Dissertation Titled

**EXPLOITING THE CAN BUS**

Submitted By

Rushabh Patel 131070005

Siddhant Shah 131070010

Abhijit Gupta 131070014

Mehul Jain 131070051

Under the guidance of Dr. S.G. Bhirud

****

Department of Computer Engineering and Information Technology

Veermata Jijabai Technological Institute

Mumbai-400019

(An Autonomous Institute affiliated to University of Mumbai)

2016-17

**ACKNOWLEDGEMENT**

I would like to thank all those people whose support and cooperation has been an invaluable asset during the stage II of this Project. I would also like to thank our Guide Dr. S.G.Bhirud for guiding me throughout this project and giving it the present shape. It would have been impossible to complete the project without their support, valuable suggestions, criticism, encouragement and guidance.

I am also grateful for all other teaching and non-teaching staff members of the Computer Engineering and Information Technology Department for directly or indirectly helping us for the completion of this stage of project and the resources provided.

**STATEMENT OF CANDIDATE**

I state that work embodied in this Project titled “EXPLOITING THE CAN BUS” forms my own contribution of work under the guidance of Dr. S.G.Bhirud at the Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute. The report reflects the work done during the period of candidature but may include related preliminary material provided that it has not contributed to an award of previous degree. No part of this work has been used by me for the requirement of another degree except where explicitly stated in the body of the text and the attached statement.

**CERTIFICATE**

This is to certify that Group GC05, students of B.Tech Computers,Veermata Jijabai Technological Institute(VJTI), Mumbai have successfully completed project titled “EXPLOITING THE CAN BUS” under the guidance of Dr. S.G.Bhirud.

Dr. S.G.Bhirud

(Guide) (Examiner - 1)

(Examiner - 2) (Examiner - 3)

**ABSTRACT**

The evolution of electronics in the previous century has changed the nature of vehicles dramatically. Nowadays, the control of different systems within a vehicle is carried out using tens of electronic control units (ECUs). These ECUs are clustered into networks having gateways in-between. Several standards are used for the communication within these networks and between them. The most widely used standard is the Controller Area Network (CAN) bus standard. In general, the design of such networks has been always concerned with reliability and safety. There was no much attention paid to the security of such networks. This is because there was no clear evidence if the security of such networks could be compromised. Practical experiments have demonstrated practical attacks on different systems inside vehicles.This included the ECUs controlling engine, brakes, lighting, climate control lighting and body controller. As a result, it is possible for an adversary to take-over the control of a vehicle and harm the passengers. This highlights a major risk for all cars; whether they have already been sold or they are still under development.

There are two types of security vulnerabilities that make these attacks possible. The first type is due to the inherent weaknesses of the used communication standards themselves. The second type is due to the deviation from security standards. The goal of this project is to investigate different methods by which the security of in-vehicle communication networks can be improved. The proposed methods consist of different levels of security enhancements ranging from prevention, protection and detection. The project focuses on achieving protection of the CAN bus against adversaries.We would like to introduce message source authentication protocol for the CAN bus. This can be achieved by designing a new lightweight authentication protocol. The main aspect of the protocol will be that it will be simple and practical so that it can be adopted directly in the automotive industry. The protocol does not need any extra hardware modifications and thus can be deployed inside cars that have already been sold. For intrusion detection of malicious attack in the CAN bus, we researched and found the Deep Learning solution using Deep Belief Network which promises to detect the malicious CAN packet better than frequency-based encoding method which is used for a packet feature in Artificial Neural Network and Support Vector Machine.

**TABLE OF CONTENTS**

[**1. Introduction**](#_vkwuzag13gdx)

