Vehicle-to-Vehicle Communications Misbehavior Detection
# Table of Contents

**Executive Summary** ...................................................................................................................... 5  

1 **Introduction** ................................................................................................................................. 7  
   1.1 Background ................................................................................................................................. 7  
   1.2 Purpose ...................................................................................................................................... 8  
   1.3 Approach .................................................................................................................................... 8  
   1.4 Report Organization .................................................................................................................... 8  

2 **Conceptualization of MBD in the OBE and SCMS** .............................................................. 10  

3 **Interface between LMBD and GMBD** .................................................................................. 12  
   3.1 OTA Interface ............................................................................................................................. 12  
   3.2 Misbehavior Report Contents .................................................................................................... 13  

4 **LMBD Design and Implementation** ...................................................................................... 16  
   4.1 OBE Software Platform for LMBD .......................................................................................... 16  
      4.1.1 Standard SW Application Processes ............................................................................... 18  
      4.1.2 LMBD Application Processes ......................................................................................... 19  
      4.1.3 Process Data Relationship ............................................................................................... 19  
   4.2 Proximity Plausibility LMBD Method Design ......................................................................... 20  
      4.2.1 Proximity Plausibility Method Overview ....................................................................... 20  
      4.2.2 Proximity Plausibility Logic ............................................................................................. 22  
      4.2.3 Proximity Plausibility Method for Detecting Vehicle Overlap ....................................... 24  
      4.2.4 Proximity Plausibility MBR Contents ............................................................................. 26  
      4.2.5 Proximity Plausibility Method Configuration ................................................................. 26  
   4.3 Warning Report Design ............................................................................................................... 27  
      4.3.1 Warning Report Method Overview .................................................................................. 27  
      4.3.2 Warning Report Logic ......................................................................................................... 28  
      4.3.3 Warning Report Processing of Driver Reaction ............................................................... 31  
      4.3.4 Warning Report MBR Contents ......................................................................................... 31  
      4.3.5 Warning Report Method Configuration ............................................................................ 32
## GMBD Design and Implementation

### 5.1 GMBD Server Network and Component Distribution

- **5.1.1 MBR Receiver**
- **5.1.2 Threshold Manager**
- **5.1.3 Linkage Information Manager (LIM)**
- **5.1.4 Repository Manager**

### 5.2 Process Flow

- **5.2.1 MBR Receipt**
- **5.2.2 Workflow Scheduling**
- **5.2.3 Query Results Retrieval**
- **5.2.4 Suspect Group Interrogation**
- **5.2.5 Revocation**

### 5.3 GMBD Methods

- **5.3.1 Device-based GMBD Method**
- **5.3.2 Event-based GMBD Method**
- **5.3.3 Feature-based GMBD Method**

### 5.4 GMBD Configuration

### 6 Test Data and Testing

- **6.1 Test Data**
- **6.2 LMBD Testing and Validation**
- **6.3 GMBD Testing and Validation**

### 7 Summary

### 8 References

### APPENDIX A. List of Acronyms
List of Figures

Figure 1: Conceptual OBE and SCMS MBD Component Structure ........................................ 10
Figure 2: OTA Communication Interface between OBE and GMBD Server ................................ 13
Figure 3: High-level OBE Platform with LMBD Support .................................................. 18
Figure 4: LMBD Software Architecture and Data Flow ...................................................... 20
Figure 5: Proximity Plausibility Virtual Box Overlap Example ........................................... 21
Figure 6: Proximity Plausibility Example Scenario ............................................................ 22
Figure 7: Proximity Plausibility LMBD Logic Flow ............................................................ 24
Figure 8: Warning Report LMBD Logic Flow ...................................................................... 31
Figure 9: GMBD Components and Workflow Overview .................................................... 34
Figure 10: Overall GMBD Process Flow ............................................................................ 36
Figure 11: MBR Receipt Flow ............................................................................................ 36
Figure 12: Workflow Scheduling ....................................................................................... 37
Figure 13: Query Results Retrieval ..................................................................................... 38
Figure 14: Suspect Group Interrogation .............................................................................. 39
Figure 15: Event-based GMBD Method – Counting Bloom Filter Example Diagram .................. 41
Figure 16: Feature-based GMBD Method – Neighbor Counting Example ............................... 43
Figure 17: Feature-based GMBD Method – Neighbor Count Clustering Visualization ............ 44
Figure 18: V2V-MD DAS Data Export Process Flow ............................................................. 47
Figure 19: V2V-MD DAS Exported Chart Relating to an FCW Event ....................................... 48
List of Tables

Table 1: MBR – Common Content ................................................................. 14
Table 2: MBR – Proximity Plausibility Content ........................................ 15
Table 3: MBR – Warning Report Content ................................................ 15
Table 4: MBR – Test Report Content ............................................................ 15
Table 5: Warning Report Method Concept ................................................. 28
Table 6: VAD / ASD Top Offenders Event List Excerpt .............................. 46
Executive Summary

This document presents the findings and results of the research conducted by the Crash Avoidance Metrics Partners LLC (CAMP LLC) Vehicle Safety Communications 6 (VSC6) Consortium pertaining to Misbehavior Detection (MBD). The members of the CAMP VSC6 Consortium are Ford Motor Company, General Motors LLC, Honda R&D Americas, Inc., Hyundai-Kia America Technical Center, Inc., Nissan Technical Center North America, and Volkswagen Group of America. The work conducted in the Project was a continuation of the MBD work performed in the Interoperability Issues of Vehicle-to-Vehicle Based Safety Systems (V2V-Interoperability).

The V2V-Interoperability Team, working with two network security experts, developed definitions, assumptions, as well as an attack model analysis that were then used in conceptualizing five (5) Local Misbehavior Detection (LMBD) methods and three (3) threshold-based Global Misbehavior Detection (GMBD) methods. MBD was addressed from two aspects; LMBD and GMBD, where:

- Misbehavior is the willful or inadvertent transmission of incorrect data within the Vehicle-to-Vehicle (V2V) network
- MBD is the process of identifying misbehavior within the V2V network
- LMBD is the process of identifying misbehavior at the local / vehicle level using in-vehicle algorithms and processing
- GMBD is the process of identifying misbehavior using more advanced processing at the Misbehavior Authority (MA) within the Security Credential Management System (SCMS)

The goal of the Project was to conceptually prove the viability of one or more of the LMBD and GMBD methods and the development of the reporting interface between LMBD and GMBD. The methods implemented in this project were not stress tested, tested in the real world, nor developed to a deployment-ready level. The algorithms were tested to validate the SW implementation and begin testing the viability of the conceptual implementation of the algorithms. Further development of the GMBD methods will take place in the SCMS Project. Following is a discussion of the efforts to support the goal mentioned above.

Given the intrinsically linked nature of LMBD and GMBD, it was desirable for the LMBD and GMBD development to proceed in parallel. To this end, the Project implemented a lab environment in which an On-Board Equipment (OBE) device was able to use a Road Side Equipment (RSE) device to communicate with the SCMS using Internet Protocol Version 6 (IPv6). The RSE was used to transmit a Wireless Access in Vehicular Environment (WAVE) Service Advertisement (WSA) which advertised the availability of the misbehavior reporting service and the Internet Protocol (IP) address of the server. Due to the existing OBE supporting another SCMS interface rather than the updated Proof-of-Concept (PoC) SCMS interface, the communication protocol for the OBE to send a Misbehavior Report (MBR) to the SCMS was simplified to use an unsecured and unencrypted IPv6 Transmission Control Protocol (TCP) / IP Connection. For this project, the MBR was based on Basic Safety Message (BSM) metadata, which consisted of:

- The time and location where the MBR was created
The reporter security Linkage Value (LV) information
- The LMBD method that caused the MBR creation
- Some combination of the start and stop time and location of the suspected misbehavior and attacker LV information, depending on the LMBD method

After the lab setup was completed and communication between the OBE and SCMS server was confirmed, the requirements and design for two of the LMBD methods identified in the V2V-Interoperability Project were prioritized for development and an existing OBE Software (SW) implementation was updated to support the LMBD methods and MBR transmission. For GMBD, two methods were selected for implementation while the framework was put in place for a third method. There was no prior SW / Hardware (HW) base to build upon for GMBD so the components of GMBD were divided between an Application Server and a Database Server. For each of the components, detailed design and implementation activities were performed. To support the conceptual proof, a number of configuration items were defined to control the behavior of the methods for both LMBD and GMBD.

The methods were tested at a conceptual level, which means the methods were not tested rigorously to validate the maturity of the method nor were the deployment ready requirements developed. Instead, the methods were tested to validate the ability to detect misbehavior by using a given data set suspected of containing misbehavior. The methods were tested with the minimum and maximum configurable parameter thresholds to support the validation. The primary data set used for the testing was data from the Data Acquisition System (DAS) installed by Virginia Tech Transportation Institute (VTTI) in each of the CAMP Vehicle Safety Communications (VSC3) Vehicle-to-Vehicle Safety System Light Vehicle Builds and Model Deployment Support (V2V-MD) Project Integrated Light Vehicles (ILVs) which were deployed in the SPMD Test Bed. Utilizing the SPMD data was desirable because, based on previous analysis, it was known that there were instances of non-ideal hardware installation as well as incorrect SW configuration on some of the Vehicle Awareness Devices (VADs) and Aftermarket Safety Devices (ASDs) which should trigger LMBD by the two methods selected for this project.

VTTI compiled a list of the VADs / ASDs that were deemed as the top offenders in causing warnings in ILVs along with the corresponding warning events and event timing. The data from these events were then exported into Comma Separated Value (CSV) data files that included all the necessary BSM data from the Remote Vehicle (RV) identified in the top offenders table, other RVs that were within communication range at the time, as well as BSM data from the Host Vehicle (HV).

For LMBD, the primary data used for testing methods were the CSV files generated from the process described above. The test data used for validating the GMBD methods were the MBRs generated by the LMBD methods. Note that since only the framework for one of the GMBD methods was developed and implemented and given the limitations on the availability of test data, this method was not tested in detail.

