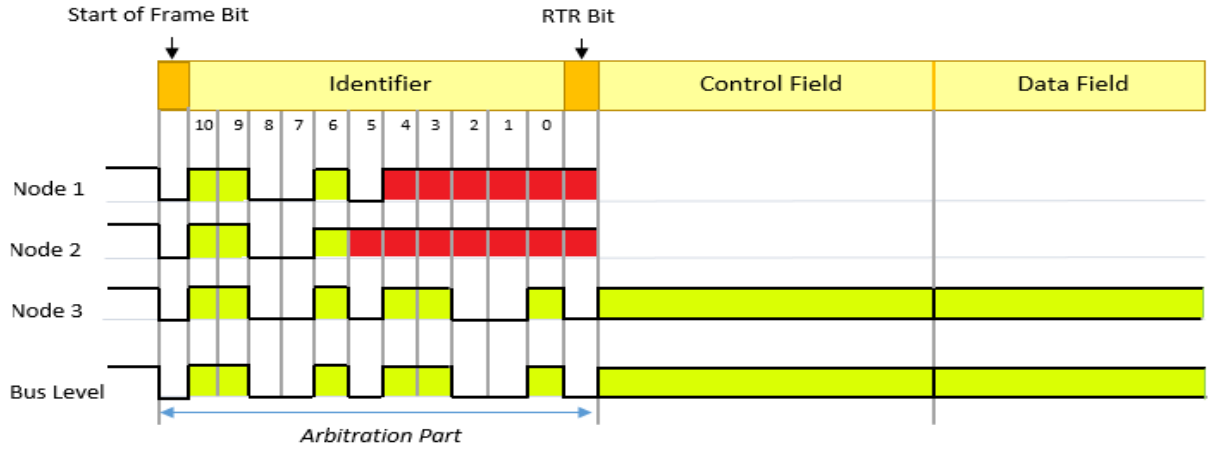


Figure 2.2.2.1. Arbitration Example

This scheme ensures that the messages with high priority gets transmitted first. It can be exploited to launch Denial of Service attack on the bus if the bus is grounded so that all the time, the bus is in the dominant state.

2.2.3. Bit stuffing:

As CAN is an asynchronous network, so, the Clock is not included in the signal. Though the synchronization is maintained with the help of bit transitions, but a sufficient number of transitions per packet are still required to synchronize all the nodes on the bus. To avoid, the timing errors, at the physical layer, after every consecutive five same bits an opposite bit is stuffed in the frame to keep track of the time. This is handled by the CAN transceivers automatically and no programming is required at application level.

2.2.4. Remote Frame (RTR):

This is a special frame which is used when a node needs more data from the sender. The architecture is essentially the same as the data frame, except that no data is transmitted in a remote frame itself. The RTR bit is set in the arbitration which indicates the Remote frame.

2.2.5. Error Frame:

This packet indicates the error in the traffic. These errors can be

- Bit Error
- Bit Stuffing Error
- CRC Error
- Format Error
- Acknowledgement Error

Whenever any of the above errors are detected by a node, it will start transmitting the error frame. The error states of every node is also maintained locally by two counters known as Receive Error Counter (REC) and Transmit Error Counter (TEC). Based on the error state (Error Active/ Error Passive), error flag is raised which can be 6-12 dominant/recessive bits. Then the error delimiter is indicated by 8 consecutive recessive bits.

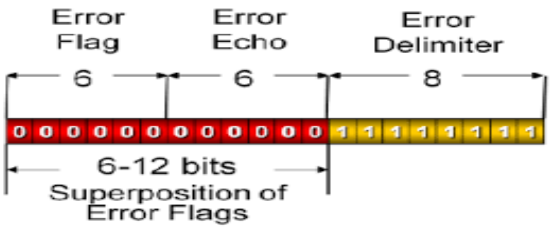


Figure 2.2.5.1. CAN Error Frame

2.2.6. CAN Overload Frame:

This frame is rarely used in the CAN-Bus traffic. It is transmitted by the receiver node when the receiver needs delay between the next frames from the sender. It is commonly due to the internal condition, where the receiver is not able to process the traffic or if there is a bit detected in intermission.

2.2.7. Upper layer protocols:

Automotive industry has defined certain protocols on top of CAN which makes it easier to implement the multilayer networking architecture. For example, a node can be assigned a particular address which can be used as sender and receiver addresses. Hence making it easier for the developers to develop the applications without the details of lower layers. Such protocols are ISO-TP (automotive diagnostic protocol), SAE J1939 (Transport protocol for heavy vehicles) and Unified Diagnostic Services- ISO 14229 etc.

Similarly, in Linux Socket-CAN is a utility which can be used to interact with the CAN devices using the Linux kernel networking approach. While using it, the CAN protocol is treated as any other networking protocol and the high-level details are handled by Linux, while the lower layers are implemented through the hardware drivers.

2.2.CAN-Bus Vulnerabilities:

As mentioned previously, CAN protocol was designed for the robust and reliable communication by replacing the wire jumble. At the time of the creation of this protocol, the security of the communication was ignored hence, the primary weaknesses in terms of security are still present and easily exploitable [11].

Confidentiality:

In CAN protocol, the confidentiality of the communication, by design is absent. Though through pre-shared keys, the CAN traffic can be encrypted but the lower payload size (8-bytes) makes it futile. However, the confidentiality is not a necessary security requirement for CAN traffic even in today's time. The primary reason is that CAN does not carry the sensitive information and require physical access.

Integrity:

Cyclic Redundancy Check (CRC) field is present but that is to prevent the propagation errors not the intentional tampering of the data. The attacker can forge the data and re-computes the CRC or bypass through the mathematical manipulation.

Authenticity:

Because of the bus topology, any node in the network can transmit the message. Moreover, the sender and recipient addresses are missing fields in the protocol, which makes it wide open to the spoofing and phishing attacks. Given the physical access, an attacker can transmit the messages to critical nodes without any authenticity [12].

Though the problems of Integrity and Authorization are solved with the help of Message Authentication Codes (MAC), but the small size of the payload in CAN makes it challenging. For providing security to 8-bytes data, at least 16-bytes are needed for MAC which makes it expensive solution.

We are going to discuss in detail how these weaknesses can be exploited remotely as well as the proposed solution to solve these problems in CAN.

CHAPTER 3: Remote Attack Surfaces

Modern vehicles contain many wireless interfaces, enabling attackers access and hence exploit the various features remotely e.g. LTE networks, Wi-Fi, Bluetooth, Infra-Red may be, Key fobs. In this chapter we will be discussing the ECUs which receives their input from external world. As a rule of thumb, the more the input surfaces are there, the more attack vectors can be harnessed in order to compromise the vehicle.

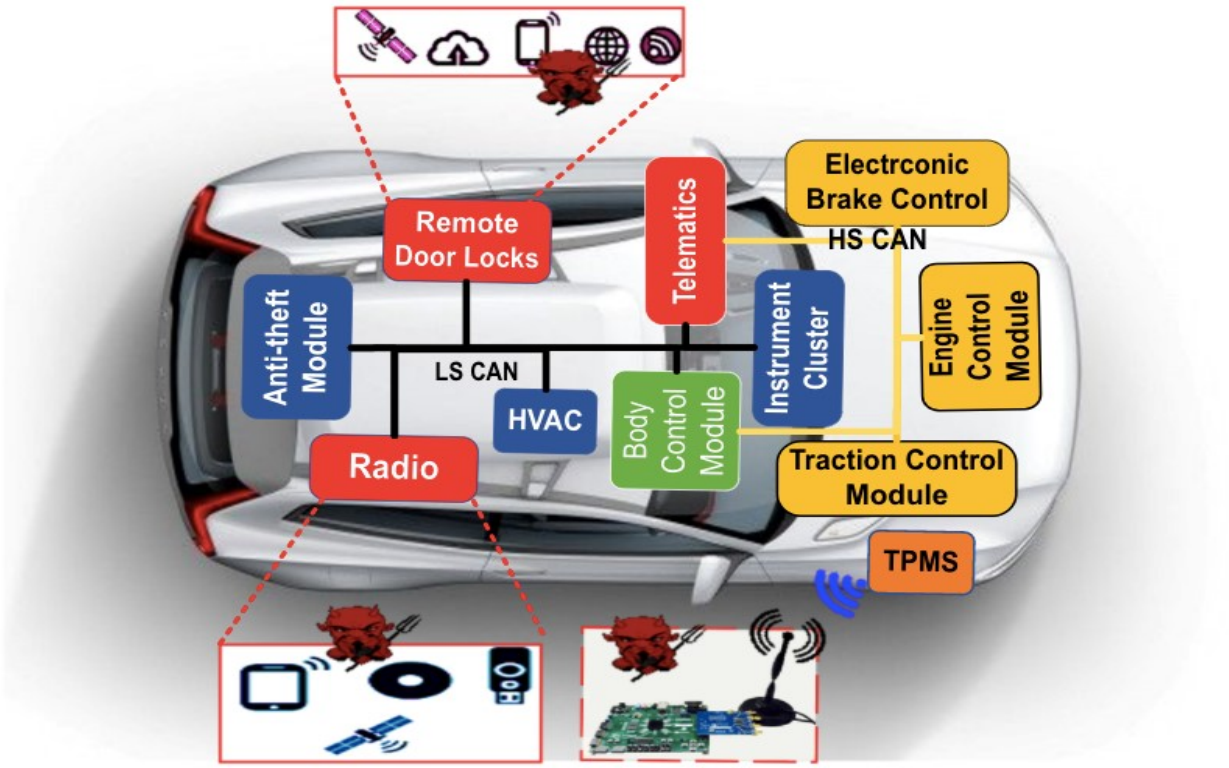


Figure 3.0.1 Remote Attack Surfaces in a modern vehicle

For example, Telematics unit is the main component when it comes to the communication with internet. It is equipped with the 3G/LTE networks modem, Wi-Fi, Bluetooth and this enables the vehicle to communicate directly to the cloud services and many more external entities. If such an ECU is compromised due to some vulnerability per se a vulnerable host running on the Wi-Fi as a result of which an attacker is able to compromise the whole ECU and then can infiltrate into the internal networks of the vehicle, exploiting the local vulnerabilities, leading to breaking down the vehicle completely.

3.1. Remote Attack Significance:

Remote attacks are the attacks which does not require the physical access to the vehicle itself and the attacker is able to interact with the vehicle wirelessly through the wireless interfaces. For example, an attacker can manipulate the Bluetooth stack running in the embedded Linux on Infotainment System and gains access to the sensitive data in infotainment system by standing couple of feet away from the vehicle.

Remote attacks may not damage the vehicle directly, but they can be leveraged by the attacker with the ability to communicate with the modules which are not directly accessible and only communicate internally. In the real-world setting, a common approach is the network segregation based on the functionalities of the ECUs. For example, the safety critical ECUs cannot get commands directly from the ECUs connected externally, in this case the attacker have to compromise into the isolated network as well by reverse engineering the network topology of the vehicle.

This situation becomes intense when the ECUs which act as gateway, run the complex Operating Systems which are prone to more vulnerabilities due to high code density. In experience, we have

observed that such ECUs run QNX, Embedded Linux, Android OS, Windows and now Apple Car. All of these operating Systems are vulnerable at some point and needs to be updated continuously. This opens another discussion which is Secure Over the Air (SOTA) firmware updates.

We may include these features in the new manufactured vehicles, but this still leaves the old vehicles insecure where such security functionalities are not implemented at all or are implemented poorly. If the firmware of the ECUs cannot be updated over the air, then a zero-day critical vulnerability can turn the situation into recall class issue. An example is the famous Jeep Cherokee hack which made Chrysler to call-back about one and a half million vehicles [4]. We will be discussing the overview of that attack shortly in this chapter. Briefly, at the time of attack discovery every Jeep connected over the internet was able to ping and port scanned leading to the vulnerability exploitation [13].