1.1 Purpose…………………………………………..………................................ 10

1.2 Scope………………………………………………………………………………... 10

1.3 Motivation……………………………………………………………………………. 11

1.4 Problem Statement………………………………………………………………..... 11

**2. Literature Review**

[2.1 EVITA Project](#_q8v6e9ienl3i)……………………………………………………………………….. 13

[2.2 Message Authentication Protocol over CAN](#_dmphhc3m6b0w)……………………………………. 15

[2.3 Multiple MAC Per Receiver](#_wu7bgt4etkwg)………………………………………………………. 18

[2.4 The TESLA protocol and its variations](#_s081tkmm38h)…………………………………………... 18

[2.5 Conclusion](#_ah82v5qahm83)…………………………………………………………………………... 18

**3. Present Investigation**

[3.1 Threat Model](#_v2c2tbj6omip)………………………………………………………………………... 20

[3.2 Levels of Protection](#_z9naqpcx41)………………………………………………………………... 20

[3.3 Security requirements](#_6cynixyhlg0)……………………………………………………………... 22

**4. Our Plans**

4.1 Prevention………………………………………………………………………….... 25

4.2 Detection…………………………………………………………………………….. 25

**5. Conclusion**…………………………………………………………………………………………..26

**REFERENCES**…………………………………………………………………………………………. 28

**ARRANGEMENT OF CONTENTS OF PROJECT REPORT**

1. Title sheet
2. Acknowledgement
3. Statement of the Candidate

1. Certificate Sheet
2. Abstract
3. Table of Contents

1. Chapters
2. References

**REPORT ORIENTATION**

The report is comprised of four chapters with different content and scenarios providing the complete details about the project. The report is completed in such a way that it first provides the background knowledge about the project and then gives the thorough details about it. The different chapters of the report are as follows.

**Chapter 1: Introduction**

This chapter will provide introduction to the project and motivation for performing it. This chapter describes current scenario, previous attempts, and Problem statement and system requirements.

**Chapter 2: Literature survey**

This chapter will provide literature studied of the project. The focus would be on the technology going to be used. In this we give detailed explanation of CAN Bus, Trends in Research, Non-scholarly Literature and Summary .

**Chapter 3: Present Investigation**

This chapter contains the details of our learnings from the research done till now which contains the ways in which CAN bus can be attacked. It also explains the security requirements of the CAN bus.

**Chapter 4: Our Plans**

This chapter provides the details of the plans which we have to carry forward the research into implementing and testing various authentication schemes.

**Chapter 5: Conclusion**

This chapter contains the conclusions of the report and brief information about the work done.

**Chapter 1**

**Introduction**

# 

# **1. Introduction and Problem Statement**

Recent advances in in-vehicle technologies have paved way to a new era of connectivity.Nowadays, the control of different systems within a vehicle is carried out using tens of electronic control units (ECUs). These ECUs are clustered into networks having gateways in-between.The most widely used standard is the Controller Area Network (CAN) bus standard.Vehicle manufacturers have already deployed various technologies for driving assistance, anti-theft, and infotainment. They are now developing ways to interface mobile devices with vehicles and provide the customer's smartphone or tablet the ability to send/receive information to/from the car. However, such an integration introduces severe security risks to the vehicle. The in-vehicle network was originally designed to operate in a closed environment and thus, security was not of concern. It has now become an important issue due to an increasing number of external interfaces to the in-vehicle network. Several studies have already shown that an in-vehicle network can be easily compromised just by connecting cheap commercial devices and doing reverse engineering. Although research efforts have been made to secure in-vehicle networks, most of them focused on denying security requirements, or presenting attack scenarios without providing any feasible solution. Also, to the best of our knowledge, there hasn't been any study with a specific focus on understanding and analyzing the security aspects of integrating mobile devices with cars.

In this report, we will discuss about the working as well as present security scenario of CAN bus and threats associated with it. In our project we will try to find ways improve communication over CAN bus.

**1.1 Purpose**

The goal of the project is to attack the existing system of CAN bus in Cars and exploit its vulnerabilities and propose a preventive and detective measures of the same. Security is utmost important in the communication channel. As CAN bus is being widely used, attackers are intruding the system and exploiting it with the intention of theft, surveillance, kidnapping and terrorist attacks. Thus it is paramount to have the security layer in the CAN bus and avoid these intentions of the attackers by not intruding them. This project will give boost to the community for security in CAN Bus.

**1.2 Scope**

We are studying the working of the CAN protocol which is the widely used inside the Car so that many different components such as Anti Braking system, Door lock, ECU, Dashboard etc. can communicate at real time with very little delay. After studying its working inside the CAN Bus, we will start exploiting its security with various attacks on it. Thus conveying the security loop holes in CAN bus. We will then try to propose an authentication protocol as a preventive measure for CAN Bus and study its effects. We will also try to detect malicious attacks using Deep Neural Network as a state of an art solution.

**1.3 Motivation**

Lack of security in the CAN Bus Protocol which is one of the most upcoming technologies - automated cars - motivated us to undertake this project. Controlled Area Network is becoming the backbone of communication amongst the components in Car, due to which many attacks are taking place inside the Car. Thus trying to add the security layer on widely used protocol in Car which will reduce these attacks to a great extent.