When provided inputs, which included device behavior consistent with that which should trigger LMBD, the results of the testing showed that the LMBD methods could detect the perceived misbehavior. Similarly, GMBD testing showed that the methods could analyze multiple reports and successfully link them as a common device and, depending on the configured thresholds, flag this device for revocation if necessary. It should be noted that these results were obtained in a lab setting in which limited field acquired data sets, selected because they mimicked misbehavior, were replayed into the OBEs. This work should be considered an initial step in MBD research.
1 Introduction

This document presents the findings and results of the research conducted by the Crash Avoidance Metrics Partners LLC (CAMP LLC) Vehicle Safety Communications 6 (VSC6) Consortium pertaining to Misbehavior Detection (MBD). The members of the CAMP VSC6 Consortium are Ford Motor Company, General Motors LLC, Honda R&D Americas, Inc., Hyundai-Kia America Technical Center, Inc., Nissan Technical Center North America, and Volkswagen Group of America. The work conducted in the Project was a continuation of the MBD work performed in the Interoperability Issues of Vehicle-to-Vehicle Based Safety Systems (V2V-Interoperability) Project. The V2V-Interoperability Team, working with two network security experts, developed definitions, assumptions, and an attack model analysis that were then used in conceptualizing five (5) Local Misbehavior Detection (LMBD) methods and three (3) threshold-based Global Misbehavior Detection (GMBD) methods. This report addressed MBD from two aspects; LMBD and GMBD, where:

- Misbehavior is the willful or inadvertent transmission of incorrect data within the Vehicle-to-Vehicle (V2V) network
- MBD is the process of identifying misbehavior within the V2V network
- LMBD is the process of identifying misbehavior at the local / vehicle level using in-vehicle algorithms and processing
- GMBD is the process of identifying misbehavior using more advanced processing at the Misbehavior Authority (MA) within the Security Credential Management System (SCMS)

This work was used as the input to begin developing preliminary higher-level aspects of a software (SW) design for both LMBD and GMBD as well as to begin defining the interface between the LMBD and GMBD.

1.1 Background

V2V technology can help reduce vehicular related crashes and fatalities by warning drivers when a crash between two or more vehicles is likely to occur. Vehicles equipped with V2V On-Board Equipment (OBE) periodically transmit Basic Safety Messages (BSMs) as specified in the SAE International (SAE) J2945/1 Standard [1]. The BSMs contain information about the vehicle’s location and dynamics. Based on this information, surrounding vehicles within the network of connected vehicles can determine if a crash with other neighboring vehicles is likely and, if so, provide warnings to drivers to avoid or minimize the severity of crashes.

If all future manufactured vehicles contain V2V OBE, the V2V network will eventually become a very large one, estimated to include 200 to 300 million vehicles. In addition to equipment/component failures that may, if not properly diagnosed, result in the inadvertent broadcast of inaccurate information, it is anticipated that a network of this size may become a target for actors with malicious intent to cause misbehavior. Within any large-scale network, anomaly or misbehavior detection becomes a very complex problem. Misbehavior within the network may differ but the outcome of misbehavior, if successful, will likely result in false warnings when a warning is not warranted or missed warnings when a warning is warranted. To do this, an attacker or a malfunctioning device will likely attempt to transmit messages with incorrect/false data. MBD is the process of identifying willful mischief or incorrect data transmission by malfunctioning devices.
In the full deployment of a V2V network, once a misbehavior and the misbehaving device have been identified by the MA within the SCMS, information related to the device credentials will be added to a Certificate Revocation List (CRL) which indicates to other vehicles within the DSRC network that the credentials of the device have been revoked. It is important to note that the CRL implementation and testing was not in the scope of the work of the Project and, therefore, is not included in this document.

1.2 Purpose

The purpose of this document is to provide an overview of misbehavior detection. Primarily, this document discusses:

- The interface developed to support LMBD and GMBD functionality
- The two (2) LMBD methods (of the original five) that were implemented
- Three (3) GMBD methods that were implemented
- The data used to test the LMBD methods
- Other ancillary topics that enabled the implementation and testing of the methods

1.3 Approach

In this project, two suppliers, DENSO and Green Hills Software, were contracted for the design, implementation, and test support of the LMBD and GMBD, respectively. The approach was to select an initial MBD method for design and development and then, after testing and analysis of this method, repeat the process such that the MBD capability was built up in an incremental fashion. This would allow the Team to more quickly test the Over-the-Air (OTA) and other interfaces and work out issues with them prior to adding further complexity. Since the focus was on conceptual method development and proof of concept, the supporting method framework (e.g., storage, reporting) was developed only to the point necessary to support this. To support the incremental testing of the systems, the implementation was separated into multiple SW releases for both the OBE (LMBD side) and GMBD.

Because the GMBD methods will eventually reside within the SCMS as part of the MA and due to the tight coupling between LMBD, GMBD, reporting, and revocation, a well-managed process was put in place to ensure that the appropriate information is shared between the projects and that the development of these items is kept in sync across the two projects. Since Green Hills Software was selected to perform the GMBD development and is also involved in the enhanced SCMS development, this aided in the information sharing.

It is important to note that the methods implemented in this project were not stress tested, tested in the real world, or developed to a deployment-ready level. The algorithms were tested to validate the SW implementation and begin testing the viability of the conceptual implementation of the algorithms.

1.4 Report Organization

This report is organized into 8 chapters.

- Chapter 1: Introduction
- Chapter 2: Discusses a conceptual OBE and SCMS structure regarding the components involved in MBD
• Chapter 3: Discusses the interface between LMBD and GMBD
• Chapter 4: Details the high-level architecture, design, and implementation activities regarding LMBD
• Chapter 5: Details the high-level architecture, design, and implementation activities regarding GMBD
• Chapter 6: Discusses the test data and test cases that were run
• Chapter 7: Provides a summary of the project activities
• Chapter 8: Contains a list of references cited in the report
2 Conceptualization of MBD in the OBE and SCMS

Figure 1 shows a conceptual OBE and SCMS structure regarding the components involved in MBD. The nonhashed out items, as shown in Figure 1, were the focus of the design, implementation, and testing since they are directly linked to the project focus of conceptual method development and proof of concept.

Figure 1: Conceptual OBE and SCMS MBD Component Structure

The left side of Figure 1 shows the information flow within the OBE. This is the basis of the MBD as it feeds information to perform GMBD. Specifically, the LMBD Methods, which take input from the vehicle Controller Area Network (CAN) and Global Positioning System (GPS) receiver, safety applications, and BSMs received OTA from surrounding vehicles, provides the basis for the generation of the misbehavior report (MBR) which is sent to the MA that resides on the SCMS.

The right side of Figure 1 shows the information flow within the SCMS. The GMBD Methods, take input from the MBR received by the vehicles and perform an aggregate analysis, which provides the basis for placing a vehicle on the CRL. The CRL is then sent to the vehicles which completes the end-to-end MBD and Revocation process.

Some notes on Figure 1:

- In the deployed system, the reports will be sent to the Registration Authority (RA) within the SCMS. Since the focus of this project was on method development, that route was bypassed.
and the reports were send directly to the MA as indicated by the dashed line on the MBR Interface in Figure 1.

- The V2V Interface and Vehicle CAN / GPS components are critical components in the LMBD process and, while not used physically in this project, the data that would be obtained from these components was obtained via emulated field data. The method / approach for doing this is discussed in Section 6.1.
3 Interface between LMBD and GMBD

Given the intrinsically linked nature of LMBD and GMBD, it was desirable for the LMBD and GMBD development to proceed in parallel. To this end, the early development focused on the OTA interface between the OBE and the GMBD servers and a preliminary MBR was developed. Then, as development progressed, updates were made to the interface and MBR as necessary.

3.1 OTA Interface

For LMBD reporting, it was desirable that any interface testing be performed OTA via a physical medium that would be used in deployment, rather than via a hard-wired connection. While multiple options exist, DSRC was used as the medium for this project. Therefore, a Road Side Equipment (RSE) was installed in the CAMP VSC6 Lab and configured to broadcast Wireless Access in Vehicular Environment (WAVE) Service Advertisement (WSA) indicating the availability of misbehavior reporting services on channel 172. Channel 172 was used to transmit WSAs to reduce the complexity of the system and eliminate the need for an additional radio or channel switching capabilities on the OBE. The RSE allowed for the OBE to establish an OTA Internet Protocol Version 6 (IPv6) connection with the GMBD server. Figure 2 shows a generalized overview of the connection methodology used in the project for the OBE to communicate with the SCMS. Note that for this setup, given that the RSE was installed in the lab, a hard-wire connection did exist from the RSE to the GMBD server.
Since the focus of the work was on conceptual LMBD and GMBD method development and proof of concept, the communication between the OBE and SCMS was simplified to use an unsecured and unencrypted IPv6 Transmission Control Protocol (TCP) / Internet Protocol (IP) connection.

### 3.2 Misbehavior Report Contents

The interface developed in this project was based on the original MBD work from the V2V-Interoperability Project which suggested including complete and signed BSMs in the report. However, given that the OBE supported the SPMD security format rather than the SCMS PoC format, BSM metadata was included in the report instead of fully-formed and signed BSMs. This was deemed acceptable, considering the focus of the work was on conceptual LMBD and GMBD method.
development and proof of concept and that the methods will primarily operate on BSM metadata. The security content is required for validating the MBR, which was outside the scope of this Project, as well as linking the reports. For the latter, an alternate means was devised (see Section 5.2.4).

The contents contained within the MBR for this project are described in Table 1 through Table 4. Table 1 lists the common data elements that are contained in each report. Table 2 and Table 3 list the additional data elements included for the Proximity Plausibility and Warning Report LMBD method generated reports. Table 4 lists the data elements of a Test Report that was defined to support testing the interface prior to the two LMBD methods being developed.