We can conclude the above discussion by classifying the remote attacks into two major categories based on the end goal of the attacker.

If the attacker wants to eavesdrop the communication going inside the vehicle or stealing the sensitive data, then the attack vector is limited to the In-vehicle Infotainment System or Telematics Unit only.

If attacker wants to damage the vehicle safety critical functionalities by having the capability to execute the code remotely or by manipulating the safety critical features. In this chapter we will be discussing some of these attacks as well.

3.2. Remote Attack Surfaces:

Sticking to the remote attack surfaces below are some of the wireless interfaces which can be exploited by the attackers to gain unauthorized access to the internal components of the vehicle which are hidden and use these to launch the further complex attacks [14].

- Passive Anti-Theft System (PATS)
- Tire- Pressure Monitoring system
- Remote Keyless Entry / Start (RKE)
- Bluetooth
- Radio Data System
- Telematics/Cellular/ Wi-Fi
- Internet/ Apps

3.2.1 Passive Anti-Theft System (PATS):

This is a feature design to prevent the vehicle theft and enables the car to respond to the correct ignition key. This prevents the car from being ignited by the arbitrary ignition-keys and is found in most of the modern vehicles in slight variations. The main working principle is given below:

There is a very small chip inside the key, which transmits a unique RF ID to a sensor in the steering column which is attached to the Instrument Cluster. When the car ignition is started, the RF signals should be received by the chip or else the car will not be ignited and will lock certain important features.

This is a very small attack surface and runs a very small amount of code. The only possible attack vector which is known at the time that, if exploited it can be used to steal the car. One more thing the range of the RF signals is pretty short in this case and the attacker must be close to the key in

order to capture the packet and the start manipulating the data which makes it infeasible and unattractive attack surface.

3.2.2 Tire Pressure Monitoring System (TPMS):

The vehicles are monitoring the tire pressures in the real-time in order to detect the defects and low-pressure situation. This is being done with the help of very small sensors which are embedded inside the tires and are constantly transmitting the data to an ECU. This ECU is attached to the Smart Junction Box in Ford vehicles.

The protocols being used here are proprietary, however there are some studies being done which shows that this can be abused easily [15]. Using the Software Defined Radios (SDRs), the vehicles can be tricked into thinking that there is a problem with the tire and hence stopping the vehicle.

If installed the RF transmitters in the proper locations on the roads and highways, the vehicles can be made to stop immediately causing the crashes. Again, the attack surface is small and the TPMS sensors are not directly attached to the internal networks of the vehicles. Though it provides a small but precise attack surface which can be exploited to launch a Denial of Service kind of attacks.

Figure 3.2.2.1 Tire Pressure Monitoring Sensor by Continental

3.2.3 Remote Keyless Entry/ Start (RKE):

RKE is the functionality which is used in the vehicles since mid 90s and used to lock/ unlock and start the vehicles remotely. This technology is a kind of similar to the garage door openers and is relatively a more abused attack surface.

The principle is that the transponder in the key transmits an encrypted signal, proprietary usually, to the immobilizer in the vehicle at the low frequency. One of the reasons for the low frequency is limit the battery usage on the immobilizer side. A code or authentication token is transmitted at the low frequency which authorizes the transponder to the immobilizer and hence the vehicle is locked or unlocked.

In the modern systems, to prevent the replay attack the rolling counters are implemented and the immobilizer is asked to perform certain computations which lead to the successful authentication. We can perceive it the inferior public key cryptography type of algorithms as well.

This technology can be attacked in the following ways which lead to Denial of Service attacks as well as unauthorized lock/ unlock requests at the same time as well [16].

3.2.3.1 Jamming the Key Fob Signal:

Just like other radio frequency signals the passband of the receiver can be filled with the garbage by the attacker using SDR. When the owner of the vehicle presses the button and a counter value or the packet from the transponder is transmitted that does not reach the vehicle's internal ECU, but attacker can capture the signal over SDR. Then the owner will re-press the button and another packet will be transmitted which as well will be captured by the attacker. Now attacker can re-transmit those recorded signals one will lock the vehicle while the other will unlock the vehicle.

This is the description of an attack launched by Samy Kamkar at Defcon 23 and is successful on the garage openers as well [17].

3.2.3.2 Pulling Response Codes from Immobilizer memory:

This attack vector deals with the immobilizer memory. A common error in security applications, the sensitive data is not flushed, the memory of the immobilizer can be scanned and if the response codes are found over there, it can easily be exploited to start the vehicle without capturing the live key fob RF signals.

3.2.3.3 Forward Prediction Attack:

As the transponders might not be equipped with the cryptographically secure random number generators but instead the pseudo random number generators (PRNGs) which get seeded by the time in most of the cases at the power cycle.

An attacker can seed the PRNG on his machine and seed with the same time, this can be leveraged to generate the stream of random number generators exactly same as the key fob and hence the communication can be hijacked easily as the attacker can predict the responses in advance.

3.2.3.4 Dictionary Attacks:

This attack may be infeasible for real time case in terms of time taken. The attacker observes and records the challenge/response pairs and start building the dictionary and then waits for the case where the challenge is repeated and can unlock the car based on the response known ahead. But this is somewhat unlikely given the challenge space.

3.2.4 Bluetooth:

Bluetooth is one of the exciting wireless feature which enables the users to almost control the phone from IVI. While driving, the Bluetooth enables the users to dial calls, see contacts and the other respective options like voice texting etc. So, it is an exciting feature which attracts the attacker community.

Bluetooth is an important attack surface given its range and the size of the respective software stack. The big software stack running on the IVI may contain the significant amount of vulnerabilities. Bluetooth devices can be attacked in below mentioned ways:

- Un-Paired Phone
- Paired Phone
- Traffic Sniffing

The attacks involving the unpaired phones are more dangerous and may depend upon the vendor of the victim which may vary the implementation of the security features.

If attacker has a paired phone, it can extract the sufficient information. There could be a possibility about the phishing attack, but the pre-shared secrets disable such kind of attacks.

Bluetooth protocol employs the encryption schemes providing sufficient level of security as well. But there is a possibility of side channel attacks to recover the plain texts and breach into the in-vehicle infotainment systems.

3.2.5 Radio Data System:

Other than the audio signals, the radio have many other remote inputs possible as well. For example, in the Ford Escape, the module is used to receive the GPS signals and the satellite radio data as well. The data here is parsed carefully and relatively a mature surface.

One possibility is the Radio Data System Data which is used to convey some additional information which is displayed as the song title and the name of the radio system.

This is relatively hard to attack with the lower possibility of the vulnerabilities.

3.2.6 Telematics/Cellular:

This is one of the most exciting feature of the vehicles and the modern vehicles rely on this module a lot to provide the features. This module makes it possible for the vehicle to communicate over the internet providing the cellular connection to the vehicle as well. Another use of this module is to provide the internet connectivity over the Wi-Fi hotspot. The other state of the art features being provided are the emergency calling, the cop calling and the vehicle tracking in case of loss. The commercial vehicles are also tracked based on this feature.

This was one of the vulnerable features which were exploited by the security researchers to compromise the Jeep over the internet anywhere [13].

3.2.6.1 Fake Cell Tower:

A simple attack is to set up a Femto cell and get the vehicle communicating to your set up cell tower. This has been proven to be successful. Once the attacker gets the fake cell tower talking with the vehicle, it can be used to launch various attacks.

Port Scanning: One of the most exciting thing which has been observed in experience is that the port-scan becomes possible and the attacker will be able to fingerprint the services. Hence, even if some a vulnerable service is running, it can be exploited easily using the open source vulnerability scanning tools.

Firmware Updates: Most of the OEMs are updating the firmware over the internet through the internet connection provided by the telematics unit. This opens two possibilities here.

The firmware can be downloaded and the extracted by the attacker in the processes of the firmware upgrading. This can be subjected to the binary analysis hence, it can be reverse engineered. The vulnerabilities can be discovered and hence exploited although this require the great determination by the attacker. Still it is an open attack vector.

The other possibility is that the attacker is able to download the custom firmware and update the ECU's himself. But this is unlikely if there is some kind of the secure integrity checks are employed already.

3.2.7 Wi-Fi:

Wi-Fi is also an easy attack vector. The security of the Wi-Fi is WPA2-Personal based on the pre-shared keys mostly. The users may leave the Wi-Fi password unchanged which is usually set at factory and may be printed on the physical dongle or somewhere.

Like any other embedded application, the scenario is aggravated when there is the same password set for every Wi-Fi device or is based on some physical property like MAC addresses. Once the attacker is able to get into the Wi-Fi, he will be able to fingerprint and launch the numerous attacks, both on the ECUs, firmware and can even redirect the traffic by DNS poisoning and the redirecting the traffic to malicious servers.

Though the telematics compromise may not give attacker the direct access to the safety critical ECUs, but it can be used as a proxy to the other inner ECUs in the vehicles.

There is an interesting situation when there are some of the open ports without the authentication for some services. In experience, it has been observed that for the debugging purposes there are some of the services which run and accessible over the Wi-Fi and can be abused to get the privileged access over the Wi-Fi. Presence of debug accounts over WiFi is a common practice as well. When the authentication is weak the attacker may become luckier and may be able to have elevated privileges to install the custom applications, run code and access to the sensitive logs.

Briefly the telematics unit is a honey pot for the attackers and there are numerous existing vulnerabilities which can be exploited remotely enabling the attacker to execute the code as well as retrieve the sensitive information.

3.2.8 Internet/ Apps:

The internet browsing capability on the IVI is a large attack surface which can be easily exploited. The situation is quite complex when the attacker has the capability to direct the internet traffic due to the network exploits.

For example, due to fake cell tower or the Wi-Fi, the attacker may be able to poison the DNS and hence launch the vulnerabilities. This also opens the window for the browser-based attacks. If the user visits some malicious website, there can be a compromise.

3.3. Features Abuse:

Some advanced features can be abused to get the undesired results. These are not essentially the exploits rather the services abuse which may create the scenario of the denial of service attack and deteriorate the quality of the service which in turns increase the risk factors for the users who allow on these features [14].

The attacker may be able to produce a message on the internal network to the safety critical ECUs to take some sort of action like applying brakes or turning the wheels. Presence of these features equips the attacker with the possibility of abusing hence resulting in some physical changes in the vehicle.

When the attacker has the ability to communicate with such ECUs, the safety of the passengers becomes risky. Such features are given below which can be abused by the attacker once he is just able to communicate. The details of these features are out of scope for this paper, but these could be the important part of any attacker's post exploitation activities [14].

- Park Assist
- Adaptive Cruise Control
- Collision Prevention System
- Lane Keep Assist

Most of these features are provided by the dedicated ECUs which listen to the certain signals originated from the sensors and a result will adjust the speed, the wheel angles and may apply the emergency brakes. Though direct communication to these ECUs is restricted but once there is a breach to other ECUs and attacker gain the access enough to communicate to the critical ECUs, then the safety of the passengers is at great risk without any doubt.

In the next chapter we will provide the existing methodologies which are employed or proposed by the research community to secure the internal networks. As mentioned earlier, CAN is the backbone of the in-vehicle networks, we will limit the discussion to CAN only.

CHAPTER 4: Literature Review

We have discussed the primary weaknesses in the CAN protocol so far. In this chapter we will be overviewing the related work being done by the research community to address the problems.