**1.4 Problem Statement**

“The project is based on attacking of CAN Bus and prevention and detection of attack on CAN Bus. Developing an authentication scheme that would add a logical layer of security over the protocol without affecting the timeliness of the messages would be our ultimate goal. As an extended domain, we will also be detecting intrusion using Machine Learning Techniques.”

**Chapter 2**

**Literature Review**

# 

**2. Literature Review**

##### **2.1** **EVITA Project**

The largest project that aimed at securing in- vehicle communication networks is the EVITA project [1]. EVITA stands for E-safety Vehicle Intrusion proTected Applications. The project took place in the period from July 2008 to December 2011. The objectives of the project were to design, to verify, and to prototype an architecture for automotive on-board networks where security-relevant components are protected against tampering and sensitive data are protected against compromise. Thus, the goal of the project was to provide a basis for the secure deployment of electronic safety aids based on vehicle-to-vehicle and vehicle-to-infrastructure communications. The target was to complement other e-safety related projects that focus on protecting the communication of vehicles with the outside by focusing on on-board network protection.

This section discusses some of the work done within that project.

The EVITA project has inferred the following set of security requirements and related functional requirements in order to satisfy the stated security objectives [2]:

* Integrity / authenticity of ECU / firmware installation / configuration: Any replacement or addition of an ECU and/or its firmware or configuration to the vehicle shall be authentic in terms of origin, content, and time. In particular, the upload of new security algorithms, security credentials, or authorizations should be protected.
* Secure execution environment: Compromises to ECUs should not result in system wide attacks. Successful ECU attacks should have limited consequences on separate and/or more trusted zones of the platform.

* Vehicular access control: Access to vehicular data and functions should be controlled (e.g. for diagnosis, resources, etc.)

* Secure in-vehicle data storage: Applications should be able to use functionality in order to ensure access control to as well as the integrity, freshness and confidentiality of data stored within a vehicle, especially for personal information and security credentials.

* Confidentiality of certain on-board and external communication: The con- fidentiality of existing software/firmware as well as updates and security credentials shall be ensured. Some applications might additionally require that part of the traffic they receive or send internally or externally should remain confidential.

* Privacy: A privacy policy shall be enforceable on personal data stored within a vehicle or contained in messages sent from a vehicle to the outside. For example, some applications should limit the ability to link sent messages.

**2.1.1** **Security Module**

In [3], Marko Wolf et al introduced the idea of using a *security module* for providing different cryptographic functionalities to vehicles. Both centralized and distributed approaches are discussed. In the centralized approach, a single security module is used for providing security to several ECUs within the car. On the other hand, the distributed approach is based on attaching a security module to each ECU that needs protection. From implementation point of view, both software and hardware can be used for realizing such security module. In the case of centralized approach, the hardware implementation is more suitable, while in the case of distributed approach, the software implementation is more practical.

**2.1.2** **Key Distribution Protocol for CAN**

In [4], a key distribution protocol has been introduced for securing in-vehicle communications over CAN bus. Due to the embedded constraints, symmetric key cryptography was used. In each ECU, a hardware security module (HSM) was attached. The HSM implements some cryptographic primitives as well as securing the key storage. Moreover, it stores metadata for the keys (called user-flags). For example, a key may be tagged for signing at a node while it is tagged for verifying at another node.

The exchange of shared keys is done through a logical entity called “Key Master” (KM). Each node, has two keys to communicate with the KM; one for authentication and the other for transporting generated keys. To establish a secure communication channel between an ECU and other ECUs, the following steps are followed:

1. The ECU generates a pair of keys; one for generation and the other for verification.

2. The ECU sends the verification key encrypted to the KM.

3. The KM forwards the key encrypted to all other ECUs.

Those generated session keys are made valid for a limited time only. It is valid for one drive cycle for at most 48 hours. In order to be able to send messages larger than the standard CAN payload of 8 bytes, segmentation of data had to be used.

**2.1.3** **Conclusion**

EVITA project has made a considerable progress in securing in-vehicle commu- nications. To achieve this, the concept of adding a hardware security module (HSM) was introduced. The main disadvantage of using HSM is that it infers additional hardware cost to be added to the cost of manufactured vehicles.