**Table 1: MBR – Common Content**

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three-Dimensional (3D) Location</td>
<td>The location of the reporting OBE, where the misbehavior was observed and the report was created</td>
</tr>
<tr>
<td>MBD Time</td>
<td>The absolute time at which the reporting OBE observed the misbehavior and created the report</td>
</tr>
<tr>
<td>Reporter LV</td>
<td>The reporter's 4-byte Linkage Value (LV)</td>
</tr>
<tr>
<td></td>
<td>• For the purposes of this project, the LV is the four byte tempID contained in Part I of the BSM</td>
</tr>
<tr>
<td></td>
<td>• This is used temporarily in place of the pseudonym certificate which will replace this when the MBR is fully signed via a separate project</td>
</tr>
<tr>
<td>Report Type</td>
<td>Indication of which LMBD method content is contained in the report:</td>
</tr>
<tr>
<td></td>
<td>• Proximity Plausibility Report</td>
</tr>
<tr>
<td></td>
<td>• Warning Report</td>
</tr>
<tr>
<td></td>
<td>• Test Report</td>
</tr>
<tr>
<td>Number of RVs</td>
<td>The number of RVs at the time that misbehavior was observed and the report was created corresponding to the 3D Location and MBD Time</td>
</tr>
</tbody>
</table>

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium
Table 2: MBR – Proximity Plausibility Content

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| Suspected Attacker LV| The suspected attacker’s 4-byte LV
  • For the purposes of this project, the LV is the 4-byte tempID                                |
| Start 3D Location    | The location of the initial proximity plausibility overlap                                             |
| Start Time           | The absolute time of the initial overlap                                                               |
| End 3D Location      | The location of the final overlap and/or the location at which the relevant thresholds were exceeded, and it was determined that an MBR is needed |
| End Time             | The absolute time of the final overlap and/or the time at which the relevant thresholds were exceeded, and it was determined that an MBR is needed |

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium

Table 3: MBR – Warning Report Content

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| Warning RV LV List   | A sequence of RV LVs that caused warnings on the HV
  • For the purposes of this project, the LV is the 4-byte tempID                                |

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium

Table 4: MBR – Test Report Content

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dummy Data</td>
<td>An octet string of dummy data used to populate the test report with varying lengths</td>
</tr>
</tbody>
</table>

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium
Chapter 4: LMBD Design and Implementation

4 LMBD Design and Implementation

The following sections provide more specific details on the components of the OBE that were implemented and/or utilized in the project and where they fit in a reference OBE architecture. For the LMBD methods, the V2V-Interoperability Project identified five potential LMBD methods for consideration:

- Proximity Plausibility
- Warning Report
- Motion Validation
- Content and Message Verification
- Denial of Service Detection

For this project, the Proximity Plausibility and Warning Report were prioritized for implementation. Proximity Plausibility was implemented first since it was determined that this method would be the easiest method to test based on the available data. This was followed by the implementation of the Warning Report Method.

4.1 OBE Software Platform for LMBD

The OBE SW used in this project was based on SW developed by DENSO in past CAMP VSC Consortia projects and runs on DENSO Wireless Safety Unit (WSU) OBE hardware (HW). As such, the WSU has been developed as a platform to support current V2V and Vehicle-to-Infrastructure (V2I) requirements. The WSU SW consists of a set of processes that run as separate executables and exchange data via shared memory. See Figure 3 as a reference for the high-level platform of the OBE. For misbehavior detection, two new processes were created and one process was revised.
Chapter 4: LMBD Design and Implementation

DENSO WSU

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium

Vehicle Safety Communications 6 (VSC6) Consortium Proprietary
4.1.1 Standard SW Application Processes

The standard software application layer processes running on the WSU with a role in LMBD and reporting include:

1. The Wireless Message Handler (WMH)
   The WMH parses and stores data from incoming BSMs and is responsible for creating outgoing BSMs from the Host Vehicle (HV). It has been modified for MBD to observe and identify Remote Vehicle (RV) certificate changes. When an RV certificate changes, it also means the RV is changing its temporary identifier (TempID). The result is that unless the WMH recognizes a new TempID is from the same RV as a previous TempID, for a configurable period of time immediately after the certificate change, there will be two separate RVs on the RV list with nearly the same coordinates. To the Proximity Plausibility Algorithm, this would appear as an overlap and would be likely to cause a false proximity violation.
   
   By performing a short-term binning of an RV as it changes certificates, the WMH can prevent the same vehicle from appearing twice in the RV list under different identities, and, thus, prevent the erroneous proximity violation.

2. DENSO Over-the-Air-Processor (DOTAP)
   For misbehavior detection, DOTAP listens for, and parses, WSA messages broadcast from an RSE. Data from a WSA is made available in shared memory for other processes to use. For misbehavior detection, the IP address and port to use for connecting to the GMBD server are the key data needed from DOTAP.
   
   DOTAP was not changed for misbehavior detection.

3. Sensor Data Handler – CAN bus (SDHCAN)
   SDHCAN monitors the vehicle CAN bus and stores relevant vehicle data in shared memory for other processes to use.
   
   SDHCAN was not modified for misbehavior detection.

4. Sensor Data Handler – GPS data handler (SDHGPS)
   SDHGPS receives time and position data from the Global Navigation Satellite System (GNSS) receiver.
   
   SDHGPS was not modified for misbehavior detection.

5. Safety Application Group and supporting processes, consisting of the Target Classification (TC), Threat Arbitration (TA) and the Safety Applications used for Warning Report generation: Forward Collision Warnings (FCW) and Blind Spot Warning / Lane Change Warning (BSW / LCW).
   
   This group of functions detects conditions which may be a threat to the HV. Alert indications are set so that the driver of the HV can be notified based on the decisions made by this group of processes.
The safety applications were not modified for misbehavior detection.

4.1.2 LMBD Application Processes

To support misbehavior detection and reporting, two new processes were created to run on the OBE:

1. Local Misbehavior Detection Control (LMBD)

The LMBD process is the heart of misbehavior detection and is responsible for logistics and control of the detection methods. LMBD performs the following operations:

- Reads the LMBD method configuration options, controls the running of the detection methods (i.e., Proximity Plausibility, Warning Report), and the timing of them running

  For the Proximity Plausibility Method, the LMBD process creates an internal list of the RVs to be processed by the algorithm. It does this by sorting the RV list according to the distance the RVs are from the HV, and then filters the list, up to a configurable maximum number of RVs, for those RVs that are within a predetermined configurable distance from the HV.

- If misbehavior is suspected and the conditions for an MBR are found (see Sections 4.2 and 4.3), it creates MBRs from the raw data created by the detection methods

- Encodes the raw data following Abstract Syntax Notation One (ASN.1) Octet Encoding Rules (OER) and packetizes it with a Hypertext Transfer Protocol (http) header

- Timestamps the report and saves it as a file in persistent memory

- If the maximum number of allowable MBRs already exist in persistent memory, the LMBD process will delete the oldest and save the newest

2. SCMS Interface (SCMS)

The SCMS process is responsible for connecting to the GMBD and transmitting MBRs. Each time it executes it does the following:

- It checks the shared memory data of DOTAP for availability of an RSE

- When in range of an RSE, the SCMS process will confirm it has one or more MBRs to send to the GMBD

- If so, it will connect to the GMBD using the IPv6 address and port supplied in the WSA and transmit the MBR

- After a response from the GMBD, the MBR file is deleted

The SCMS process also handles TCP level connection retries and checks MBR files for expiration. When an MBR expires, the SCMS process deletes the file.

4.1.3 Process Data Relationship
Figure 4 shows relationships between the data sources in the OBE and between the standard processes and new misbehavior processes.

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium

Figure 4: LMBD Software Architecture and Data Flow

4.2 Proximity Plausibility LMBD Method Design

The following sections contain an overview of the Proximity Plausibility LMBD method, followed by the high-level design logic and configuration aspects of the method.

4.2.1 Proximity Plausibility Method Overview
An HV running the Proximity Plausibility LMBD method attempts to identify a locally misbehaving vehicle by performing the checks to see if two vehicles (e.g., HV with an RV, or RV with another RV) appear to be either partially or wholly occupying the same physical space at the same time (see Section 4.2.3 for details). The HV does this by attempting to determine the location of each of its RVs at any given time and determining if any of the RVs are reporting a position that places them on top of the HV or another RV.

This is accomplished by the HV creating a virtual box around the horizontal center position of itself as well as the surrounding RVs as depicted in Figure 5. The virtual box dimensions are configurable as a percentage of length and width of the vehicle as reported in the BSM, and are smaller than the size of the vehicle. The overlap of the virtual boxes are used instead of the physical boundaries (lengths and widths) of the vehicles to account for potential errors in the reported GPS location data to limit false detections.

![Figure 5: Proximity Plausibility Virtual Box Overlap Example](image)

Figure 5: Proximity Plausibility Virtual Box Overlap Example

Figure 6 depicts an example scenario in which a proximity overlap / violation is detected. In the scenario, Vehicle 2 (V2) is depicted as having a GPS offset which makes it appear as if it is in another lane than it really is. As the vehicles drive through the scenario, V2 drives “through” vehicles V3 and V4. In this case, the algorithm increments a counter for each vehicle that is involved in an overlap with another vehicle. In this scenario, all vehicles will have violation counters incremented against vehicles V2, V3, and V4, as all are involved in instances of overlap. However, in the final scene of the scenario, the vehicles are able to assume that, even though V2, V3 and V4 all had overlap violations, V2 is likely the root cause of the overlap violations since it had multiple overlap counts associated with it due to it overlapping with each of V3 and V4. Therefore, V2 is likely the misbehaving or malfunctioning device and will be reported to the MA as a misbehaving vehicle.
4.2.2 Proximity Plausibility Logic

The Proximity Plausibility Algorithm processes pairs of reported vehicle positions for overlap. The main algorithm runs every 100 ms and performs the following:

1. Process a list of all RVs that BSMs are being received from

2. Sort this list based on the range each RV is from the HV, with RVs being closer to the HV being at the top of the list and then, given the large number of RVs that may be present in dense traffic environment, truncate this list to a maximum configurable number of RVs for further Proximity Plausibility processing

   Since Proximity Plausibility is looking for overlaps, clustering vehicles that are close to the HV, and, therefore, likely close to each other, will maximize the potential for detecting overlaps.

3. Process the RVs on the sorted list to look for overlaps between all vehicle pairs, i.e., the HV and any of the RVs and each of the RVs with any of the other RVs

4. If any overlaps are identified, add these vehicles to a misbehavior list

5. Evaluate the misbehavior list to see if the configurable proximity thresholds have been met

6. If a threshold has been met, then generate an MBR and store the MBR in non-volatile memory to await transmission to the SCMS when connectivity becomes available

---

1 Note: The thresholds and other configurable items mentioned throughout this report were varied to support the testing and available test data used in validating the proof of concept of the LMBD and GMBD methods and reporting interface. More rigorous testing and analysis would need to be performed to set the appropriate thresholds should any of the methods be further considered.
Figure 7 shows the logic flow for steps described above.