Wang et al [18], proposed an idea based on the Message Authentication Codes (MAC) using the symmetric crypto on SHA-3 calculated over the message. A message, termed as authentication message, has to be sent by the message senders to authenticate the sent messages. To avoid the overhead, the ECUs are labelled as high trust group and the low trust group. The low trust group are the ECUs which communicate with the external world like OBD-II, telematics etc. To communicate with High Trust ECUs, a shared secret must be known by the sender otherwise the high trust group will discard the message. One of the biggest challenge is the digest calculation overhead. It seems suitable to employ the hardware crypto devices to provide the sufficient security. But it becomes expensive in that case. If the crypto is applied directly in the ECU, then the side channel attacks become more viable. Similarly, the key management becomes a problem as well. Not all the ECUs are designed to run the cryptographic operations securely. The architecture is that eight bytes of authentication is sent, which is relatively a narrow space and there are more chances of collision here, though finding collision is hard. The approach itself to send a separate message is flawed and is susceptible to the strong denial of service attack scenario.

Nilsson, Larson and Johnson [20] proposed the similar solution but their approach was to distribute the whole CBC-MAC value in the different packets in the CRC field. This approach is even more

susceptible to the denial of service situation where all the attacker has to do is to just tamper one message of the whole stream. It is a complex solution and the security becomes challenge in the high velocity applications.

Herwege et al. [19] proposed a modified version of the CAN protocol known as the CAN+. They proposed to hide the authentication bits in the payload based on the MAC computation. Again, the cryptographic solution has certain limitations in the embedded world which has been discussed previously. Another problem with adopting this approach is that it is not a backward compatible solution. CAN will probably be replaced in near future with the better protocols to enhance the speed limitations and provide better security. The approach adopted in CAN+ could be utilized over there but for the existing protocol this approach does not seem to be viable and adopted by the industry.

Murvay and Groza [21] has proposed an approach to physically fingerprint the devices based on the unique characteristics hidden in the physical electrical signals. The approach they have utilized is the statistical modelling of the devices based on the fingerprints. We are essentially following this idea, but we have proposed the robust and easily computable parameters. This work needs to be investigated under the harsh conditions of the vehicular environment where the electromagnetic interference is a challenge.

Another interesting property for the in-vehicle communication which is exploited is the periodicity of the messages. Muter and ASAJ [22] has proposed an Intrusion Detection System based on this property of the CAN network traffic. Their proposed IDS regulate the period of the certain IDs in CAN traffic and then measure the entropy of the contents, which in result is used for the intrusion detection. Similar proposal was made by the Jeep Cherokee hackers Charlie Miller and Chris Valasek, where they proposed an IDS which detects the anomaly in the message period distribution

[13]. Other than the frequency of the messages, the message contents verification is also proposed. Though these proposals, somehow provides the certain level of security against the simple attacks but these fail to provide the security against the attackers who can control the frequency. Moreover, it is possible to transmit the malicious payload containing CAN messages at the same frequency as of the legitimate messages so bypassing such IDS is relatively easier.

[6] One of the state of the art solutions is based on the unique characteristics of source clock of the sender node. The clocks in digital devices are excited by the crystals. Just like the electronic devices, the crystals show some asymmetry in the structure which is unique and is not replicable. In asynchronous networks, the clock information is not embedded in the signal itself and hence, it is entirely dependent upon the source of the clock which is sender. To fingerprint the devices in networks such as TCP/IP based on the sender clock's uniqueness have been proposed by the researchers previously. Most of the network protocols get the time information about the source clock in the packet header and hence from there it can be used to extract the unique characteristics. However, in the simpler embedded networks, the timing information is not present in the clock hence it had been considered as a big challenge to adapt the such techniques for such networks like CAN.

Authors in [6] proposed a method to estimate these characteristics in CAN network for the periodic messages. In most of the automotive CAN networks, a particular message ID is being sent by the particular ECU, so an ECU's clock behavior can be tracked based on the message IDs it transmits. So, the receiver/ monitoring node, models the source clock behavior based on the particular message ID and the received messages for that particular IDs are used for tracking the behavior. If the estimated parameters of the source clock deviate from the model, it indicates that the source

of the messages has been changed. In the next chapter we are going to analyze this particular IDS in detail and will be proposing a successful attack to bypass the IDS.

CHAPTER 5: Attacking Clock Based Intrusion Detection System

As the gist of the Clock based Intrusion Detection System (CIDS), the clock parameters estimated at the receiver or monitoring node are used to authenticate the sender of the message. It is based on the fact that the clock crystals possess the uniqueness in the behavior due to the asymmetric density of the material. This uniqueness is estimated based on the statistical parameters which are used to construct a mathematical model of the sender's clock and when a deviation from the model is observed, it indicates the source of the messages is being spoofed.

Before detailing the weaknesses of this approach, we will be discussing the working architecture of the Clock based Intrusion Detection system.

5.1 Clock Parameters:

Below are the parameters which are used for the fingerprinting in this technique:

5.1.1 Clock Offset:

The difference between the reported time of the subjected clock and the reference clock is called the clock offset, *O*. Mostly the reference clock is the clock of the monitoring node or can be the receiving node.

5.1.2 Clock skew:

The rate of change of the offset with respect to the reference clock is known as the clock skew, *S*. It arrives due to the difference in the frequencies of the clock.

5.1.3 Clock frequency:

The clock frequency is the rate of the reference clock.

In an asynchronous network, the clock offset, and the clock skewness is entirely dependent upon the message sender's clock. If we are able to estimate these parameters accurately, we can uniquely fingerprint the source clock.

5.2 CIDS working Overview:

The working of CIDS consists of two major stages as below:

1. Clock behavior modelling
2. Anomaly (Attack) Detection

5.2.1 Clock Behavior Modelling:

This stage is further divided into sub-stages.

Offset Calculation:

Let's assume that a message with particular ID is being transmitted periodically at the period of T from the sender node. The overall timing information to reach the receiver from sender can be broken down as below:

- Propagation delay d.
- Random noise in the measurement attributed as n.
- Offset between the sender and the receiver's clock O_i

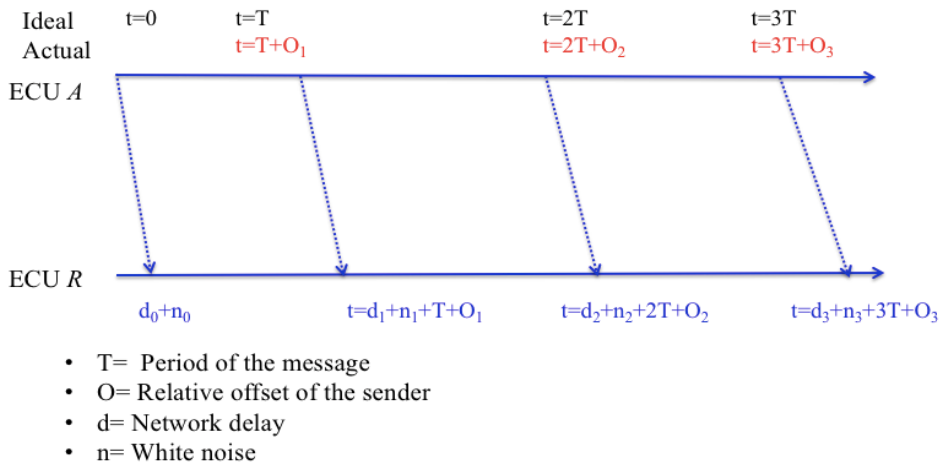


Figure 5.2.1.1 Detailed Break Down of the Time from Sender to Receiver

If the first message is transmitted at *t=0*, the subsequent *i* messages are received at the receiver side *i* messages will be received at t_i according to the below equation.

$$t_i = O_i + n_i + d + i * T \qquad \text{(Eq. 5.1)}$$

For all of the messages, the propagation delay is going to be same as well as the average noise value can also be ignored.

In case of the ideal clock situation i.e. if there is no offset and skew difference, Eq. 5.1 is reduced to Eq. 5.2:

$$t_{ideal} = O_i + n_i + d + i * T \qquad \text{(Eq. 5.2)}$$

Which is strictly the time, t_{ideal}, the messages of the given ID are expected.

But given the factor *d* and the n_i are going to have a uniform impact over all of the messages' timings and can be ignored as well.

Now if we subtract the above two equations, we will be able to compute the offset, which is simply the difference between the expected time of the i^{th} message and the actually receiving time of the message.

$$O_i = t_i - t_{ideal} \quad \text{(Eq. 5.3)}$$

Skew Calculation:

As the skew of the subjected clock is essentially the rate of the change of the offsets but the difference between two subsequent offset values is negligible and very hard to calculate tit precisely. The authors proposed that the absolute average offset values for *N* subsequent messages can be summed together known as the cumulative sum, O_{sum}. It is important to note that the difference between subsequent values of offsets can be negligible but the average value of the offsets for N messages cannot be zero.

The cumulative sum grows linearly with time; hence its slope will yield the skew of the source clock which is supposed to be unique in an asynchronous network. As long as the source of the messages is same, the constructed behavior follows the linearity but whenever there is a change in the source of the clock it results in the sudden jump, indicating the abnormality in the clock behavior which is the change of the source clock.

The equation for the cumulative sum, for the k steps (each step consists of N messages), can be formulated as the linear line equation:

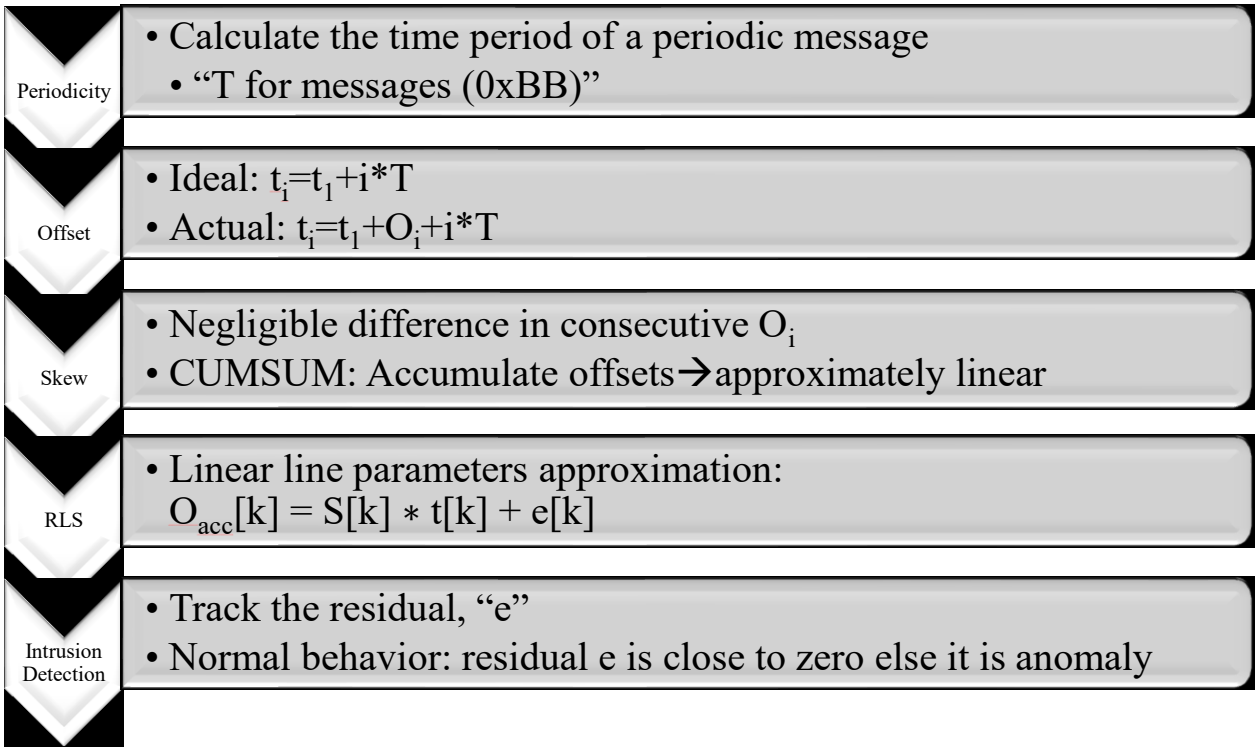
$$O\_sum\ [k] = S[k-1] * t + e[k] \quad \text{(Eq. 5.4)}$$

Where the slope S of the cumulative sum is computed using the Recursive Least Square Algorithm (RLS).

