Enhanced DSRC Security

Final Report

by

Richard R. Brooks
313-C Riggs Hall
PO Box 340915
Clemson, SC 29634-0915
USA
Tel: 864-656-0920
Voicemail: 864-986-0813
e-mail: rrb@acm.org
web: http://www.clemson.edu/~rrb
PGP: 48EC1E30 Clemson University and Benedict College

Fei Sun
Clemson University

Gurcan Comert
Benedict College, Columbia, SC

March, 2022

Center for Connected Multimodal Mobility (C^2M^2)

CLEMSON UNIVERSITY

Benedict College | The Citadel, The Military College of South Carolina | SC State University | University of South Carolina

200 Lowry Hall, Clemson University
Clemson, SC 29634

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by the Center for Connected Multimodal Mobility (C^2M^2) (Tier 1 University Transportation Center) Grant, which is headquartered at Clemson University, Clemson, South Carolina, USA, from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

The U.S. DOT retains non-exclusive rights.

ACKNOWLEDGMENT

The research team would like to thank Dr. James Martin, Clemson University, and his research group for sharing DSRC units. The team would also like to thank Jon Oakley of Clemson University for making the Wave communication protocol available to us.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle	5. Report Date
Enhanced DSRC Security	
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Fei Sun Richard Brooks, Ph.D. Gurcan Comert, Ph.D.; ORCID: 0000-0002-2373-5013	

9. Performing Organization Name and Address	10. Work Unit No.
Center for Connected Multimodal Mobility (C^2M^2) Clemson University 200 Lowry Hall, Clemson, SC 29634	
	11. Contract or Grant No. 69A3551747117

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Center for Connected Multimodal Mobility (C^2M^2) Clemson University 200 Lowry Hall, Clemson, SC 29634	Final Report (December 2018 - November 2020)
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Applications using dedicated short-range communication (DSRC) are being developed to prevent automobile accidents. Many DSRC implementations, applications, and network stacks are not mature. They have not been adequately tested and verified. This study illustrates the security evaluation of a DSRC wireless application in vehicular environments (DSRC/WAVE) protocol implementation. We set up a simulation of a working road side unit (RSU) on real DSRC devices. Our experiments work on the Cohda testbed with DSRC application wsm-channel. We extended the functionality of wsm-channel, an implementation of WAVE short message protocol (WSMP) for broadcasting GPS data in vehicular communications, to broadcast car information and RSU instructions. Next, we performed Denial of Service attacks to determine how few packets needed to be dropped to cause automobile crashes. Hidden Markov Models (HMM) are constructed using sniffed side-channel information since operational packets would be encrypted. The inferred HMM tracks the protocol status over time. To test HMM's ability to predict which packets will be dropped, we used a simulation-based experiment and implemented a DSRC-supported stop light application. Using these simulations, we were able to show that we could accurately identify the packets we needed to drop by using timing and packet size side channels. The attack simulation using inter-packet delay side-channel features worked best to drop necessary packets with a 2.5 % false-positive rate (FPR) while the attack using a packet size side-channel worked with a 9.5% FPR.

17. Keywords	18. Distribution Statement
DSRC, Cybersecurity, CAV	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	**30**	NA

Table of Contents

List of Tables

List of Figures

EXECUTIVE SUMMARY

Applications using dedicated short-range communication (DSRC) are being developed to prevent automobile accidents. Many DSRC implementations, applications, and network stacks are not mature. They have not been adequately tested and verified. This study illustrates the security evaluation of a DSRC wireless application in vehicular environments (DSRC/WAVE) protocol implementation. We set up a simulation of a working road side unit (RSU) on real DSRC devices. Our experiments work on a Cohda device running a DSRC wsm-channel application. We extended the functionality of wsm-channel, an implementation of WAVE short message protocol (WSMP) for broadcasting GPS data in vehicular communications, to broadcast car information and RSU instructions. Next, we performed Denial of Service attacks to determine how few packets needed to be dropped to cause automobile crashes. Hidden Markov Models (HMM) are constructed using sniffed side-channel information since operational packets would be encrypted. The inferred HMM tracks the protocol status over time. To test HMM's ability to predict which packets will be dropped, we used a simulation-based experiment and implemented a DSRC-supported stop light application. The attack simulation following the timing side-channel worked best to drop necessary packets with a 2.5 % false-positive rate (FPR) while the attack following size worked with 9.5% FPR.

CHAPTER 1

Introduction

1.1 Motivation

Dedicated Short Range Communication (DSRC) is 802.11p based wireless communication technology. It's widely used for communication between vehicles and the surrounding infrastructure. Wireless access in vehicular environments (WAVE) is one of the communication protocols of DSRC. It provides stable, high-speed communication between connected vehicles. Many applications based on DSRC/WAVE are being developed to improve traffic efficiency and assist driving. Vehicle to vehicle (V2V) technology is in many new cars. V2V is DSRC based. Vehicles use V2V and a global positioning system (GPS) to share and detect information within range. This could alert and warn drivers of emergencies that are not easy to see. For example, Left Turn Assist (LTA) systems help avoid blind spots when drivers turn left. It warns drivers if they are driving in front of another vehicle traveling in the opposite direction. It could help reduce traffic collisions.