All RV List
Maintained by WMH in shared memory

Scan and sort all RVs by distance from HV
Truncate the list to the configured number of closest RVs

Sorted RV List
- Includes any RVs from the current Misbehavior List, and current sorted list from shared memory
- Re-created each time the LMBD process runs
- The max size of this list is configurable

Process RVs on the sorted list for overlaps between each other and the HV
Create/update entries on the misbehavior list when overlaps are found

Misbehavior List
Contains RVs that have at least one instance of an overlap, but have not reached the threshold for an MBR

Evaluate the Misbehavior List for vehicles exceeding overlap thresholds or expiring.
Generate MBRs when necessary

MBRs saved to non-volatile memory
4.2.3 Proximity Plausibility Method for Detecting Vehicle Overlap

Within Step 4 in Section 4.2.2, to determine if an overlap exists between the HV and an RV or an RV and another RV, the HV first checks to confirm the two vehicles are at a similar elevation and that they are roughly "near" each other based on their distance from the HV. These checks are performed very quickly based on data that already exists from the sorted RV List and does not have to be calculated. Since the balance of the algorithm is computationally intensive, the goal of this initial step is to filter out pairs of vehicles which could not possibly be in an overlap condition. Next, the positions of the remaining vehicle pairs are extrapolated to a common point in time and then evaluated with a separating-axis theorem analysis to determine if an overlap condition exists.

The separating axis theorem determines if a line can be drawn between two objects. This theorem was used because it can be performed relatively quickly, especially when the number of axes is known in advance. As it runs, once any axis is found to be a separating axis, it is then known that the two objects do not overlap and processing can stop at that point.

Each overlap instance is recorded as a single-overlap event against both vehicles. Tallies of overlap events continue until the criteria for an MBR is reached, or the time window for reaching the criteria expires. In that case, the overlap data is discarded.

Following are the steps for performing this analysis.

1. Find a pair of vehicles with reported 3D center points and heading that indicate the positions could potentially be overlapping. In this simple example, Veh 1 and Veh 2 do meet the criteria.

2. Take the reported position data and extrapolate it to current time based on the heading, speed and the difference between the current time and secMark time included in the BSM.
3. Calculate the coordinates of each vehicle's corners based on the extrapolated center point, length and width of the vehicle as well as the vehicles heading and then translate Veh 2 relative to Veh 1.

4. Then, the reported vehicle length and width dimensions are reduced by a configurable amount to produce boxes for overlap detection. Reduced-size boxes are shown below as dashed lines.

5. Using the corner coordinates of the reduced-box dimensions for input, the separating axis theorem is run on Veh 1 versus Veh 2 and Veh 2 versus Veh 1. The first separating axis found by the algorithm, if any, terminates processing. If no separating axis is found, this indicates vehicle overlap.

Following is a continuation of the example above where a separation axis does exist so there is no overlap and thus no suspected Proximity Plausibility violation.
Following is an example where a separation axis does not exist so there is an overlap condition and, depending on the configuration settings of the Proximity Plausibility Method, a suspected Proximity Plausibility violation may be triggered.

4.2.4 Proximity Plausibility MBR Contents

In this phase of the misbehavior detection, a Proximity Plausibility MBR consists of the vehicle identifiers (i.e., in the project, the BSM Part I tempID) for the vehicle suspected of causing the proximity overlaps along with the location and time of the initial proximity plausibility overlap and the final proximity overlap that triggered the generation of the MBR.

Refer to Section 3.2 Table 1 and Table 2 for the contents of the MBR triggered by the Proximity Plausibility LMBD method.

4.2.5 Proximity Plausibility Method Configuration

Following are some of the key configuration items that are used to control the behavior of the Proximity Plausibility Algorithm and trigger a suspected Proximity Plausibility Violation.

- Detect Certificate Changes: If this is set to TRUE, then an attempt will be made upon reception of a BSM with a new certificate and thus TempID, to see if this BSM is from a current RV on the all RV list or a new RV
- Maximum Proximity Plausibility RVs: This is the maximum number of RVs that will be placed on the sorted RV list for processing by the Proximity Plausibility Algorithm
• Elevation Threshold: If the elevation difference between two BSMs, from a vehicle pair, exceeds this threshold they are filtered out and not considered for Proximity Plausibility processing between that vehicle pair

• Proximity Boundary Length: The ratio of the vehicle length that will be used to construct the length of the boundary box that is placed around the vehicle center

• Proximity Boundary Width: The ratio of the vehicle width that will be used to construct the width of the boundary box that is placed around the vehicle center

• Proximity Operation Interval: The interval of time, starting with an initial vehicle pair overlap, over which other vehicle pair overlaps, if any, will be considered for multi-vehicle overlap configurations

• Proximity Operation Window: The interval of time which an individual vehicle pair overlap will be monitored

• Proximity Overlap Count Threshold: The number of proximity overlaps, for a given vehicle pair, that must be identified before a proximity violation will be declared for that vehicle pair

• Proximity Violation Threshold: The number of unique vehicles a vehicle must be declared as overlapping (via the Proximity Overlap Count Threshold logic) before that vehicle will be declared as a suspected misbehaving vehicle and an MBR generated against that vehicle

• Proximity Report Expiry Time: The time after which a Proximity Plausibility MBR is considered too old to report, and thus is deleted from non-volatile memory

4.3 Warning Report Design

The following sections contain an overview of the Warning Report LMBD method, followed by the high-level design logic and configuration aspects of the method.

4.3.1 Warning Report Method Overview

The Warning Report LMBD method generates an MBR when information from one or multiple RVs causes the safety applications within the OBE / HV to trigger a warning. In this project, the Warning Report Method was applied to two safety applications; FCW and BSW / LCW. Because generating MBRs on every warning event might cause excessive amounts of MBRs to be transmitted to the SCMS, this method has additional logic built in to reduce the potential number of MBRs generated. To do this, the algorithm can be configured to consider a combination of the:

• Number of warnings triggered over a predefined time window

• Number of unique devices causing warnings within the same time window

• Driver reactions to the warnings

Considering these, the algorithm is able to reduce the number of MBRs generated because it can infer the validity of the true positive warnings and not generate MBRs and infer suspicion about the real false positive warnings. Table 5 shows the possible combinations of the number of warnings, number
of devices causing the warnings, and if driver reaction is considered and a corresponding mapping to the level of MBRs that would be generated and the level of misbehavior suspicion (i.e., the likelihood that this is a misbehaving device) that may be applied to the suspected misbehaving device. Note that these are conceptual levels to begin to understand the potential tradeoffs between the different method configurations.

**Table 5: Warning Report Method Concept**

<table>
<thead>
<tr>
<th>Events</th>
<th>Driver Reaction Not Considered (DRNC)</th>
<th>Driver Reaction Considered (DRC)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of MBRs</td>
<td>Misbehavior Suspicion</td>
</tr>
<tr>
<td>A single warning is triggered by a single device (SWSD)</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Multiple warnings are triggered by a single device (MWSD)</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Multiple warnings are triggered by multiple devices (MWMD)</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium

The Warning Report Method currently evaluates the driver reaction through two primary vehicle dynamics data which are deceleration and yaw rate of the vehicle. Deceleration and yaw rate are used for detecting driver reaction during FCWs and yaw rate is used for detecting driver reaction during BSW / LCWs. For example, when an FCW warning is triggered, the method looks for either sufficient deceleration values to indicate an attempt to slow the vehicle to avoid the collision, or sufficient yaw rate value to indicate an attempt to swerve to avoid the collision. These are evaluated before or after the warning. If either the deceleration or yaw rate exceeds a configurable predefined threshold during the configurable predefined time windows before or after the warning occurs, the method will register a driver reaction and not generate an MBR.

In case the system triggers multiple warnings within a predefined time window, the method will also evaluate the uniqueness of the RVs causing the warnings. If the method determines that the multiple warnings are being triggered by a unique device, the method will reduce the number of MBRs and only generate one MBR for that given time window. As an example, if an RV causes an FCW and a BSW / LCW within a configurable predefined time window, the method will generate only one MBR instead of two MBRs.

**4.3.2 Warning Report Logic**

The goal of Warning Report detection is to detect RVs that are suspected of causing false or erroneous V2V safety application warnings in the HV. As shown in the Table 5, an MBR occurs when a configurable number of RVs cause a configurable number of warnings in the HV. To determine this,
the Warning Report Algorithm keeps a record of all the RVs that cause either of the two types of warnings and the number of warning events caused by each RV during a configurable time window. A typical BSW / LCW or FCW will last through many computation cycles, and a total event time of 1 to 2 seconds, and more, is not unusual. The Warning Report Algorithm runs every 100ms, and each time it runs it performs the following:

1. Reads TA shared memory for an indication that an FCW or BSW / LCW warning flag is set
   Using this flag, the algorithm knows when a warning begins, if it continues to be active, and if the warning has completed.

2. If a warning flag is set at the beginning of the warning event, the ID of the RV causing the warning is recorded into the internal warning buffer, if the RV has not already been added to the buffer

3. If this is a new warning event for this RV, then the warning information is also added to the buffer, otherwise, the warning event information (e.g., location, time) is updated

4. The warning continues to be monitored each time the algorithm executes, until the flag is no longer set
   - Once it is no longer set, if it does not return to active for that RV for a configurable period (e.g., 500 ms), the algorithm will declare the warning complete
   - If the same warning occurs again for the same RV before the end of the configurable waiting period, it is considered a continuation of the original warning
   - Once complete, the warning count total for that RV is incremented

5. The warning buffer is then processed against the Warning Report MBD threshold and other criteria to determine if misbehavior is suspected, e.g.,
   - Driver reaction considered
   - Number of warnings
   - Number of RVs causing warnings

6. If a threshold has been met, then an MBR is generated and stored in non-volatile memory to await transmission to the SCMS when connectivity becomes available

Figure 8 shows the logic flow for steps described above.
Chapter 4: LMBD Design and Implementation

Flowchart:

Start

Is a Warning Flag Set?
  Yes
  Is vehicle already in the warning buffer?
    No
    Create a vehicle entry in the warning buffer
    Yes
    Does this warning already exist for this vehicle?
      No
      Add new warning information for this vehicle
      Yes
      Update this warning information for this vehicle
  Check for end time of active warnings
  Process the warning buffer against the MBD criteria
  Misbehavior Suspected?
    No
    Remove suspected vehicle from the warning buffer
    Yes
    Generate an MBR

MBRs saved to non-volatile memory

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium

Vehicle Safety Communications 6 (VSC6) Consortium Proprietary
4.3.3 Warning Report Processing of Driver Reaction

One of the configurable options for the Warning Report Method is considering HV driver reaction to the warning to see if the warning should be considered valid or not. If configured for HV driver reaction and if a HV driver reacts during a warning event, that warning is removed from the list of warnings for that RV. If there are other warnings for that RV without HV reactions, those warnings remain on the list and the processing of that RV continues.