5.2.2 Detection:

As from the equation Eq. 5.4, the residual error e indicates that the deviation of the received message clock from the constructed behavior. If the message is from the actual source then this value is close to zero while in case of the deviation, which is the change of the cock source, the residual value shoots indicating the abnormality in the clock behavior.

This is the gist of the clock-based intrusion detection system for CAN. The threshold of the residual errors is statistically computed and can be configured as the sensitivity level of the instruction detection.



Flow Chart. 5.1. CIDS Working

5.3 Attack Models:

1. **Suspension Attack:** This is defined as a kind of the denial of service attack. Let us assume that the attacker has compromised an ECU A which is sending the messages with ID 0xAA. The attacker is able to stop the transmission of the messages 0xAA leading to a denial of service attack.

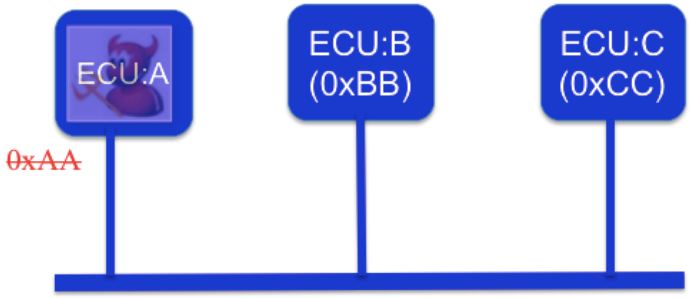


Figure 5.3.1 Description of Suspension Attack

CIDS is able to detect this attack Figure 5.3.2. As we can see that at the time of the attack, there is a sudden jump in the clock behavior, showing the anomaly in the network.

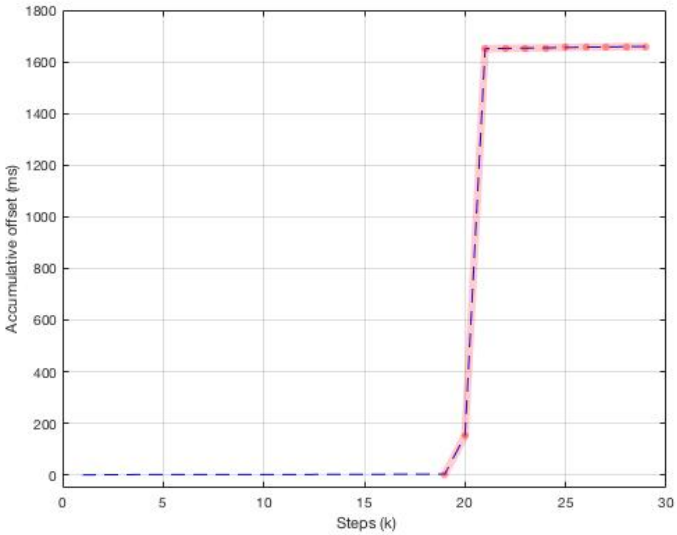


Figure 5.3.2 Suspension Attack Detection at k=19 for message with ID=0xAA (Period= 50ms)

2. **Fabrication Attack:** This is a kind of spoofing attack. If the attacker has compromised a node A in such a way that he has the ability to transmit the messages with arbitrary IDs. In the system legitimate messages with ID 0xBB originates from the ECU B. Given the attacker's ability to transmit the arbitrary messages he will start injecting the fabricated message stream containing the malicious content.

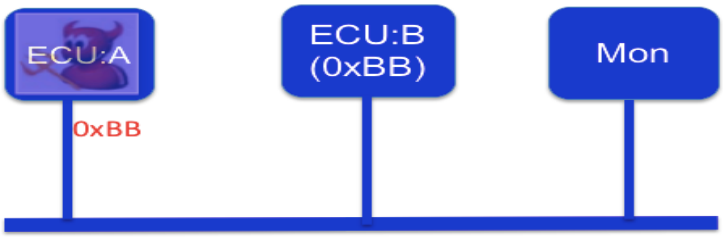


Figure 5.3.3 Description of Fabrication Attack

This kind of attack is successfully detected by the CIDS as well with high accuracy. As we can see in the Figure 5.3.3, there is a sudden jump at the point when there were malicious messages were injected into the network by the attacker.

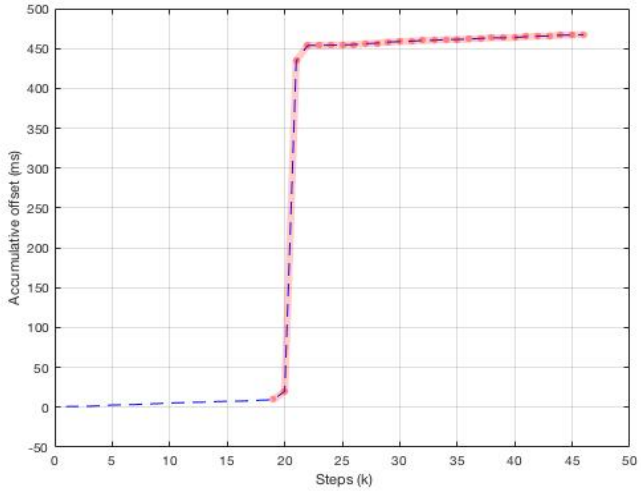


Figure 5.3.4 Fabrication Attack Detection at k=19 for message with ID=0xBB

3. **Masquerades Attack:** This is a hybrid attack of both of the previously mentioned attacks. The attacker has compromised two ECUs in the system in such a way that from the ECU A, he can launch the suspension attack while from the ECU B he can launch the fabrication attack. When both attacks are launched simultaneously, i.e. attacker stops the messages 0xBB from ECU B while start sending the fabricated messages 0xBB' from ECU B at the same rate even, he has launched the Masquerades attack.

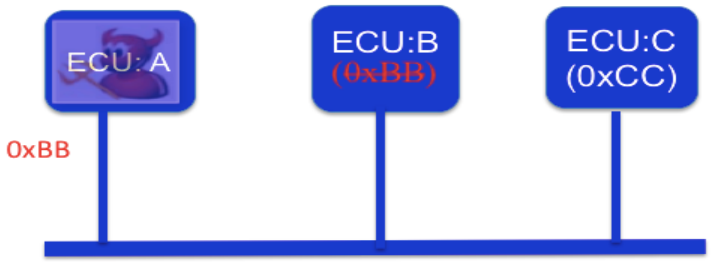


Figure 5.3.5 Description of Masquerades Attack

This is an advanced attack which is usually not detectable by most of the state-of-the-art Intrusion Detection Systems. However, CIDS is able to detect the change of origin for the messages which are fabricated by the attacker with the fine control even.

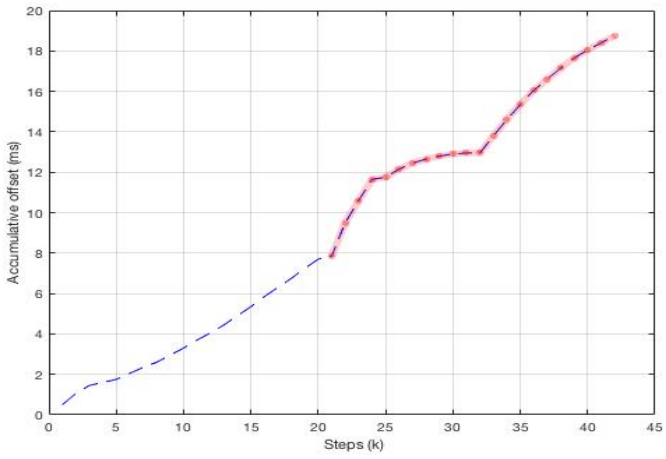


Figure 5.3.6 Masquerades Attack Detection at k=19 for message with ID=0xBB

5.4 Weaknesses in CIDS:

As we have seen that though CIDS is successful in detecting the advanced attacks and compromises in the system, we have found the potential weaknesses in the CIDS which lie in the core of the architecture of the technique and can easily be exploited by the attacker to attack the system. We have practically shown this vulnerability and exploited it.

There are two potential vulnerabilities in the system which can be exploited:

5.4.1 Estimated parameters' dependence on Time Period:

The parameters are calculated based on the difference of the arrival time and the expected time of the periodic messages for the given ID. Ideally, for a given node, the parameters should be independent of the message period. If a node transmits two messages with ID 0xAA and 0xAB with time period T1 and T2, the estimated clock behavior should be same at the monitoring node because the clock source is same.

But it has been observed that this does not hold true. We have observed that when the messages of different periods are transmitted from the same node, the constructed behavior of the clock at the monitoring node, differs a lot Figure 5.4.1.1.

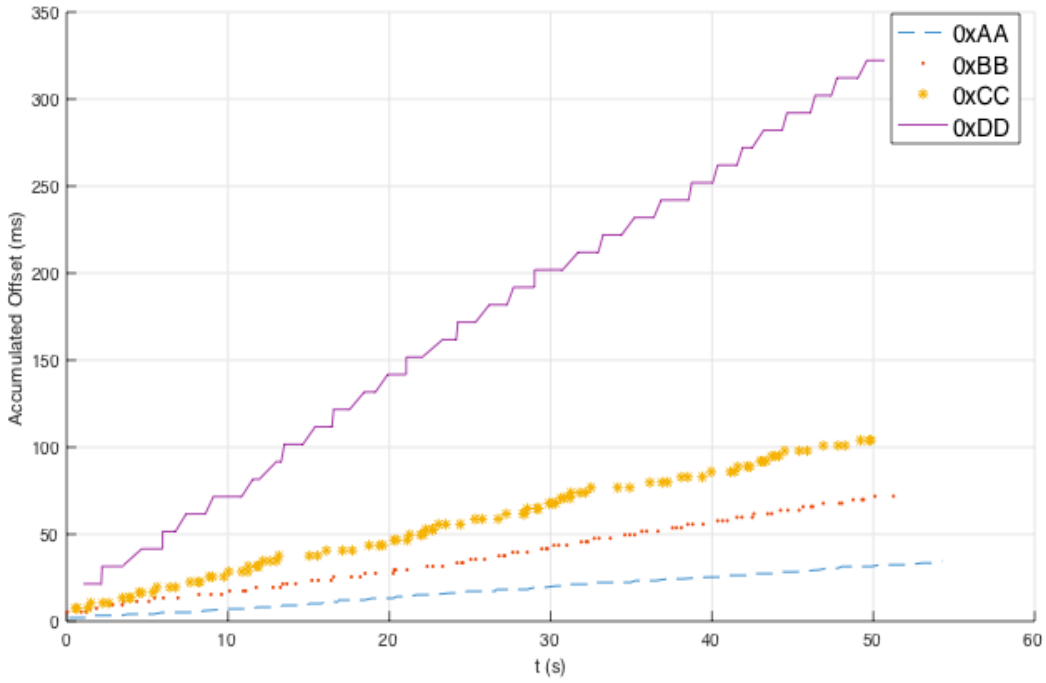


Figure 5.4.1.1 Dependence of clock behavior on period of message. (0xAA: 50ms, 0xBB:100ms, 0xCC:150ms and 0xDD:500ms)

This opens a window for the attacker to adjust the period of the transmitted malicious messages such that it starts depicting the clock behavior as of the original messages. But here the challenge is that how much the message periods should be adjusted to produce the desired clock behavior at receiver side.