##### **2.2** **Message Authentication Protocol over CAN**

**2.2.1** **Overview**

The problems associated with implementing a backward compatible message authentication protocol on the CAN bus has been discussed in [5]. The CAN bus - since its invention in 1986 by Robert Bosch GmbH - had its design being focused on safety. As a result, it has no built-in ways for inferring security. This lead to the demonstration of several means of attacks such as controlling brakes. Authentication protocol requirements listed in [5] are:

* Message authentication

It is required to make sure that the message has been sent from a trusted node.

* Replay attack resistance

Re-sending a previously sent message should lead to the message being discarded by all its receivers.

* Group keys

It is possible to use the same key for authenticating a group of messages.

* Backward compatibility

Nodes supporting authentication can co-exist with old nodes not supporting authentication.

The first requirement could be met by attaching a message authentication code (MAC) to a message. However, due to hard real time constraints on CAN messages, the used algorithm for generating the code and verifying it need to be fast. The algorithm proposed in the paper is HMAC [5] requesting the hash function used to be fast.

The second requirement could be met by inserting a counter value inside MAC calculations. The strength of this method will depend on the length of the counter. The larger the length of the counter, the less probable it is used more than once in a considerable amount of time.

For the third requirement to be met, number of messages in a group for which a single key can be used has to be specified within a particular CAN dataframe. This could create a problem for preempting communications for dataframes which does not belong to that particular group.

The last requirement is challenging because adding any extra data to a message will exceed the maximum possible length of a message (8 bytes). The proposed solution for use was to use an out-of-band protocol like CAN+ [6]. Using CAN+, additional data bits can be inserted within the transmission period of each CAN bit.

**2.2.2** **Authentication Protocol**

The authentication protocol starts with the establishment of a session key for each group of messages. The length of this key is 128 bits. This step assumes that all receiving nodes have a pre-shared key for this group of messages. It is also assumed that those keys are stored securely within each node. The node sending the group of messages (in case all messages of the group are sent by the same node) or one of the senders (in case not all the messages are sent by the same node) initiates the key establishment process. In this step, the responsible node sends a counter value together with a random number encoded as CAN+ data.

Each of the receiving nodes, will apply HMAC using the preshared key to the counter and the random number.

The purpose of the counter value used here is to prevent replay attacks. In such attacks, the adversary may try to resend a previously sent key establishment message. To prevent this scenario, each of the receiving nodes stores the last received counter value in non-volatile memory. The node does not accept key establishment unless the received counter value is greater than the last used value. This puts limitation on the lifetime of the protocol since the counter value has a limited length.

In the second part of key establishment, the initiating node sends a signature as shown in figure.

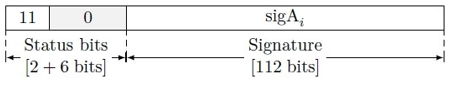


Figure 2.3: CANAuth Key Establishment Signature

By this step, all nodes verify that the initiating node have the same session key.

In runtime, the authentication of messages goes as follows.

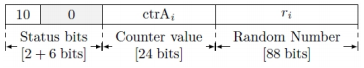


Figure 2.4: CANAuth Runtime authentication

As shown in the figure, authentication data consists of counter value and a signature. The counter value is used to prevent replay attacks. A node accepts a message, when the received counter value is greater than the last value. When the counter value is about to saturate, a new session key has to be established.

The handling of unauthorized messages uses the same error mechanism used by regular CAN nodes. In a regular CAN bus, any node can send an error frame at any time and thus invalidating the message that is being transmitted.

**2.2.3** **Security Analysis**

* **Adversary Model**

It is assumed that the adversary has access to the CAN bus and all messages transmitted on it. Hence, he can easily make a man-in-the-middle attack. However, it is assumed that the adversary has no access to the pre-shared keys stored within each node.

* **Denial of Service attacks**

During the key establishment phase, the adversary may alter the message that is being sent. This results in generating two different sessions keys. Using the protocol, this can be easily detected using the 2*nd* message of this phase. Thus, the nodes will retry to establish a new session key and so on.

**2.2.4** **Conclusion**

The idea of using CAN+ to send authentication data does not affect the communication bandwidth since data is sent out-of-band. This means that it does not require any modifications to be done to the existing CAN messages sets. However, it needs to use a modified physical layer. Therefore, it cannot be used with existing CAN controllers and transceivers.