As discussed in Section 4.3.1, for the safety applications selected for this project, the determination of the HV driver’s reaction is based on the HV deceleration and/or yaw rate, with a configurable threshold for each. To be considered valid and to ensure that a reaction is observed relative to a warning, the algorithm processes HV reaction data during a configurable window of time prior to the event (e.g., 500 ms), and continues to process reaction data during a configurable window of time after the warning event completes (e.g., 2 seconds). The procedure that is performed is as follows:

1. To preserve pre-event reaction data, the configurable duration of reaction data is continually buffered, but not processed, until a warning event begins. At that point, the buffered pre-event data is set aside to be processed after the warning event completes, depending on the outcome of the following steps.

2. While the warning event is active, each time the algorithm runs it checks the most recent HV driver reaction data (obtained from the CAN bus data) for a reaction. If the appropriate driver reaction for the warning is detected, reaction processing stops.

3. If there was no reaction in active event data, once the warning event ends, reaction data continues to be processed until the end of the post-event data period. The purpose is to be able to observe a reaction that might occur after the conditions that created the warning event are no longer active. If the appropriate driver reaction for the warning is detected, reaction processing stops.

4. If there was no reaction in the post-event data, the pre-event data is processed for a driver reaction. If the appropriate driver reaction for the warning is detected, reaction processing stops.

5. If there was no reaction in the pre-event data, the warning is retained in the warning buffer and counted toward the MBR threshold.

For any of steps 2 - 4, if a HV driver reaction is observed and the driver reaction processing stops, the warning is still processed until it finishes, but it is considered a valid warning and not a case of misbehavior. Once the warning ends, it is removed from the warning buffer and not counted toward an MBR threshold.

4.3.4 Warning Report MBR Contents

In this phase of the misbehavior detection, a Warning Report MBR consists of a list of RV identifiers (i.e., in the project, the BSM Part I tempID) for those RVs that were involved in meeting the criteria for the MBR. In the case of a SWSD or MWSD MBR, this is a single RV ID. In the case of an MWMD MBR, there will be multiple RV IDs in the MBR (See Table 3).

Note that the warning algorithm does not record the warning type. As each warning event completes (BSW / LCW or FCW), the algorithm will simply increment the count of events against that vehicle.
a multiple warning configuration, a vehicle could conceivably have created warning events of both types.

Refer to Section 3.2, Table 1 and Table 3 for the contents of the MBR triggered by the Warning Report LMBD Method.

4.3.5 Warning Report Method Configuration

Following are some of the key configuration items that were put in place to control the behavior of the Warning Report Algorithm and determine what is required to trigger a suspected Warning Report Violation.

- Operation Mode: The combination of number of warnings and / or number of RV causing a warning that should be considered for generating a MBR (i.e., SWSD, MWSD, MWMD from Table 5)

- Driver Input: Enables or disables the consideration of driver reaction to the warning to see if the warning should be considered valid or not (Note: this can be enabled / disabled separately for FCW and BSW / LCW)

- Driver Input Pre-Warning Window: The detection window / period before the warning in which the driver’s reaction, if any, is considered

- Driver Input Post-Warning Window: The detection window / period after the warning in which the driver’s reaction, if any, is considered

- Yaw Rate Threshold: Indicates the minimum absolute value of change in yaw rate required to indicate a driver reaction to the warning (Note: this can be configured separately for FCW and BSW / LCW)

- Deceleration Threshold: Indicates the minimum deceleration required to indicate a driver reaction (Note: this is only considered for FCW)

- Warning Detection Window: For multiple warning operation modes, the period after an initial warning to monitor for additional warnings

- Warning Occurrence Threshold: The number of suspicious warnings (i.e., when considering driver reaction, if configured) that must occur before an MBR will be generated (Note: this can be configured separately for MWSD and MWMD operational modes)

- RV Threshold: The number of unique RVs triggering a suspicious warning (i.e., when considering driver reaction, if configured) before an MBR will be generated (Note: This is only considered for the MWMD operation mode)

- Warning Report Expiry Time: The time after which a Warning Report MBR is considered too old to report and thus is deleted from non-volatile memory


5 GMBD Design and Implementation

GMBD analyzes the MBRs received from the OBEs / vehicles and processes the MBRs to identify misbehavior and determine which device(s), if any, is the root cause of the misbehavior. The following sections provide more specific details on the components of the GMBD server that were implemented and / or utilized in the project. For the GMBD methods, the V2V-Interoperability Project identified three potential methods for consideration:

- Device – For single-event misbehavior observed by several reporters
- Event – For multiple-event misbehavior that takes place within the same region and time window
- Feature – For advanced misbehavior involving data or other reporting manipulation

For this project, the primary ones selected for implementation were the Device and Event Methods while the framework was put in place for the Feature Method. These are discussed in more detail in Section 5.3. As discussed in Section 1.3, the methods implemented in this project were not developed to a deployment-ready level rather they were developed to a level to begin understanding the viability of the methods.

5.1 GMBD Server Network and Component Distribution

The components of GMBD are divided between an Application Server and a Database Server. For the purposes of the GMBD Project, a single Application Server and single Database Server were used. However, the system has been designed (but not functionally tested) to scale the Database Server to a server cluster to handle larger volumes of data. Figure 9 describes distribution of the key software components, highlighted in blue, on the server network and the high-level interactions between the software components.
Figure 9: GMBD Components and Workflow Overview

The primary components for GMBD on the Application Server are the MBR Receiver, Threshold Manager, and the Linkage Information Manager (LIM) Service. The Application Server hosts an Apache Karaf™ environment with an Apache Camel™ integration framework in which Open Source Gateway Initiative (OSGi) containers run to execute the components. The Database Server hosts the Repository Manager component which includes the Apache Oozie™ workflow scheduler, an Apache Hbase® distributed big data store, and an Apache Spark™ engine. Following are descriptions of these components.

5.1.1 MBR Receiver

The MBR Receiver exposes a publicly facing Representational State Transfer (REST) service to receive MBRs from vehicles. This component validates the contents of the MBR and stores the MBR in the Data Repository.

5.1.2 Threshold Manager

The Threshold Manager is the main component of the GMBD application. This component:

---

2 Apache Karaf, Apache Camel, Apache Oozie, Apache HBase, and Apache Spark are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries. No endorsement by The Apache Software Foundation is implied by the use of these marks.
• Configures work flows to be executed by the Repository Manager. There are separate workflows for each GMBD method, in some case multiple work flows per method.
• Retrieves query results from Repository Manager
• Determines if results of any periodic query produce a potentially revocable suspect group of linkage values
  o If found, this component queries LIM to determine if certificates can be correlated to a single entity enough times to meet revocation threshold
  o If the revocation threshold is met, the identity of the offending device is added to the Revocation List

5.1.3 Linkage Information Manager (LIM)
The LIM is a service that is used by the Threshold Manager. The LIM:

• Correlates linkage values within a suspect group to the identity of the End Entity (EE) device (e.g., OBE) that has the certificates containing the linkage values
• If enough linkage values in the suspect group correlate to the same EE device, the component provides the identity to the Threshold Manager as well as the total number of correlated linkage values found in the suspect group

5.1.4 Repository Manager
The Repository Manager:

• Stores the MBR data
• Receives work flow configurations from the Threshold Manager
• Executes simple queries against the database based on the configured workflows
• Returns query results to the Threshold Manager

5.2 Process Flow
Figure 10 shows the overall process flow for GMBD with a description of each process step provided in the following sub-sections.
5.2.1 MBR Receipt

Figure 11 shows the MBR Receipt process. The MBR Receiver accepts an ASN.1 encoded MBR message via the exposed REST service. The MBR Receiver uses the ASN.1 Encode/Decode library to de-serialize the message. The MBR Receiver then stores the contents of the de-serialized message in the database.
5.2.2 Workflow Scheduling

Workflows in the context of the GMBD implementation are the steps specific to each GMBD algorithm (i.e., Device, Event, and Feature). The workflows are predefined algorithms executed on the Database Server orchestrated via the Apache Oozie™ workflow scheduler. Workflows generally involve multiple queries against the database, data transformations, and logical evaluations leading to the next query steps. The final results of the workflows are saved in a results table in the Apache HBase® data store. The Threshold Manager schedules the execution of the workflows and continuously executes the workflow schedules until shutdown. The workflows are executed by the Threshold Manager triggering the workflows via an Apache Oozie™ REST service.

Figure 12 shows the Workflow Scheduling process. For details on the GMBD method implementation contained in the Workflow Scheduling, refer to Section 5.3.

5.2.3 Query Results Retrieval

The workflows run asynchronously from the Application Server processes on the Database Server. The Threshold Manager polls the Apache Oozie™ services to determine when a workflow has completed. When Apache Oozie™ replies to the polling that a workflow is complete, the Threshold
Manager directly queries the Apache HBase® results table to retrieve the workflow results as depicted in Figure 13.

![Figure 13: Query Results Retrieval](image)

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium

**Figure 13: Query Results Retrieval**

### 5.2.4 Suspect Group Interrogation

The workflow results pulled from Apache HBase® results table are the final output of a GMBD method algorithm. This output represents a suspect group of linkage values. In a deployed system, the linkage values will come from reported certificates used by EEs during suspected misbehavior that meet the criteria of the detection algorithm. The detection algorithms are designed to attempt to find certificates that may have been involved in a single or related misbehavior. However, the results of these algorithms do not prove that the certificates are in fact related. It is possible that the certificates detected by the algorithm are simply individual instances of atypical behavior and not consistent misbehavior by a single (or coordinated group) of entities.