5.4.2 Non-linearity of the clock behavior:

The underlying assumption in the design of Clock based Intrusion Detection System is that the clock behavior constructed by the cumulative sum is strictly linear and the deviation is computed from the normal behavior using the residual value. The high residual value indicates the potential deviation of the clock behavior from the intended clock behavior. For the normal operation, the residual is tracked and should be within the certain range. If the residual is computed to be out of the normal range, it is implied that there is an attack.

This property gives the attacker a margin of error and a challenge that, exploiting the dependence of parameters on the period of the messages, can be exploited in such a way that the resulting clock behavior on the side of monitoring node lies within that range.

5.5 Exploitation of the vulnerabilities- Clock Spoofing:

If both of the above vulnerabilities in the CIDS are exploited carefully, the attacker is able to transmit the malicious messages in such a fashion to bypass the CIDS. For simplifying the discussion, we assume that there are three nodes in the network:

Node L: The legitimate node which sends the messages 0xAA

Node M: The monitoring node which monitors the clock behaviors in the system.

Node A: The attacker node which will transmit the malicious messages 0xAA.

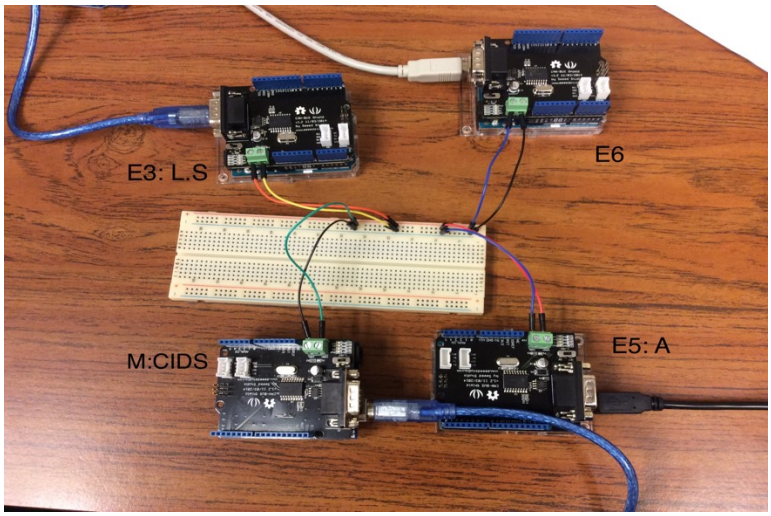


Figure 5.5.1. Attack Setting

The node M, fingerprints the node L and computes the skew and offset values denoted as S_{LM} and O_{LM}. These parameters indicate the legitimate sender's clock difference with respect to the monitoring node.

Let's assume that the attacker is listening to the messages on the compromised node and starts fingerprinting the clock behavior of the victim node L as well. We call these parameters as S_{LA} and O_{LA}. These parameter values represent the differences of the clocks between the victim node and the attacker node. With this knowledge of the difference, the attacker can compute the absolute amount of change needed in the time period for the messages to be transmitted, so that on the monitoring node, the calculated parameters on the malicious messages indicate the clock behavior of the node L.

5.6 Launching Attack:

5.6.1 Attacker Side:

In our implementation, we are running a local copy of the CIDS on the attacker node and the attacker is able to listen to the victim messages, while constructing the clock behavior as mentioned. He computes the parameters S_{LA} and O_{LA}.

The offset value indicates the difference of the reported time between the legitimate sender node L and the attacker node A. So, if the attacker changes the time period of the malicious messages to be transmitted by this amount, it will be similar to the clock of the legitimate sender.

The skew S_{LA} indicates the rate of the change of the offset value w.r.t the attacker, so the attacker can predict or update the offset O_{LA} for the malicious messages to be transmitted.

Mathematically the situation can be represented as:

$$T_{iA} = T_{LA} + O_{LA} * (1 + S_{LA})^{i} \qquad \text{(Eq. 5.5)}$$

Where

T_{iA} : The time period of the i^{th} malicious message transmitted by node A

T_{LA}: The relative Time period of the legitimate messages observed at node A

O_{LA}: Relative offset of the legitimate sender w.r.t to node A

S_{LA}: Relative skew of the legitimate sender w.r.t to node A

5.6.2 Monitoring Node Side:

When the messages with adjusted period i.e. malicious messages reach the monitoring node, the clock behavior is calculated based on the parameters O_{AM}' and S_{AM}' which are the offsets observed on the adjusted values of the time period from node A.

$$O_{AM}' = O_{AM} + O_{LA} = t_A - t_M + t_L - t_A = t_L - t_M = O_{LM}$$

As per the definition of offset, it is the difference of the reported time and the reference time. The notation is explained as below

$O_{LA} = t_L - t_A$: the relative offset of the legitimate sender's clock is the difference of the reported time by the legitimate clock (t_L) and reported time at that moment by the node A's clock (t_A).

Similarly,

$O_{LM} = t_L - t_M$: the relative offset of the legitimate sender's clock is the difference of the reported time by the legitimate clock (t_L) and reported time at that moment by the node M's clock (t_M).

O_{AM}' is the value of the offset computed at the monitoring node based on the malicious messages which has been proven to be same as the O_{LM} that's the value computed in the case of the legitimate messages transmitted by the node L and observed at monitoring node M. Based on these values the residual which is computed on the O_{AM}' will be e' and will remain in the normal behavior range as original e.

The O_{AM}' is updated by an attacker based on the skew rate S_{LA} which will keep the future message offset values to behave as of the legitimate node.

In the Figure 5.6.2, the attack is depicted and successfully bypassed, we can see that there is no update sudden jump or change in the clock behavior although the subjected messages are under attack.

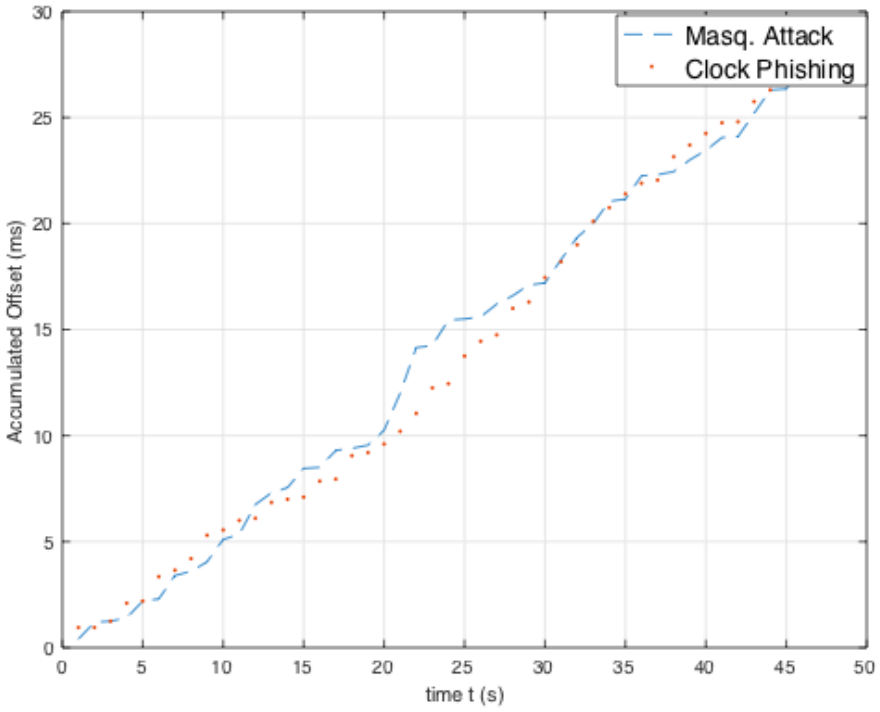


Figure 5.6.2. Clock Phishing Attack at t=5 s on message 0xAA (Period: 50ms)

5.7 Conclusion:

As we have shown the weaknesses in the CIDS as well as exploited them to spoof the clock of the sender. As the weaknesses exist in the architecture of the technique, so it is not possible to defend the clock spoofing attack. We may have to derive the high order parameters or try to learn the unique behavior based on the machine learning techniques.

In the next sections, we will be discussing the techniques which we have proposed in order to defend and authenticate the senders at transmitter's inimitable physical characteristics estimated from the raw signals at physical layer.

CHAPTER 6: Physical Unclonable Functions (PUFs)

Electronic devices are powered by the silicon chips and it is obvious that no matter how symmetrical the architecture of the material is, at atomic level, no two given volumes of the material can have exactly same number of atoms and same arrangement. In other words, there is always an atomic level asymmetry between any given two devices. That asymmetry is unique and if it is identifiable of estimated correctly, it can lead to the identification/ authorization of the electronic device.

6.1 Concept:

This is the basic principle of a very powerful state-of-the-art security technique applied for the electronic devices with higher security requirements. PUFs depend upon the physical uniqueness of the microstructure of the device which is introduced during the manufacturing process randomly. Such randomly introduced factors are unpredictable and uncontrollable, which makes it impossible to replicate these devices.

The device authentication is established using the challenge-response based techniques. When a physical stimulus is applied to such device, the device response is unique based on the factors which have been introduced randomly at the manufacturing time.

Another way of using this property is to construct the cryptographic keys based on the unique artifacts which when computed results in the same value. Then the challenge-response mechanism can be realized to establish the device identity. This establishes the root of trust at hardware which

cannot be deceived or altered and replicated through the life of device and is proven to be one of the most secure designs.

6.2 Realizations:

Physical un-clonable functions can be realized through two basic ways:

- Extrinsic Randomness
- Intrinsic Randomness

6.2.1 Extrinsic Randomness:

In this technique the randomness is introduced explicitly but it is introduced in the random fashion. R. Pappu et al. proposed the idea of the physical one-way functions based on the optical properties which are introduced in the manufacturing process [23,24]. This technique involves the doping of the transparent material with the light scattering particles. When the laser interacts with the material, a unique and random speckle pattern will be produced due to the mutual interference of the reflected waveforms. This lays a solid foundation for the device authentication.

A similar technique based on different physical implementation is proposed by Skoric et al. Their proposed technique is based on the idea of coating the IC chips with the network of the metal wires. This is then filled with the randomly placed dielectric material which introduces the random capacitance and cannot be replicated or reproduced [25, 26, 27]. This is the idea which is used for the unique RFID tags creation.

6.2.2 Intrinsic Randomness:

The intrinsic randomness is more attractive area of the research as it does not involve the changes in the manufacturing process. It is based on the natural randomness present in the material.

One implementation relies on the randomness in the path traversed by the particles when a voltage (challenge) is applied. Different race conditions set up in the material in such situations. Two transitions are tracked to see which reaches first in the other end and a latch is implemented to produce “1” or “0”. When a circuit layout mask is implemented on different devices, the logic function is different because of the different delay values which result due to the randomness in the path. Such implementations are known as the Delay PUF. Suh et al [28] has shown that the multiplexer-based PUFs can be used in the secure processor designs.

Tehrenipoor et al [29,30] proposed the intrinsic un-clonable functions based on the DRAM. DRAM capacitors are initialized to a random value at startup and then this starts decaying to zero. Ideally all of the cells in a DRAMs are supposed to be identical but there are the above-mentioned imperfections which cause the difference and hence somewhat uniqueness. When there is a power cycle applied to a certain DRAM chip, all the cells are initialized, and they will start decaying to zero again with the time. The randomness that which cell will decay to zero is a window for the creation of a PUF. Similar approaches have been applied for the SRAM based technologies as well.

CHAPTER 7: Device Fingerprinting in Embedded Networks

In this paper, we are extending the idea of PUFs to the authentication and the fingerprinting of the devices in the embedded networks. Our proposed idea includes the two naturally random artifacts:

- Device's Natural Artifacts
- Channel's Natural Artifacts

Device's artifacts exist due to the material asymmetry in the device naturally. Similarly, channel artifacts exist naturally when the electrical signal traverses a given channel.