##### **2.3** **Multiple MAC Per Receiver**

Using symmetric cryptography for multicast authentication has been introduced in [7] and [8]. In those papers, the sender creates multiple MACs for each message. Each MAC is calculated using a key that is shared between the sender and one of the receivers. The sender appends all those MACs to the message being transmitted. Each receiver uses its key to verify part of the MAC. The papers [7] [8] are concerned with multicast authentication for automotive networks including CAN, FlexRay and Time-Triggered Protocol. Concerning CAN, it is proposed to use only half of the payload of each CAN message for carrying data, while using the remaining 4 bytes for carrying MACs. If each MAC is composed of one byte, then the message can carry 4 MACs corresponding to 4 receivers.

##### **2.4** **The TESLA protocol and its variations**

TESLA protocol was proposed in [9] as a new protocol for multicast authentication. The main idea behind it is to achieve asymmetric properties by using delayed disclosure of keys. However, this leads to delayed authentication. This delayed authentication has two main drawbacks. First, the receiver need to have storage for some unauthenticated messages till their key is disclosed. This increases the effect of DoS attack where an attacker may flood the receiver with many wrong messages. The second point is that this makes TESLA not suitable for real-time systems.

##### 

##### **2.5** **Conclusion**

Several CAN authentication protocols have been proposed before. Some of them were based on hardware while others were based on software. Using hard- ware is more efficient either from point of view of security strength. However, it requires additional cost to be added to each ECU that is being manufactured. Using software looks more promising if it can provide the required security strength. Therefore, it is required to design an efficient CAN authentication protocol that can be implemented by software.

**Chapter 3**

**Present Investigation**

# 

**3. Present Investigation**

##### **3.1** **Threat Model**

In-vehicle networks consist of various ECUs that are interconnected using communication buses. There are many types of communications buses used like CAN , LIN , FlexRay and MOST. CAN is the most widely used type of buses. As discussed earlier, CAN messages do not contain any source or destination addresses. Also, they do not provide any means of authentication. Hence, any adversary node that succeeds to have access to the bus can listen to any transmitted message. Moreover, it can inject malicious messages into the network. In our design we will aim to mitigate two major types of attack: replay attacks and brute force attacks.

##### **3.2** **Levels of Protection**

In order to protect vehicles from various attacks, we recommend three levels of protection:

• Level 1: Prevent ECUs from being compromised

• Level 2: Protect the vehicle’s internal network against a compromised ECU

• Level 3: Detect the compromise of an ECU

**3.2.1** **Level 1: Prevent ECUs from being compromised**

The first level of protection is to protect every ECU inside the vehicle from being compromised. There are many ways by which an ECU can be compromised. Here is a list of some of them:

• **Flash Bootloader.** An ECU can be compromised by replacing the flashed software by another malicious software using the flash bootloader. Typically, the flash bootloader is a piece of software that facilitates the update of the software flashed on an ECU. It is used by car manufacturers to update the software of ECUs when necessary. Basically, accessing the bootloader is secured by challenge-response pair. To protect the ECU, the value of the key used in the authentication algorithm shall be unique for each ECU. Also, the length of the key used should be large enough in order to be protected against brute-force attacks. The value of the key stored in the flash memory of the microcontroller shall be secured against malicious reading. In order to protect the ECU from brute-force attacks, if the wrong response is entered for few times, the ECU shall be locked and cannot enter bootloader mode anymore.

• **Flasher.** In this case, the adversary has direct physical access to the ECU. The protection against such attack can be done by censoring the flash memory during the production. By that way, the bootloader is the only entity that is granted the access to flash the ECU. It is worth noting that this type of attack is special because the attacker who can connect a flasher to the ECU can replace the microcontroller itself or even the whole ECU rather than replacing the software.

• **Exploiting vulnerabilities.** When the software of the ECU contains some vulnerabilities, they can be used by an attacker to insert his malicious code. Several types of vulnerabilities exist. Among these are the buffer overflow. The way of protection for such attacks is to perform static analysis to the software to ensure that the software is free from such vulnerabilities.

**3.2.2** **Level 2: Protect the vehicle’s internal network against a compromised ECU**

In the second level of protection, we assume that at least one ECU was compromised by an adversary. The compromised ECU can be one of two cases:

* **Case A:** An ECU which is part of the vehicle’s internal network is compromised (using any of the ways mentioned in Level 1 protection).
* **Case B:** Any malicious node that is connected to the bus by the adversary.