To determine if the linkage values in the suspect group do originate from the same EE, the suspect group is interrogated by the LIM. After the Threshold Manager receives the suspect group, it calls the LIM. The LIM correlates the linkage values to EE identities. The LIM then counts how many linkage values correlate to the same EE identity. The LIM returns any EE identity that correlates to more than the minimum global GMBD revocation threshold along with a single representative linkage value that correlated to that identity and a count of the number of linkage values in the suspect group that correlated to that identity.

In this implementation, a pseudo-anonymized vehicle ID was used rather than using certificates for the linkage operations, since the MBR contained BSM metadata rather than complete and signed BSMs. This ID was encoded in the linkage value which simplified the testing and EE identification, making correlation trivial. Integration into the SCMS MA would require replacing this correlation with the SCMS investigation process involving communications with other SCMS components. This is the primary reason the LIM was separated into its own software component even though in this implementation it preforms simplified processing.

Refer to Figure 14 for the logic involved in the Suspect Group Interrogation process.
Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium

**Figure 14: Suspect Group Interrogation**
5.2.5 Revocation

The Threshold Manager examines the results of the LIM investigation. For any EE identity existing in the results, the Threshold Manager will compare the correlated linkage value count against the revocation threshold for the GMBD method which produced the suspect group. If the count is greater than the revocation threshold, then the Threshold Manager will initiate revocation procedures.

In this implementation, since the GMBD development was not fully integrated into the MA component of the SCMS, the revocation procedure is simply adding the vehicle identity to a revoked vehicle list. Therefore, no logic is provided.

5.3 GMBD Methods

Section 5.2.2 discusses the workflow scheduling which runs in the context of each of the GMBD methods. The detection methods are the algorithms which scan the MBR database to find groups of MBRs that are suspected of coming from the same or coordinated EEs. The result of these detection methods is the linkage values that make up the Suspect Group inspected by the LIM.

In the Project, three types of detection methods were implemented. Each of the current three detection methods uses a different approach to detecting misbehavior and has its own strengths and weaknesses. All the approaches are used together to provide a multi-layered net to detect misbehavior. While some types of misbehavior may not be caught by one detection method they may be caught by another and vise-versa. The following sections provide a high-level overview of each of the methods and the method processing.

5.3.1 Device-based GMBD Method

The Device-based Threshold Method was selected as the initial GMBD method for development as it was deemed as the most easily testable. Device-based Detection is the simplest of the misbehavior detection methods and likely to catch only the least sophisticated attacks and misbehaviors. Even though the detected attacks and misbehavior types are the least sophisticated, these are the types that may be the most common, and, thus, the method adds high value.

This detection method scans the database of received MBRs to count the number of MBRs reporting the exact same offending linkage value. If any linkage value totals more than the revocation threshold, the linkage value is submitted for revocation. The results of this detection method do not need to be submitted to the LIM as there is no need to correlate the linkage values since they are all the same.

As EEs only use the same certificate (and thus the same linkage value) for a short period of time, this detection method is designed to catch single-event misbehavior observed by several reporters. This will likely be a common scenario for unintended misbehavior (i.e., faulty) EEs or crudely designed and implemented attacks.

5.3.2 Event-based GMBD Method

Event-based detection is designed to catch misbehavior and attacks that take place in the same physical location and during a time window (e.g., minutes, hours, days), even if the misbehaving device(s) uses multiple pseudonym certificates for the misbehavior. This detection method first scans the database of MBRs passing each MBR through a Counting Bloom Filter (CBF). This filter creates counts for how many MBRs in the database report misbehavior that occurred in the same time.
windows and in the same geographic area. The result of this filter is a series of “bins” that each represent a specific time and geographic location combination.

Figure 15: Event-based GMBD Method – Counting Bloom Filter Example Diagram

Figure 15 illustrates the concept of a CBF and how misbehavior with different pseudonym certificates is binned. For example, in Figure 15 a bin may represent all events happening within a specific square mile (or larger) area between a specific time and on a specific date. After the filter processing is complete, each bin contains the count of how many MBRs correspond to that bin’s time and location. The steps to revoke the pseudonym certificates are as follows:

1. The system scans the list of bins to find bins that have counts higher than the configurable revocation threshold for Event-based Detection
2. The system then scans the database and extracts all MBRs associated with the bins with high counts. The assumption is that all of the MBRs found to associate with the same time and location bin are likely involved in the same or tightly related misbehavior.
3. The system extracts the Linkage Values (LVs) from the MBRs and this LV set becomes the Suspect Group to be examined by the other entities of the SCMS (e.g., MA, RA) as part of the misbehavior investigation process of the SCMS. Note, since this development was not integrated within the full SCMS, these entities were mimicked in the implementation.
4. If enough linkage values in the suspect group correlate to the same misbehaving device, the device is revoked from the system.
5.3.3 Feature-based GMBD Method

Feature-based Detection is designed to catch advanced attacks and misbehaviors. This method is best thought of as a framework on which many detection algorithms can be built. New instantiations of this method can be created by adjusting the type of parameters (usually specific fields in the MBR) that the algorithm processes allowing the method to detect different types of attacks.

In an instantiation of this method, specific factors of the MBR are selected and given weights. These factors are referred to as dimensions. The dimensions selected for the instantiation implemented in this project were the neighbor count, time, longitude, and latitude fields of the MBR. The MBRs are then clustered based on relative similarity in dimensions via a K-Means clustering algorithm. The result of this algorithm is groups of MBRs called clusters that share relatively similar dimensions.

These clusters are examined to ensure they are of high quality, meaning that they show truly distinct groupings. High-quality clustering is recognized when the factors / parameters selected from the MBRs for analysis within a cluster are in close agreement with each other (i.e., difference between the values are not too spread out) while the individual clusters are not too close to each other. If the clusters are not of high quality, then the algorithm did not find anything suspicious as the MBRs look unrelated. If the clustering is of high quality, then the clusters can be compared and examined in different ways depending on what type of misbehavior the instantiation is attempting to catch.

To prove the viability of the concept of this method, for this project's instantiation, the misbehavior targeted is an attack where a misbehaving device reports misbehavior with fraudulent neighbor count in order to delay or accelerate the revocation since neighbor count impacts misbehavior detection and revocation. As an example, consider a misbehaving vehicle (red vehicle in Figure 16) that has offset its position such that it triggers a series of Proximity Plausibility Reports against it (by the blue vehicles in Figure 16). The red misbehaving vehicle will send its own Proximity Plausibility Report, potentially falsely accusing one of the other vehicles, but with an inflated vehicle count in the hopes that GMBD will delay revoking it.
Note that in this account, one of the reporters of misbehavior is actually the attacker. The Feature-based GMBD Method will cluster the neighbor counts by the geographical area as shown in Figure 17. In this instance, the misbehaving vehicle neighbor count value would result in a small cluster versus the larger neighbor count cluster of the other non-misbehaving vehicles. This could then lead to further investigation of this vehicle.

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium

Figure 16: Feature-based GMBD Method – Neighbor Counting Example
Figure 17: Feature-based GMBD Method – Neighbor Count Clustering Visualization

Note that in the above example, one of the reporters of misbehavior is actually the attacker. This can be determined by examining the clusters and seeing if any cluster stands out as being much smaller than its peers. If a relatively small cluster is found, it is suspected to represent misreporting of neighbor count (Note: this is dependent on there being enough other devices in the area of the event also reporting the misbehavior event to produce comparable large clusters to the fraudulent cluster). Linkage values are then extracted from the reporters of MBRs in the small cluster and investigated by the LIM to confirm they correlate to the same offending device.

Note that in the project, the framework for Feature-based Detection was implemented and tested with the simple attack described above. However, there is potential for devising more sophisticated dimension and weighting schemes for the Feature-based Detection Method instantiations to catch other attacks.

5.4 GMBD Configuration

Following are some of the high-level configuration items that were put in place and control the behavior of the GMBD methods and what is required to trigger a revocation. The values used varied depending on the test being run and available test data.

- Geo-temporal Bin Size: The size in meters of a grid square for analysis for the Event-based and Feature-based GMBD Methods (note that each had their own configuration item)
- Geo-temporal Time Window: The duration of the analysis window for the Event-based and Feature-based Methods (note that each had their own configuration item)
- Query Threshold: The number of MBRs that must be received and believed to potentially link to the same device before the LIM will be queried to see how many of the certificates (i.e., tempID in this project) correlate to the same device (note that different configuration items existed for each of the GMBD methods)
- Revocation Threshold: The number of MBRs that must link to the same device before the device can be revoked
- Unique Reporter Threshold: The number of unique reporters, within a LIM MBR query bundle, to verify the uniqueness / variability of the reporting devices before revoking a device
6 Test Data and Testing

The testing of the LMBD and GMBD methods was performed in the CAMP VSC6 Lab and utilized a combination of OBEs on a bench communicating, via an RSE installed in the lab, with the GMBD Servers located at the CAMP office. The following sections discuss one of the primary sources of test data as well as a summary of the tests and test results. As a reminder, the methods implemented in this project were not stress tested, tested in the real world, nor developed to a deployment-ready level. The algorithms were tested to validate the SW implementation and begin testing the viability of the conceptual implementation of the algorithms. Therefore, this section provides a discussion on the areas of testing rather than providing specific test results.

6.1 Test Data

The initial SW functionality for the LMBD included the support for creating dummy test MBRs (see Section 3.2, Table 4) encoding the MBRs with OER and transmitting the MBR to the GMBD Server in order to test the interface between the OBE and the GMBD Server. To do this, the OBE was required to listen for WSAs, broadcast from the lab RSE, and use the information within the WSAs (e.g., IP address, port) to establish a connection with the GMBD Server. Once connection was established, the OBE transmitted the MBRs through the RSE to the Server. The initial SW release for GMBD required the system to be able to receive, decode and store the MBRs.

Then, to support the initial testing and proof of concept of the LMBD methods, it was desirable to use data representative of what may be observed in the real world. Thus, Virginia Tech Transportation Institute (VTTI) was selected to assist with utilizing the data from USDOT SPMD Test Bed which was located in Ann Arbor, Michigan. VTTI was selected due to their involvement with the CAMP VSC3 Vehicle-to-Vehicle Safety System Light Vehicle Builds and Model Deployment Support (V2V-MD) Project. In that project, VTTI installed a Data Acquisition System (DAS) into each of the CAMP VSC3 Participant-built Integrated Light Vehicles (ILVs) which were deployed in the SPMD Test Bed. The DAS logged metadata from the HV transmitted BSMs as well as the BSMs received by RVs encountered while driving in the Test Bed.