The challenge here is to uniquely estimate these artifacts and then use them to authenticate. As per our proposal, the randomness and the uniqueness which is present in the electrical device itself, can be estimated in the raw physical signal as well and when this signal traverses the path in the network, the network channel will impose its unique artifacts in the signal as well.

So, based on the artifacts which are computed from the signal, we can also link it to the path or the channel which it has traversed, which in terms can be used to identify the malicious of compromised paths in the strict small networks like CAN. We have shown in our studies that we can fingerprint the channels as well.

This provides the two levels of security or authentication:

- Device Authentication
- Path Authentication

Even if the attacker is able to create the malicious packets or somewhat defeat the device artifacts, it will be unlikely for him to spoof the channel.

7.1 Unique Artifact Estimation:

We are proposing different techniques to estimate the uniqueness:

- Statistical Signal Analysis
- Impulse Response

Once we can mathematically compute the parameters based on these techniques we can apply a machine learning based classifier to classify the signal and link the received signal to its source from the raw signal.

7.1.1 Statistical Signal Analysis:

In [21], the authors employed the similar signal analysis technique in order to fingerprint the CAN nodes. We are extending this approach further by reducing the number of parameters required along with the channel fingerprinting. We are able to identify the CAN Bus channels even when the same signals were transmitted by the same node but on different CAN Buses.

The uniqueness, infringed by the device physically, can be observed both in the time domain as well as the frequency domain. So, if we statistically analyze the received raw signal, we will be able to observe this uniqueness.

After the signal is sampled in time domain, it is converted to the frequency domain and then computed the parameters mentioned in Table 7.1.1 and 7.1.2.

Table 7.1.1. Time-domain feature set

Feature name	Equation

Maximum	$m_{ij} = (\text{Min}(y_{ij}(i)) \mid i=1\ldots N)$
Minimum	$M_{ij}=(\text{Max}(y_{ij}(i)) \mid i=1\ldots N)$
Mean	$\mu_{ij}= \frac{1}{N} \sum_{i=1}^{N} y_{ij}(i)$
Variance	$\sigma_{ij}^2=\sqrt{\frac{1}{N-1} \sum_{i=1}^{N} y_{ij}(i)-\mu_{ij}}$
Skewness	$\rho_{ij}= \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_{ij}(i)-\mu_{ij}}{\sigma_{ij}}\right)^3$
Kurtosis	$\kappa_{ij}= \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_{ij}(i)-\mu_{ij}}{\sigma_{ij}}\right)^4 -3$

Table 7.1.2. Frequency-domain feature set

Feature Name	**Equation**
Spectral Std-Dev	$\sigma_s=\sqrt{(\sum_{i=1}^{N} (y_f(i))^2*(y_m(i))) / \sum_{i=1}^{N} (y_m(i))}$
Spectral Skewness	$\rho_s= (\sum_{i=1}^{N} y_f(i)(y_m(i)) / \sigma_s^3$
Spectral Kurtosis	$\kappa_s=(\sum_{i=1}^{N} (y_m(i)-C_s)^4*y_m(i)) / \sigma_s^4-3$
Spectrum Centroid	$C_s=(\sum_{i=1}^{N} y_f(i)y_m(i)) /(\sum_{i=1}^{N} y_m(i))$
Irregularity-K	$IK_s=\sum_{i=2}^{N-1} \mid y_m(i) - \frac{y_m(i-1)+y_m(i)+y_m(i+1)}{3} \mid$

y_m and y_f are the magnitude and the frequency vectors

The computed feature vector seems sufficient to identify the devices as well as the traversed channel. The detailed experimental results have been shown in the next section.

7.1.2 Impulse Response:

If we treat our system as really an input and output system. Where the input to the system is the ideal waveform of the signal while the output is the signal which is distorted and contains the unique artifact. We can model the whole system easily.

Let's say the input digital signal is x(t) while corresponding the signal generated by the ECU i is s(t), then we can say the impulse response is h_i and the relationship is given as:

$$s_i(t) = h_i(t) * x(t) \quad \text{(Eq. 7.1)}$$

Because of the uniqueness of the electrical devices, the output signals are going to be unique at physical layer even with the same set of input signals, it indicates that the impulse responses of the systems are going to be unique essentially.

To compute the impulse response of the system we have converted the whole systems into frequency domain first.

$$S_i(w) = H_i(w).X(w) \quad \text{(Eq. 7.2)}$$

Where

$$S_i(w) = Fourier(s_i(t))$$

$$H_i(w) = Fourier(h_i(t))$$

$$X(w) = Fourier(x(t))$$

When $H_i(w)$ is known we can take the inverse Fourier and can compute the time domain impulse response.

The hurdle here is the computation of x(t). Because we are dealing with the digital signals here, we expect the signal to be high or low only and the voltage is defined for these thresholds at system

level. For example, we can say that one level is 3.5 Volts and the other is 1.7 volts. Then every sample in the output which corresponds to one level, we can get the corresponding input sample. E.g. for an output sample of 3.3v we can tell that its corresponding input sample should be at 3.5 V. This is how the input is constructed from output and hence the impulse response can be computed easily.

When we include the channel response as well to provide the second layer of authentication, the situation can be described easily as shown in Fig.

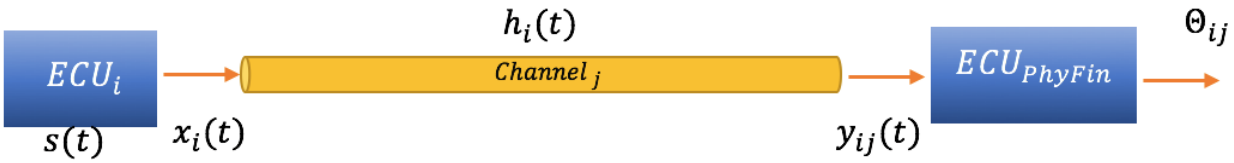


Figure 7.1.2.1 Physical input signal and channel response

Let S_i be the signal coming from the ECU i and this signal traverses the channel j. The channel j's impulse response w.r.t. ECU i is h_j and the signal received at the PhyFin node (Fingerprinting node) is y_{ij}.

$$y_{ij}(t) = h_j(t) * S_i(t) \qquad \text{(Eq. 7.3)}$$

In Figure 7.1.2.2 we can visualize that even though the signal source is same but the it is traversed through different CAN Bus channels, the difference in the physical signals is evident.

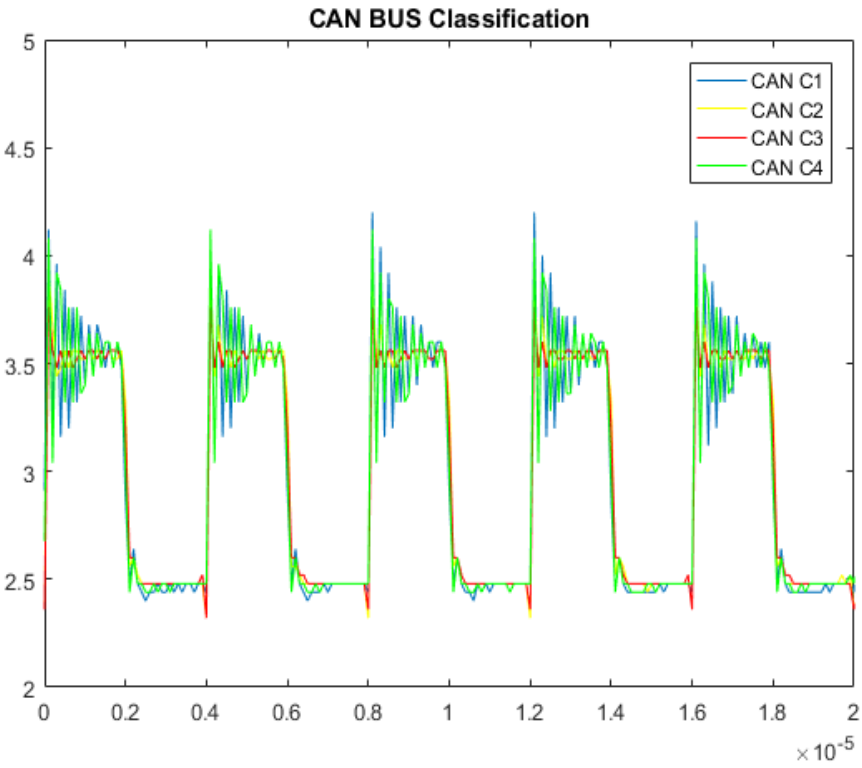


Figure 7.1.2.2 CAN Bus signals, when the signal is same but signal propagates through different CAN Bus channels.

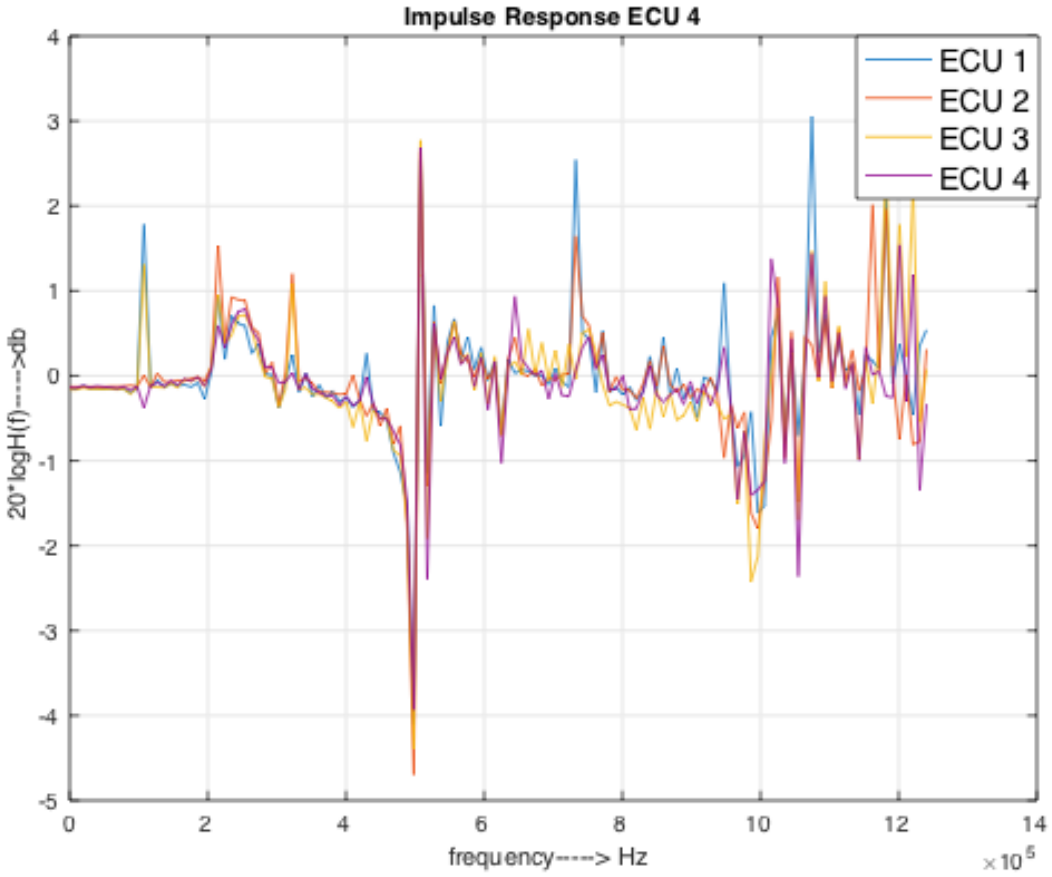


Figure 7.1.2.3 The impulse responses of 4 different ECUs transmitting the same signals.