The goal of protection in this level is to prevent those compromised ECUs from harming the vehicle’s internal communication network. The compromised ECU can harm the vehicle’s network by doing any of the following actions:

1. Send a malicious message on behalf of another node.

This type of attack arises from the nature of the CAN bus since it does not specify methods of source authentication. As a result, the compromised ECU may send false messages on behalf of other ECUs. This can be protected by using a source authentication protocol in the communication in the CAN bus.

2. Send a malicious message that it normally sends.

Unfortunately, this type of attack cannot be protected after the ECU is already compromised. This is because whatever cryptographic methods used, the compromised software can still use this secured communication stack to send messages containing false data.

3. Perform a denial of service attack on the CAN bus. This can be protected by using something like the bus guardian of the FlexRay protocol.

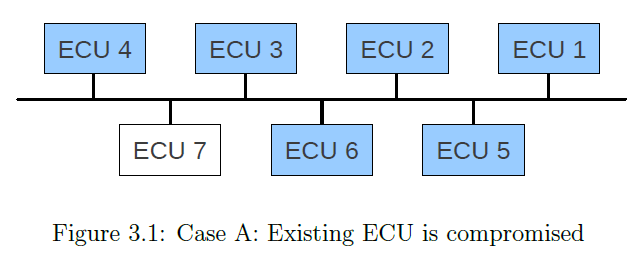
**3.2.3** **Level 3: Detect the compromise of an ECU**

As a 3*rd* level of protection, it should be possible to detect the compromise of any node of the vehicle’s internal network. The process of checking ECUs can either be done regularly by the vehicle or at least be done during regular service of the vehicle. The proposed way is to verify the checksum of the flash contents of each ECU. This can be the responsibility of a new Security ECU or the body controller. When the software of any ECU is updated, then the new checksum must be communicated to the Security ECU.

##### **3.3** **Security requirements**

Levels 1 and 3 can be viewed as recommendations that can be adopted directly by vehicles’ manufacturers. However, level 2 needs some research. We will focus on solving the problems mentioned above in Level 2. In this case, it is required to protect the vehicle’s internal network from a compromised node. We will focus on the two cases A & B:

**Case A:** The threat model of this case is as follows. The CAN bus has some ECUs connected to it. One of these ECUs was previously compromised by an adversary.



**Case B:** The threat model of this case is as follows. The CAN bus has some ECUs connected to it. All ECUs are communicating correctly. The adversary node is attached to the CAN bus at some point of time. According to the CAN bus specification, the attached node can listen to all exchanged CAN messages. Also, it has the ability to send any CAN message on behalf of any other node. For both cases, in order to protect the network against such attacks, message source authentication is required. However, the CAN bus protocol does not specify any means of authentication. As a result, it is required to design a higher level authentication protocol that can be adopted in automotive CAN networks.

The following requirements are needed when designing the authentication protocol:

1. The message shall contain an evidence that can be *generated* only by its trusted sender.

2. The receiver shall be able to *verify* that evidence.

3. The receiver shall not be able to *re-transmit* the message masquerading the trusted sender.

4. The protocol shall add a small communication overhead. The payload of any CAN message is already too small (8 bytes by maximum). For the current networks, those 8 bytes are highly utilized. Thus, the smaller the overhead used, the easier to deploy the authentication protocol in the currently designed networks with minimum re-formatting of messages.

5. The protocol shall not require either heavy computation or high memory consumption. This is because the currently produced ECUs use microcontrollers with limited resources.

**3.3.1 Basics of authentication**

Message source authentication between a sender and a single receiver can be achieved by adding a message authentication code (MAC) to each transmitted message. The sender calculates the MAC based on the message contents and a shared secret. On the other side, the receiver uses the same shared secret to verify the MAC. This can be achieved using symmetric cryptography.

When there is more than one receiver - the case of multicast communication - another way shall be used. The shared secret cannot be shared between more than two entities. Therefore, asymmetric cryptography is needed. For each transmitted message, the sender creates a MAC using its private key. Various receivers use the sender public key to verify the MAC. However, asymmetric cryptography is not recommended for our domain because it needs high computational capabilities.

TESLA is one of the most famous authentication protocols for multicast communications. However, if we try to adopt it in CAN networks we will find that we will need large communication overhead. This overhead is because each message shall contain the original data to be transmitted in addition to the MAC and a key. Also, the delay introduced by TESLA in unacceptable for in-vehicle networks as real-time systems.