The ILVs, along with the greater than 2,000 deployed DSRC Vehicle Awareness Devices (VADs) and Aftermarket Safety Devices (ASDs), also logged Packet Capture (PCAP) data for every transmitted message which contained the fully encoded and signed BSM contents. The initial plan was to use the PCAP data logs collected on all of the OBEs in the SPMD in combination with the ILV logged interactions with these devices to generate sets of PCAP files that, when re-played over-the-air, would attempt to recreate those interactions. It was determined that the PCAP files were not available for use, thus, an alternate approach was required. The approach selected was to use the HV and RV information logged in the DAS on each of the ILVs for this purpose.

Utilizing the SPMD data was desirable because, based on previous analysis, it was known that there were instances of non-ideal hardware installation as well as incorrect SW configuration on some of the VADs and ASDs. Numerous ILV safety application false warnings were attributed to these installation and configuration errors. Given the nature of many of these false warning, the Proximity Plausibility and/or Warning Report LMBD Methods should flag them as misbehavior. Therefore, the Team
worked with VTTI to identify ILV interactions with VADs / ASDs that the Team believed would have triggered the detection of misbehavior at the local level.

Throughout the V2V-MD Project, VTTI compiled all the DAS logs into a database for archive and analysis. For the Project, VTTI first compiled a list of the VADs / ASDs that were deemed as the top offenders of causing warnings within each ILV under circumstances when a warning was not warranted (e.g., data received from an RV was inaccurate). This list also included the corresponding warning events and the time when the events occurred. An excerpt of this list is shown in Table 6 where the HV ID identifies the ILV vehicle IDs and the RVID identifies the IDs of the various RV VADs / ASDs that caused the warnings on those ILVs.

**Table 6: VAD / ASD Top Offenders Event List Excerpt**

<table>
<thead>
<tr>
<th>File ID</th>
<th>HV ID</th>
<th>RV ID</th>
<th>Alert Type</th>
<th>Alert Time Stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1717875</td>
<td>14</td>
<td>17101</td>
<td>FCW</td>
<td>2013/05/16-20:28:50.100</td>
</tr>
<tr>
<td>1807912</td>
<td>42</td>
<td>17101</td>
<td>FCW</td>
<td>2013/05/08-17:08:28.900</td>
</tr>
<tr>
<td>1592532</td>
<td>43</td>
<td>17101</td>
<td>FCW</td>
<td>2013/05/01-13:32:31.500</td>
</tr>
<tr>
<td>1589457</td>
<td>62</td>
<td>17101</td>
<td>FCW</td>
<td>2013/04/08-15:17:40.200</td>
</tr>
<tr>
<td>1726652</td>
<td>65</td>
<td>17101</td>
<td>FCW</td>
<td>2013/05/04-02:13:44.800</td>
</tr>
<tr>
<td>2320234</td>
<td>66</td>
<td>17101</td>
<td>FCW</td>
<td>2013/07-29-19:50:16.500</td>
</tr>
<tr>
<td>1617688</td>
<td>87</td>
<td>17101</td>
<td>FCW</td>
<td>2013/04/03-14:03:53.100</td>
</tr>
<tr>
<td>1718426</td>
<td>14</td>
<td>7165</td>
<td>FCW</td>
<td>2013/05/14-21:33:20.900</td>
</tr>
<tr>
<td>1972297</td>
<td>16</td>
<td>7165</td>
<td>FCW</td>
<td>2013/06/24-22:18:50.400</td>
</tr>
<tr>
<td>1530766</td>
<td>32</td>
<td>7165</td>
<td>FCW</td>
<td>2013/03/26-04:07:47.400</td>
</tr>
<tr>
<td>1783966</td>
<td>32</td>
<td>7165</td>
<td>FCW</td>
<td>2013/06/22-23:01:31:500</td>
</tr>
<tr>
<td>1835067</td>
<td>62</td>
<td>7165</td>
<td>FCW</td>
<td>2013/05/01-15:11:15:600</td>
</tr>
<tr>
<td>1532521</td>
<td>32</td>
<td>7178</td>
<td>FCW</td>
<td>2013/04/17-22:30:23:800</td>
</tr>
<tr>
<td>2189330</td>
<td>41</td>
<td>7178</td>
<td>FCW</td>
<td>2013/07/22-14:22:300</td>
</tr>
<tr>
<td>1633324</td>
<td>86</td>
<td>7178</td>
<td>FCW</td>
<td>2013/04/16-19:31:38:300</td>
</tr>
<tr>
<td>2382835</td>
<td>34</td>
<td>7267</td>
<td>FCW</td>
<td>2013/07/26-16:50:14:900</td>
</tr>
<tr>
<td>1808516</td>
<td>44</td>
<td>7267</td>
<td>FCW</td>
<td>2013/04/01-11:42:25:000</td>
</tr>
<tr>
<td>1448802</td>
<td>47</td>
<td>7267</td>
<td>BSW</td>
<td>2013/03/28-14:15:51:600</td>
</tr>
<tr>
<td>1741380</td>
<td>51</td>
<td>7267</td>
<td>FCW</td>
<td>2013/06/03-11:52:27:500</td>
</tr>
<tr>
<td>2251834</td>
<td>44</td>
<td>3052</td>
<td>BSW</td>
<td>2013/08/13-11:57:20:400</td>
</tr>
<tr>
<td>2251834</td>
<td>44</td>
<td>3052</td>
<td>BSW</td>
<td>2013/08/13-11:57:32:000</td>
</tr>
<tr>
<td>2019282</td>
<td>67</td>
<td>11129</td>
<td>FCW</td>
<td>2013/07/10-18:18:15:900</td>
</tr>
<tr>
<td>2019282</td>
<td>67</td>
<td>11129</td>
<td>FCW</td>
<td>2013/07/10-18:19:36:800</td>
</tr>
<tr>
<td>2258897</td>
<td>67</td>
<td>11129</td>
<td>FCW</td>
<td>2013/07/10-20:28:06:000</td>
</tr>
</tbody>
</table>

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium
After identifying the list of relevant warning events that are likely to trigger the Proximity Plausibility and/or Warning Report LMBD Methods, VTTI then performed a series of steps depicted in Figure 18 to extract and transform the Event Data into several file formats to support the testing activities. The ILV Event Data Extraction Process involved pulling data from the ILV DAS Database and SPMD Vehicle Configuration Data File. This Event Data was then transformed into a CAMP VSC6-defined Comma Seperated Value (CSV) file. The Event Data CSV file included all the necessary BSM data from the RV identified in the top offenders table, other RVs that were within communication range at the time, as well as the necessary BSM data of the HV.

Source: Crash Avoidance Metrics Partners LLC (CAMP) Vehicle Safety Communications 6 (VSC6) Consortium

**Figure 18: V2V-MD DAS Data Export Process Flow**

After the data export package was generated, during the OBE Replay/Test and Report to GMBD process step, each CSV was then replayed in the OBE to test both LMBD methods to see if one or both of the methods detected misbehavior and, if so, generate and send a report to GMBD. In addition to this Event List and the Event Data CSV files, VTTI also created Event Charts depicting the relevant post-processed data, such as the location of the warning event on a map, relative distances between the RV and HV, as well as frame captures of the recorded video corresponding to the alert event timespan of interest within these CSV files. Figure 19 shows an example Event Chart exported for an FCW event. The ‘0’ Time to Warning (ms) value in right most charts represents the moment in which the FCW event occurred.
Figure 19: V2V-MD DAS Exported Chart Relating to an FCW Event

The data export corresponding to Figure 19 was an example data set used to test the Warning Report LMBD Method as well as the Proximity Plausibility LMBD Method. It represents an example in which an FCW warning is triggered in an ILV as it drives past the RV in the adjacent lane which implies that the ILV and RV must have overlapped in time and space to a certain degree. These data were used to support the Team’s interpretation of the events so they could refine and evaluate the performance of the LMBD methods.

6.2 LMBD Testing and Validation

The primary data used for testing the LMBD methods were the CSV files generated from the process described in Section 6.1. Because the data used for testing was of specific examples of prerecorded and uncontrolled real-world driving scenarios, LMBD testing was a manual and iterative process. To reduce the amount of testing time, a small subset of data was used to test the Warning Report and Proximity Plausibility LMBD Methods. This smaller subset was used, along with a combination of method configurations, to prove the viability of the method concepts. The reasons such an approach was chosen is because there was no effective and efficient way of predetermining precisely how the vehicles in the real world interacted with regards to the overlaps, the amount of time the overlaps occurred or the exact types of driver reactions to name a few examples.
The testing focused on validating the method implementations and general viability of the methods and included tests which validated:

- **Proximity Plausibility**
  - RV filtering, to limit the number of RVs that are processed
  - Elevation filtering, to remove from consideration vehicles that are at different elevation (e.g., due to bridges crossing roads)
  - Overlap detection
  - Minimum to maximum boundary box thresholds, which are configured to be a percentage of the length and width of the vehicle reported in the BSM
  - Minimum to maximum overlap duration between two vehicles
  - MBR generation

- **Warning Report**
  - Detecting and considering or not considering driver reactions based on the configuration
  - Single warning versus multiple warnings from a single device
  - Time related thresholds for considering driver reactions, considering multiple warnings, etc.
  - MBR generation

Note that given the limited ILV data available for testing, the aspects of the methods that require more than two RVs to be involved in potential misbehavior could not be tested. For Proximity Plausibility, this included the aspect of identifying the misbehaving vehicle by observing overlaps of multiple vehicles (more than 2) over a given time and identify the vehicle with the most overlap count. For Warning Report, this included the method supporting detection of multiple warnings from multiple vehicles in a given period of time. These aspects were tested and validated, however, as part of the code development with unit tests and contrived data.

### 6.3 GMBD Testing and Validation

The test data used for validating the GMBD methods were the MBRs generated by the LMBD methods. Since the LMBD methods were tested with the data export as detailed in Section 6.1, the GMBD methods also used the same data, except they were abstracted into the MBR format as described in Section 3.2.