An interesting thing has been observed while analyzing the impulse responses, is that there are only few components of the frequency which show the uniqueness so instead of using most of the spectrum for the fingerprinting, we can use a small number of the frequency components for this purpose.

We are going to discuss the implementation and our test beds where these techniques have been validated in a CAN network, in next chapter.

CHAPTER 8: Experimental Results and Validation

A number of experiments were conducted to implement and validate the proposed techniques. We analyzed the CAN signals at physical layer on CAN High as well as CAN Low.

Our testbed is based on the Arduinos and CAN shields as a proof of concept. Because we are providing the physical layer security, the automotive industry standard cables were used. The detailed setup is explained as below:

8.1 Experimental Setup:

8.1.1 ECU Emulation:

To emulate the CAN Traffic, we used the Arduino UNO R2 microcontroller kit which was interfaced with the MCP 2515 CAN-bus controller and MCP 2551 CAN transceiver. Arduino UNO is powered with Atmega328 with 20 digital I/Os. It is interfaced with the CAN-Bus Shield over SPI [31].

This combination is an effective simulation of an in-vehicle ECU with the ability to communicate over the CAN bus. This is equipped with the OBD-II interface as well which enables it to collect the data from the vehicles directly.

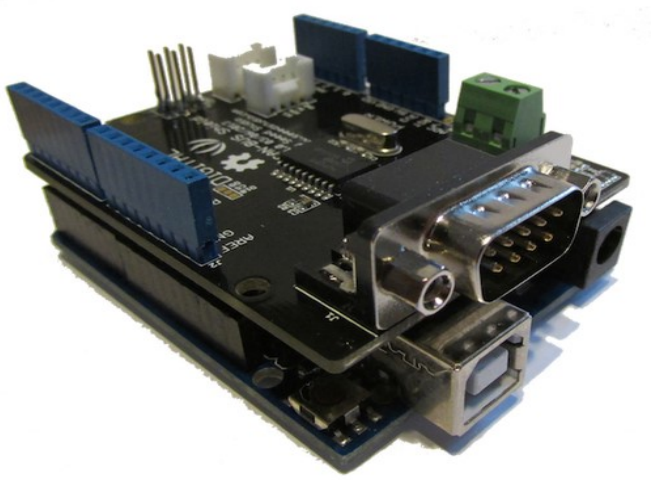


Figure. 8.1.1.1 Arduino Uno R2 interfaced with CAN Sheild

8.1.2 Channel Realization:

To construct the CAN Bus same as used in the real-world vehicle, we designed our experiments based on the three different types of the cables.

- **GXL:** Deployed in the engine compartment of the vehicles.
- **TXL:** Used in the applications requiring the light weight and small diameter
- **CAN-bus data cables SAE J1939-15:** Different ECUs connection

Different lengths of each of these cables (0.5m, 1m, 2m, 3m, 4m and 6m) were used.

8.1.3 Data Acquisition:

To acquire the raw voltage values off the CAN Bus directly, we needed the high speed and high precision ADC. To shortcut the time of development of such board, we used DSO 1002A oscilloscope which gives us a sampling rate up to 2GSa/second.

The ability to collect the data directly from Oscilloscope into the MATLAB workspace made it immediate and first choice for the activity of the data acquisition. We used the MATLAB

Instrument Control Tool Box to interface it over the USB port. We could control the resolution and the settings of the oscilloscope from MATLAB directly. To automate the process, we have created a MATLAB script. The oscilloscope reads the voltage values and then transmit to the MATLAB over a USB connection (VISA- Protocol).

The data is acquired over the CAN High wire with reference to the ground. The same can be done on the CAN Low as well.

Once the data is acquired it can be saved as the .csv files so that it can be re-used. For every experiment we created a root or parent directory which contains the numerous sub-directories. Each subdirectory contains the .csv files for each different ECU for the given experiment. This topology is very helpful in the data management and manipulation.

8.1.4 CAN Traffic:

As a proof of concept, the identical signals are transmitted. We selected the signals which produces the 1010101010 patterns on the physical layer so that, we can easily analyze the signals. This pattern includes the 1s and 0s equally and can easily help in the computation of impulse response as well. The pulse train like structure is widely used in the signal processing domain.

The CAN Bus speed is set to 500 Kbps for the experiments but can be configured easily. At this speed and the sampling rate of the oscilloscope, we get the 40 samples per pulse i.e. 1 and 0.

8.1.5 Neural Network Implementation:

To train and test, the MATLAB's neural network Tool Box is used [32]. This toolbox gives us the flexibility to customize the neural network architecture and the number of layers as well as the number of neurons in the network.

An automated script was developed to take the training data as input and train the classifier over it. Here the feature vector contained the features computed over the signal in the time as well as frequency domain. The size of the neural network depends upon the experiments. For the experiments with the large number of ECU nodes, we need to have the powerful networks requiring multiple layers and more number of neurons.

8.1.6 Feature Vector Preparation:

For given experiment, the raw voltage values are read from the csv files stored in the corresponding sub-directories. To make the dataset uniform in each cycle, the data is trimmed down and then the time domain and the frequency domain features are computed for each recorded cycle. The feature vector is constructed in this way.

Once the feature vectors are computed for the whole experiment, the dataset is subjected to the neural network training. The neural networks are trained using backpropagation algorithms. By hit and trial method the size of the neural network is adjusted to meet the best performance in the minimum size.

8.1.7 Testing:

Once the neural network is trained, it is saved in the form of a MATLAB function, which can be called over the computed feature vector. From the live traffic, the data is captured using the oscilloscope and once the data is in MATLAB's workspace, the features are computed on the fly. The feature vector is prepared and fed into the trained network. The network predicts the source ECU based on the confidence. If the confidence for the maximum class is below than a threshold we identify that as the hostile ECU detecting the spoofing attack.

To improve the performance of the system, we employ the majority voting scheme for the given number of samples. For the experiment we get 15 cycles of the CAN bus. Again, one cycle represents two bits1 and 0. The neural network predicts for 15 cycles and the maximum number of the predicted class is then considered to be the final prediction of the neural network. This diminishes the poorly trained network's errors as well.

8.2 Statistical Analysis:

For statistical analysis we performed two classifications. The channel classification and the ECU classification itself. The test matrices are given below.

8.2.1 ECU Fingerprinting:

- Number of ECUs: 4
- Feature Vector: Table 7.1.1 and Table 7.1.2
- Neural Network Architecture:

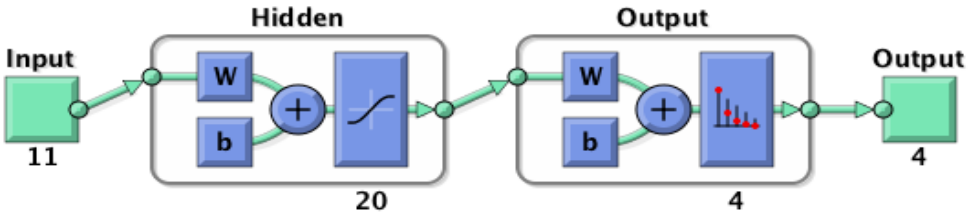


Figure. 8.2.1.1 ECU Classifier Neural Network Architecture

- Performance: In the tables below, 1 sample = 1 pulse cycle

Table 8.2.1.1, Training confusion matrix for ECU classifier

Predicted Class						
	E1	389 24.9%	0 0.0%	3 0.2%	0 0.0%	99.2% 0.8%
	E2	0 0.0%	398 25.5%	0 0.0%	0 0.0%	100% 0.0%
	E3	3 0.2%	0 0.0%	379 24.3%	0 0.0%	99.2% 0.8%
	E4	0 0.0%	0 0.0%	0 0.0%	398 24.9%	100% 0.0%
	Class label	99.2% 0.8%	100.% 0.0%	99.2% 0.8%	100% 0.0%	**99.6%** **0.4%**
		E1	E2	E3	E4	
		Target Class				

Table 8.2.1.2. Testing confusion matrix for ECU classifier

Predicted Class						
	E1	200 23.8%	0 0.0%	6 0.7%	0 0.0%	97.1% 2.9%
	E2	0 0.0%	202 24.0%	0 0.0%	0 0.0%	100% 0.0%
	E3	7 0.8%	0 0.0%	212 25.2%	0 0.0%	96.8% 3.2%
	E4	1 0.1%	0 0.0%	0 0.0%	212 25.2%	99.5% 0.5%
	Class label	96.2% 3.8%	100.% 0.0%	97.2% 2.8%	100% 0.0%	**98.3%** **1.7%**
		E1	E2	E3	E4	
		Target Class				

8.2.2 Channel Fingerprinting:

- Number of Channels: 6

- Feature Vector: Table 7.1.1 and Table 7.1.2
- Neural Network Architecture:

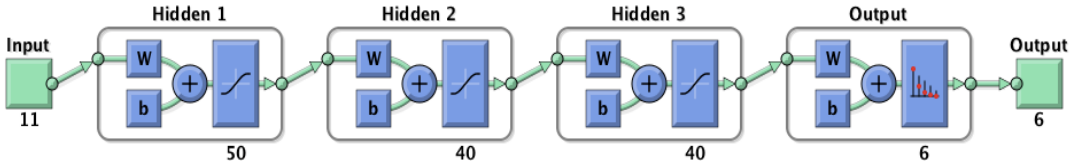


Figure 8.2.2.1: Channel Classifier Neural Network Architecture

- Performance: In the tables below, 1 sample = 1 pulse cycle

Table 8.2.2.1. Training confusion matrix for channel classifier

Predicted Class								
	C1	365 15.6%	4 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.9% 1.1%
	C2	30 1.3%	378 16.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92.6% 7.4%
	C3	2 0.1%	0 0.0%	376 16.1%	12 0.5%	0 0.0%	0 0.0%	96.4% 3.6%
	C4	1 0.0%	0 0.0%	8 0.3%	382 16.3%	0 0.0%	0 0.0%	97.7% 2.3%
	C5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	388 16.6%	0 0.0%	100% 0.0%
	C6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	394 16.8%	100% 0.0%
Class label		91.7% 8.3%	99.0% 1.0%	97.9% 2.1%	97.0% 3.0%	100% 0.0%	100% 0.0%	**97.6%** **2.4%**
		C1	C2	C3	C4	C5	C6	
		Target Class						

Table 8.2.2.2 Test confusion matrix for channel classifier

Predicted Class								
	C1	176 14.0%	10 0.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	94.6% 5.4%
	C2	22 1.7%	205 16.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.3% 9.7%
	C3	3 0.2%	3 0.2%	203 16.3%	9 0.7%	0 0.0%	0 0.0%	93.1% 6.9%
	C4	1 0.1%	0 0.0%	13 1.0%	197 15.6%	0 0.0%	0 0.0%	93.4% 6.6%
	C5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	212 16.8%	0 0.0%	100% 0.0%
	C6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	206 16.3%	100% 0.0%
Class Label		87.1% 12.9%	94.0% 6.0%	94.0% 6.0%	95.6% 4.4%	100% 0.0%	100% 0.0%	**95.2%** **4.8%**
		C1	C2	C3	C4	C5	C6	
		Target Class						

8.3 Impulse Response

The impulse response of the ECUs is computed using the Fourier Transform and for the proof of concept we only classified the ECUs and believe that the channels can be classified in the same manner.

8.3.1 ECU Fingerprinting:

- Number of ECUs: 4
- Feature Vector: Frequency components of impulse response {23:32,77:82} of 128-point Fourier Transform
- Neural Network:

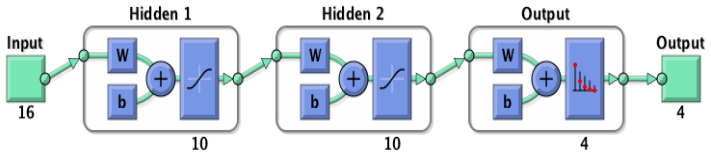


Figure 8.3.1.1: ECU Classifier Neural Network Architecture

- Performance:

In the following table, 1 sample corresponds to 1*40 cycles i.e. (signal: 10).