**Chapter 4**

**Our Plans**

**4. Our Plans**

In this section we will discuss various ways in which we will deal with the attacks on CAN bus in our project.

**4.1 Prevention**

Prevention of an attack on CAN bus can be done in two ways.

* Encryption
* Authentication

CAN's small data field makes it difficult to implement encryption. Hence, while block and stream cipher algorithms are commonly used for data encryption in regular networks, such mechanisms are not practical in vehicular networks.It is therefore difficult to use encryption as a preventive measure.

In our project, we will try to analyse different authentication schemes discussed in various papers by manually implementing them on our hardware. In addition to this, based on our research we will try to propose an authentication scheme which will be having advantages over other schemes.

**4.2 Detection**

We will try to implement an intrusion detection system (IDS) using a deep neural network (DNN) that is proposed to enhance the security of in-vehicular network. The parameters building the DNN structure are trained with probability-based feature vectors that are extracted from the in-vehicular network packets. For a given packet, the DNN provides the probability of each class discriminating normal and attack packets, and, thus the sensor can identify any malicious attack to the vehicle. As compared to the traditional artificial neural network applied to the IDS, this technique adopts recent advances in deep learning studies such as initializing the parameters through the unsupervised pre-training of deep belief networks (DBN), therefore improving the detection accuracy. It is demonstrated with experimental results that this technique can provide a real-time response to the attack with a significantly improved detection ratio in controller area network (CAN) bus.

**Chapter 5**

**Conclusion**

We have understood the working of CAN Bus Technology by going through all the research and review paper. We have purchased the CAN BUS Shield consisting of MCP2515 CAN Controller and MCP2551 CAN Transceiver. We have started with the sending and receiving of the CAN Packets. Our main goal is to exploit the security of the CAN BUS Protocol and perform various attacks on it and implement the prevention and detection of the malicious attacks on CAN BUS Protocol. Intrusion Detection of attack in CAN BUS can be promisingly detected using Models of Deep Neural Networks which will be our main focus in future to understand its possibility and implementing the same within our domain.

**REFERENCES**

[1] E-safety vehicle intrusion protected applications (EVITA). Available:<http://evita-project.org/>

[2] L. Apvrille, R. El Khayari, O. Henniger, Y. Roudier, H. Schweppe, H. Seudi´e, B. Weyl, and M. Wolf, “Secure automotive on-board electronics network architecture,” in *FISITA 2010, World Automotive Congress, 30 May-4 June 2010*, Budapest, Hungary, May/Jun. 2010.

[3] M. Wolf, A. Weimerskirch, and T. J. Wollinger, “State of the art: Embedding security in vehicles,” *EURASIP Journal on Embedded Systems*, vol. 2007, 2007.

[4] H. Schweppe, Y. Roudier, B. Weyl, L. Apvrille, and D. Scheuermann, “Car2x communication: Securing the last meter - a cost-effective approach for ensuring trust in car2x applications using in-vehicle symmetric cryptography,” in *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, Sep 2011, pp. 1–5.

[5] A. V. Herrewege, D. Singelee, and I. Verbauwhede, “CANAuth - a simple, backward compatible broadcast authentication protocol for CAN bus,” in *9th Embedded Security in Cars Conference*, Dresden, Germany, Nov 2011.

[6] T. Ziermann, S. Wildermann, and J. Teich, “CAN+: A new backward- compatible controller area network (CAN) protocol with up to 16x higher data rates.” in *Design, Automation Test in Europe Conference Exhibition, 2009. DATE ’09.*, Apr 2009, pp. 1088–1093.

[7] C. Szilagyi and P. Koopman, “A flexible approach to embedded network multicast authentication,” in *3rd Workshop on Embedded Systems Security (WESS’2008)*, Atlanta, Georgia, USA, Oct 2008.

[8] “Flexible multicast authentication for time-triggered embedded con- trol network applications,” in *Proc. of the International Conference on Dependable Systems and Networks (DSN 09)*, Lisbon, Portugal, Jun/Jul 2009, pp. 165–174.

[9] A. Perrig, R. Canetti, J. Tygar, and D. Song, “Efficient authentication and signing of multicast streams over lossy channels,” in *Security and Privacy, 2000. S P 2000. Proceedings. 2000 IEEE Symposium on*, Berkeley, CA, USA, May 2000, pp. 56–73.