Similar to the LMBD methods, the GMBD methods were tested to validate the logical implementation of the methods. And, thus, the methods were tested with minimum and maximum thresholds to verify if the methods are able to detect misbehavior.
Device Based being the simplest method, it was implemented and tested first. This method relied on multiple MBRs accusing one specific linkage value of misbehavior. Once the MBRs accusing a specific LV of misbehavior exceeded a specific threshold (e.g., 2 or 20), the Device-based Method was able to identify the LV for revocation. By modifying the thresholds at the LMBD to generate more or fewer MBRs and/or by changing the GMBD thresholds, the method was tested and it was verified that misbehavior is detected by the Device-based thresholding method.

For initial testing, the inputs used for the Event-based Method Testing were Proximity Plausibility MBRs that were generated based on three (3) independent vehicle proximity violations that occurred very closely to each other in the SPMD environment on April 16, 2013, April 17, 2013 and July 12, 2013. Using these reports as input into the Event-based Method, the CBF as well as the general misbehavior detection functionality could be tested to see if the CBF thresholds were exceeded by verifying if the MBRs were stored in specific bins/buckets within the CBF. Similar to the tests in Device-based, this was tested by modifying the configuration thresholds / parameters of the CBF to make the geo-temporal bins larger or smaller. The testing validated that if the configurable sizes of the geo-temporal location and time bins were small (smaller resolution), then the 3 events (with multiple reports) fell into three separate bins without exceeding the configurable threshold in any specific bin and thus no misbehavior was determined, as expected. If the geo-temporal configuration parameters were configured to be sufficiently large, then the 3 events fell into the same bin and exceeded the threshold and thus misbehavior was determined, as expected.

Since only the framework for the Feature-based Method was developed and implemented and given the limitations on the availability of test data, this method was not tested in detail.
7 Summary

The Project implemented a lab environment in which an OBE was able to use an RSE to communicate with the SCMS using IPv6. The RSE was used to transmit a WSA which advertised the availability of misbehavior reporting service and the IP address of the server. Due to the existing OBE supporting the USDOT SPMD SCMS interface rather than the updated SCMS PoC interface, the communication protocol for the OBE to send a MBR to the SCMS was simplified to use an unsecured and unencrypted IPv6 TCP/IP Connection. For this project, the MBR was based on BSM metadata which consisted of:

- The time and location where the MBR was created
- The reporter LV information
- The LMBD method that caused the MBR creation
- Some combination of the start and stop time and location of the suspected misbehavior and attacker LV information, depending on the LMBD method

After the lab setup was completed and communication between the OBE and SCMS Server was confirmed, the requirements and design for two LMBD methods, Proximity Plausibility and Warning Report, were developed and an existing OBE SW implementation was updated to support the LMBD methods and MBR transmission. For GMBD, the primary methods selected for implementation were the Device-based and Event-based Methods while the framework was put in place for the Feature-based Method. There was no prior SW/HW base to build upon for GMBD so the components of GMBD were divided between an Application Server and a Database Server. For each of the components, detailed design and implementation activities were performed. To support the conceptual proofs, a number of configuration items were defined to control the behavior of the methods for both LMBD and GMBD.

The methods were tested at a conceptual level, which means the methods were not tested rigorously to validate the maturity of the method nor were the deployment ready requirements developed. Instead, the methods were tested to validate the ability to detect misbehavior by using a given data set suspected of containing misbehavior. The methods were tested with the minimum and maximum configurable parameter thresholds to support the validation. The primary data set used for the testing was data from the DAS installed by VTTI in each of the CAMP VSC3 V2V-MD Project ILVs which were deployed in the SPMD Test Bed. Utilizing the SPMD data was desirable because, based on previous analysis, it was known that there were instances of non-ideal hardware installation as well as incorrect SW configuration on some of the VADs and ASDs which should trigger LMBD by the two methods selected for this Project.

VTTI compiled a list of the VADs/ASDs that were deemed as the top offenders in causing warnings in ILVs as along with the corresponding warning events and event timing. The data from these events were then exported into CSV data files that included all the necessary BSM data from the RV identified in the top offenders table, other RVs that were within communication range at the time, as well as BSM data from the HV.
For LMBD, the primary data used for testing methods were the CSV files generated from the process described above. The test data used for validating the GMBD methods were the MBRs generated by the LMBD methods. Both the LMBD and GMBD methods were tested with minimum and maximum configurable parameter thresholds to support the validation. Note that since only the framework for the Feature-based Method was developed and implemented and given the limitations on the availability of test data, this method was not tested in detail.

When provided inputs which included device behavior consistent with that which should trigger LMBD, the results of the testing showed that the LMBD methods could detect the perceived misbehavior. Similarly, GMBD testing showed that the methods could analyze multiple reports and successfully link them as a common device and, depending on the configured thresholds, flag this device for revocation if necessary. It should be noted that these results were obtained in a lab setting in which limited field acquired data sets, selected because they mimicked misbehavior, were replayed into the OBEs. This work should be considered an initial step in MBD research.
8 References

### List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>ASD</td>
<td>Aftermarket Safety Device</td>
</tr>
<tr>
<td>ASN.1</td>
<td>Abstract Syntax Notation One</td>
</tr>
<tr>
<td>BSM</td>
<td>Basic Safety Message</td>
</tr>
<tr>
<td>BSW</td>
<td>Blind Spot Warning</td>
</tr>
<tr>
<td>CAMP LLC</td>
<td>Crash Avoidance Metrics Partners LLC</td>
</tr>
<tr>
<td>CAN</td>
<td>Controller Area Network</td>
</tr>
<tr>
<td>CBF</td>
<td>Counting Bloom Filter</td>
</tr>
<tr>
<td>CRL</td>
<td>Certificate Revocation List</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Value</td>
</tr>
<tr>
<td>DAS</td>
<td>Data Acquisition System</td>
</tr>
<tr>
<td>DOTAP</td>
<td>DENSO Over-the-Air-Processor</td>
</tr>
<tr>
<td>DRC</td>
<td>Driver Reaction Considered</td>
</tr>
<tr>
<td>DRNC</td>
<td>Driver Reaction Not Considered</td>
</tr>
<tr>
<td>EE</td>
<td>End Entity</td>
</tr>
<tr>
<td>FCW</td>
<td>Forward Collision Warning</td>
</tr>
<tr>
<td>GMBD</td>
<td>Global Misbehavior Detection</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>http</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HV</td>
<td>Host Vehicle</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>ILV</td>
<td>Integrated Light Vehicle</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>IPv6</td>
<td>Internet Protocol Version 6</td>
</tr>
<tr>
<td>ITS</td>
<td>Intelligent Transportation Systems</td>
</tr>
<tr>
<td>LCW</td>
<td>Lane Change Warning</td>
</tr>
<tr>
<td>LIM</td>
<td>Linkage Information Manager</td>
</tr>
<tr>
<td>LMBD</td>
<td>Local Misbehavior Detection</td>
</tr>
<tr>
<td>LV</td>
<td>Linkage Value</td>
</tr>
<tr>
<td>MA</td>
<td>Misbehavior Authority</td>
</tr>
<tr>
<td>MBD</td>
<td>Misbehavior Detection</td>
</tr>
<tr>
<td>MBR</td>
<td>Misbehavior Report</td>
</tr>
<tr>
<td>MWMD</td>
<td>Multiple Warning Multiple Device</td>
</tr>
<tr>
<td>MWSD</td>
<td>Multiple Warning Single Device</td>
</tr>
<tr>
<td>OBE</td>
<td>On-Board Equipment</td>
</tr>
<tr>
<td>OER</td>
<td>Octet Encoding Rules</td>
</tr>
<tr>
<td>OSGi</td>
<td>Open Source Gateway Initiative</td>
</tr>
<tr>
<td>OTA</td>
<td>Over-the-Air</td>
</tr>
<tr>
<td>PCAP</td>
<td>Packet Capture</td>
</tr>
<tr>
<td>PoC</td>
<td>Proof of Concept</td>
</tr>
<tr>
<td>RA</td>
<td>Registration Authority</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>RSE</td>
<td>Road Side Equipment</td>
</tr>
<tr>
<td>RV</td>
<td>Remote Vehicle</td>
</tr>
<tr>
<td>SAE</td>
<td>SAE International</td>
</tr>
<tr>
<td>SCMS</td>
<td>Security Credential Management System</td>
</tr>
<tr>
<td>SDHCAN</td>
<td>Sensor Data Handler CAN</td>
</tr>
<tr>
<td>SDHGPS</td>
<td>Sensor Data Handler GPS</td>
</tr>
<tr>
<td>SPMD</td>
<td>Safety Pilot Model Deployment</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SWSD</td>
<td>Single Warning Single Device</td>
</tr>
<tr>
<td>TA</td>
<td>Threat Arbitration</td>
</tr>
<tr>
<td>TC</td>
<td>Target Classification</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>3D</td>
<td>Three-Dimensional</td>
</tr>
<tr>
<td>V2I</td>
<td>Vehicle-to-Infrastructure</td>
</tr>
<tr>
<td>V2V</td>
<td>Vehicle-to-Vehicle</td>
</tr>
<tr>
<td>V2V-Interoperability</td>
<td>Interoperability Issues of Vehicle-to-Vehicle Based Safety Systems (Project)</td>
</tr>
<tr>
<td>V2V-MD</td>
<td>Vehicle-to-Vehicle Safety System Light Vehicle Builds and Model Deployment Support (Project)</td>
</tr>
<tr>
<td>VAD</td>
<td>Vehicle Awareness Device</td>
</tr>
<tr>
<td>VSC</td>
<td>Vehicle Safety Communications</td>
</tr>
<tr>
<td>VSC3</td>
<td>Vehicle Safety Communications 3 (Consortium)</td>
</tr>
<tr>
<td>VSC6</td>
<td>Vehicle Safety Communications 6 (Consortium)</td>
</tr>
<tr>
<td>VTTI</td>
<td>Virginia Tech Transportation Institute</td>
</tr>
<tr>
<td>WAVE</td>
<td>Wireless Access in Vehicular Environment</td>
</tr>
<tr>
<td>WMH</td>
<td>Wireless Message Handler</td>
</tr>
<tr>
<td>WSA</td>
<td>WAVE Service Advertisement</td>
</tr>
<tr>
<td>WSU</td>
<td>Wireless Safety Unit</td>
</tr>
</tbody>
</table>