Table 8.3.1.1, Confusion matrix for ECU classifier

Predicted Class						
	E1	25 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	E2	0 0.0%	25 0.0%	0 0.0%	0 0.0%	100% 0.0%
	E3	0 0.0%	0 0.0%	25 0.0%	1 1.0%	96.2% 3.8%
	E4	0 0.0%	0 0.0%	0 0.0%	24 24.0%	100% 0.0%
	Class label	100.% 0.0%	100.% 0.0%	99.2% 0.8%	100% 0.0%	**99.0%** **1.0%**
		E1	E2	E3	E4	
		Target Class				

CHAPTER 9: Discussion

9.1 Real-World Implementation:

To incorporate this solution into a vehicle, our proposed techniques require the high resolution and high sampling ADCs. We may or may not need the hardware changes because the vehicle now measures the voltage level on the bus using existing ADCs.

If an ADC is connected to the CAN Bus, we can use that node as monitoring node, which monitors the traffic on bus. In training phase, the message IDs and the corresponding physical signatures are classified. Once the training is done, we can use that node to monitor the node.

For the new vehicles, we have the liberty to include the ADCs in different ECUs. The ideal location for the inclusion of ADCs would be the ECUs acting as the bridge between different sub-networks or work as the gateways. This makes the design simpler and the security is achieved at many levels through distributed mechanisms. Each central node is responsible for the security of the traffic for the respective sub network. It may include the non- CAN networks as well. Like Ethernet and Flex-Ray etc.

9.1.1 Performance Impact:

The performance impact in terms of the real-time implementation is minimal, because once a neural network is trained, it is not an expensive function and may be accelerated using the

hardware. The training time is however a big challenge and depends that how many ECUs are connected to the given subnet.

9.1.2 Possible Attack Vector:

As the input to our authentication system is the raw physical signal, the attacker can recreate such signal only and only if attacker is given access to the DAC with high resolution. Let's assume the attacker has compromised an ECU B which is equipped with DAC as well and he is able to receive the traffic of ECU A as well. Now the attacker wants to replay the traffic and just use the DAC pins (connected to CAN, which is unlikely but for the attack model we assume it happens) to replay the traffic. Because the ECU A and ECU B channels will be different w.r.t the monitoring gateway M, the channel artifacts prohibit this spoofing attack.

9.2 Challenges and Limitations

The experiments which we have performed and discussed are based on the lab environment and we even have faced the difficulties there when the lab environments are changed from one place to another place.

Specially the automotive environment is quite hostile in terms of the background noise and which is the next challenge for these techniques. The below factors will play an important role to drive the research in the right direction for implementation and testing in the real-world vehicles:

- Temperature
- Aging
- EM Noise
- Power fluctuations

As the proposed methods depend on the physical layer and the rely on the micro-architecture of the material. This structure depends on various factors and subject to change with the changes in the environment. Temperature is one of the factors, which impact the quality of the features. If the temperature is subjected to change on large scale, the performance of the proposed system will be impacted. The automotive environment is harsh and usually a vehicle experiences the changes in the temperature externally as well as internally. So, this plays an important role and yet the performance needs to be tested on this factor as well.

Aging is another factor which changes the structure of the material with time. The impact of the aging is also needs to be evaluated. On average a vehicle has a life of up to 10 years and the aging impacts the structure for this time of period. So, there are two solutions, that either compute the temperature resistant features or include the impact of the temperature on the features.

Similarly, the noise environment of the vehicle is a challenge itself. The noise can tamper the quality of the signals so, the signal processing techniques to remove the noise can be employed. But the noise removal may result in change of the characteristics and the error limit will give attacker a window to play.

The power is not constant in the vehicle and is subjected to change over the range of operations. While braking or accelerating the current flow is different than the vehicle at rest. All these factors pose a big challenge to our solution and needs to be taken into account.

9.3 Future Direction

Our work can be applied to broad range of the applications. In the paper, we presented CAN network as a Proof of concept. Ideally, this can be used to link the transmitted electrical signal to its source. So, it is equally applicable in the field of Digital forensics. Our work can be applied to

solve the problem of authentication and forgery without the cryptographic applications and provide the security at upper level in theory. The challenges which are concerning have been discussed above.

The above-mentioned challenges need to be addressed and is deciding the direction of the research on the topic. Another big milestone is the realization of the plug-in device which can be used in the real vehicle to provide the authentication.

The proposed ideas are not tested in the real vehicles yet so, another important goal is to validate the ideas on the vehicle.

Apart from that, we are looking for new features which are independent of the environment. Our future direction also includes the validation of these ideas on other embedded networks. We can extend these ideas to the Digital device fingerprinting. We want to show that even on the same board different pins have different characteristics and even the same signal is transmitted through them, we can identify the source form the physical Signal itself.

9.4 Conclusion

In this paper, we have provided an overview of the cyber threats being faced today by the Internet of Things world whose foundation lay on the shoulders on embedded systems. We discussed automotive applications as a use case and the discussed work is relevant to the threats for other IoT applications equally. We discussed the major attack vectors which can be used to bypass the security mechanisms for the modern vehicles.

We highlighted the basic flaws of the proposed research solutions and even attacked an IDS successfully, which is based on the inimitable characteristics of the clock crystal in the source. We were able to launch the successful spoofing attack against the CIDS.

We proposed a solution based on the Physical Unclonable Functions (PUFs) for the device authentication. We proposed the signal-processing based approaches to address the problem. Our proposed solution harnesses the power of machine learning and can authenticate the channel from which the signal is received even.

We proposed the two-layer physical authentication approach, which makes it impossible for attacker to launch the spoofing attack on the simpler network like CAN without using any cryptographic approach. The attacker would need to replicate the node as well as the channel of the signal. Neural Network based solution is faster once trained.

We also discussed the possible challenges and attack vectors on our approach. We are concluding this work on the positive note that this work contains the novel idea and step towards the security through unconventional approaches like signal processing and hardware root of trust.

Bibliography

[1] Checkoway, Stephen, et al. "Comprehensive Experimental Analyses of Automotive Attack Surfaces." USENIX Security Symposium. 2011.

[2] CAN-Bus Specifications Rep. Robert Bosch GmbH. Postfach 50, D-7000. Stuttgart 1Print.

[3] Ivan Studnia, Vincent Nicomette, Eric Alata, Yves Deswarte, Mohamed Kaaniche, et al. Survey on security threats and protection mechanisms in embedded automotive networks. 2nd Workshop on Open Resilient Human-aware Cyber-Physical Systems (WORCS-2013).

[4] Greenberg, A.: Hackers Remotely Kill a Jeep on the Highway (2015), http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[5] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy,D., Kantor, B., Anderson, D., Shacham, H. "Experimental Security Analysis of a Modern Automobile". In: 31st IEEE Symposium on Security & Privacy (S &P 2010), on. pp. 447:462

[6] Cho, Kyong-Tak, and Kang G. Shin. "Fingerprinting electronic control units for vehicle intrusion detection." 25th USENIX Security Symposium (USENIX Security 16). USENIX Association, 2016.

[7] https://www.pinterest.com/pin/43558321375378402/

[8] Ran, P., Wang, B., & Wang, W. (2008, April). The design of communication convertor based on CAN bus. In Industrial Technology, 2008. ICIT 2008. IEEE International Conference on (pp. 1-5). IEEE.

[9] K. Zdeněk and S. Jiří, "Simulation of CAN bus physical layer using SPICE," 2013 International Conference on Applied Electronics, Pilsen, 2013, pp. 1-4.

[10] http://doc.ingeniamc.com/emcl2/command-referencemanual/communications/interfaces/can-interface

[11] Tindell, K., & Burns, A. (1994, September). Guaranteeing message latencies on control area network (CAN). In Proceedings of the 1st International CAN Conference. Citeseer.

[12] Nilsson, D. K., Larson, U. E., Picasso, F., & Jonsson, E. (2009). A first simulation of attacks in the automotive network communications protocol flexray. In Proceedings of the

International Workshop on Computational Intelligence in Security for Information Systems CISIS'08 (pp. 84-91). Springer, Berlin, Heidelberg.

[13] Miller, C. & Valasek, C. "Remote Exploitation of an Unaltered Passenger Vehicle"

[14] Miller, C. & Valasek, C. "A Survey of Remote Attack Surfaces"

[15] Wen, V. "Security on Tire Pressure Monitor System"

[16] Smith,. C "The Car Hacker's Handbook" 2016.

[17] Thompson, Cadie (2015-08-06). "A hacker made a $30 gadget that can unlock many cars that have keyless entry". Tech Insider. Retrieved 2015-08-11.

[18] Q. Wang and S. Sawhney, "VeCure: A practical security framework to protect the CAN bus of vehicles," *2014 International Conference on the Internet of Things (IOT)*, Cambridge, MA, 2014, pp.13-18.doi: 10.1109/IOT.2014.7030108

[19] A. V an Herrewege, D. Singelee, and I. V erbauwhede, CANAuth a simple, backward compatible broadcast authentication protocol for CAN bus. in 9th Embedded Security in Cars Conf., 2011.

[20] NILSSON, D., LARSON, D., AND JONSSON, E. Efficient In- Vehicle Delayed Data Authentication Based on Compound Mes- sage Authentication Codes. In VTC-Fall (2008).

[21] Pal-Stefan Murvay and Bogdan Groza Source Identification Using Signal Characteristics in Controller Area Networks. IEEE SIGNAL PROCESS- ING LETTERS, VOL. 21, NO. 4, APRIL 2014

[22] MUTER, M., AND ASAJ, N. Entropy-based anomaly detection for in-vehicle networks. IEEE IVS (2011).

[23] R. Pappu, "Physical One-Way Functions", PhD Thesis, MIT, 2001.

[24] Pappu, R.; Recht, B.; Taylor, J.; Gershenfeld, N. (2002). "Physical One-Way functions". Science. 297 (5589): 2026–2030.

[25] Skoric, B.; Maubach, S.; Kevenaar, T.; Tuyls, P. (2006). "Information-theoretic analysis of capacitive physical unclonable functions". J. Appl. Phys. 100 (2): 024902

[26] B. Skoric, G.-J. Schrijen, W. Ophey, R. Wolters, N. Verhaegh, and J. van Geloven. Experimental hardware for coating PUFs and optical PUFs. In P. Tuyls, B. Skoric, and T. Kevenaar, editors, Security with Noisy Data – On Private Biometrics, Secure Key Storage and Anti-Counterfeiting, pages 255-268. Springer London, 2008

[27] Pim Tuyls, Geert-Jan Schrijen, Boris Skoric, Jan van Geloven, Nynke Verhaegh and Rob Wolters: "Read-proof hardware from protective coatings", CHES 2006

[28] Suh, G. E.; O'Donnell, C. W.; Devadas, S. (2007). "Aegis: A Single-Chip secure processor". IEEE Design and Test of Computers. 24 (6): 570–580

[29] Tehranipoor, F., Karimian, N., Xiao, K., & Chandy, J., "DRAM based intrinsic physical unclonable functions for system level security", In Proceedings of the 25th edition on Great Lakes Symposium on VLSI, (pp. 15-20). ACM, 2015

[30] Tehranipoor, F., Karimian, N., Yan, W., & Chandy, J. A. "DRAM-Based Intrinsic Physically Unclonable Functions for System-Level Security and Authentication". IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2017

[31] Badamasi, Y. A. (2014, September). The working principle of an Arduino. In Electronics, Computer and Computation (ICECCO), 2014 11th International Conference on (pp. 1-4). IEEE.

[32] https://www.mathworks.com/products/neural-network.html