With DSRC becoming the accepted automotive wireless mobility standard, DSRC development groups are concerned that DSRC protocols, applications, and stacks are not mature. Similarly, many applications using the DSRC protocol have not been adequately tested and verified. In this research, we are interested in a "black box" analysis of WAVE short message protocol (WSMP), the messaging protocol used by DSRC/WAVE. We assume the WSMP packets are encrypted in the "black box" analysis, and analysis does not depend on the contents.

1.2 Problem Statement

We conducted our security research on Cohda fifth-generation On-Board Unit (OBU), provided by Dr. James Martin, School of Computing, Clemson University. We designed a traffic control system simulation of autonomous vehicles to implement the security analysis. The system can run on road side units (RSUs, i.e., road side units – DSRC equipped devices with computational capabilities), and it has the potential to replace traffic lights. The application works to avoid crashes for automatic driving. We did a side-channel analysis of the sniffed WSMP traffic. A Hidden Markov Model (HMM) was built using sniffed packet traces. We identified and predicted critical packets in the system using the HMM. The critical packet refers to the stop instruction packet sent from RSU to stop the vehicle. With the known weak points, we can generate a targeted attack. We performed the DDoS attack simulation with HMM predictions in offline experiments. We dropped the important packets and caused car crashes.

This report includes a literature review of DSRC status and the techniques used to build the HMM, a review of the method used for protocol analysis and side-channel analysis of the traffic control system, and a description of the building and testing of the HMM (Sun et al., (2022)).

CHAPTER 2

Literature Review

2.1 DSRC

At this time, DSRC is the communication technology that dominates connected vehicle applications (Yang. et al., 2018). As DSRC provides reliable and real-time communication between DSRC-equipped vehicles, it starts to be widely used to coordinate driving and road management. From the U.S. Department of Transportation (USDOT) report (Bettisworth, 2015), we know that most lights and traffic signals will enable DSRC in twenty years. It's reasonable because DSRC could provide real-time crash-avoiding alerts. DSRC-equipped vehicles can share critical information, providing the possibility of in-obstructed awareness.

2.2 Protocol Analysis

Figure 1 shows the architecture of the DSRC implement standard (Kenney, 2011). The physical protocol, including PHY layer and medium access control (MAC) sub-layer, is defined in IEEE 802.11p wireless access in vehicular environments (WAVE), which enhances IEEE 802.11 (WIFI- standard) to support Intelligent Transportation System (ITS). It provides real-time data exchanging by removing the general channel-establish in network communication. It defines the spectrum of channels for DSRC in the US. Authentication and data confidentiality mechanisms provided by the IEEE 802.11 standard cannot be used. DSRC equipped vehicles in a specific sight range can receive data frames as soon as they arrive on the communication channel.

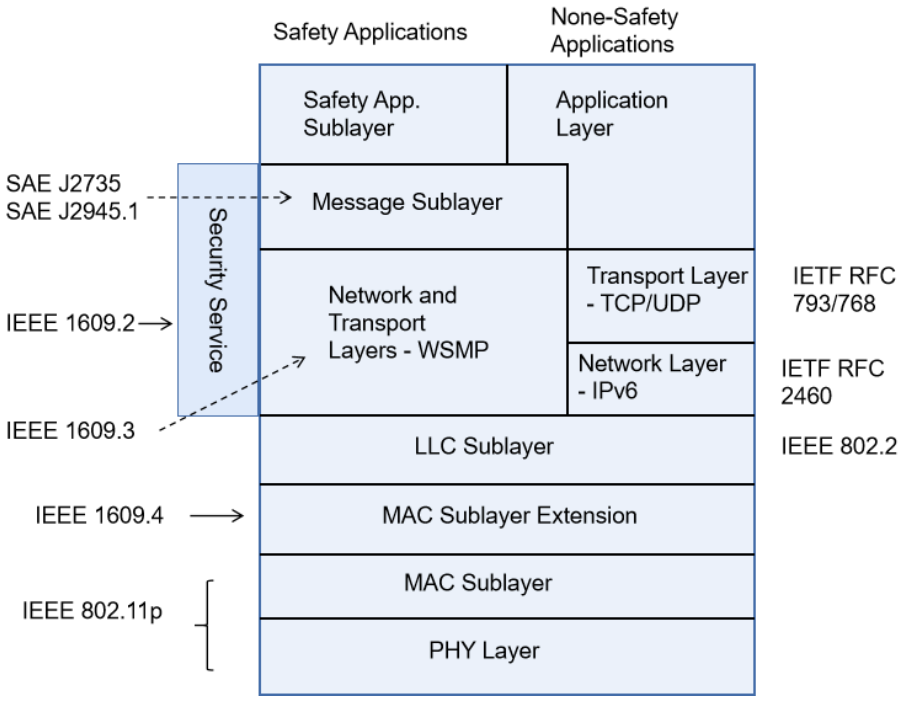

Figure 1: **DSRC layers and standards** [1]

[1] J. B. Kenney, 2011

2.3 Denial of Service Attack on DSRC

We used a Denial of Service (DoS) attack on DSRC channels to perform our attack. DoS attack refers to a network attack method where attackers hamper legitimate users' access to a network service by occupying network resources or machine calculation resources. There are several methods to perform a DoS attack. Packet flooding aims at network bandwidth; amplification attack exploits the flaw of some services where response frames are much larger than requests, e.g., DNS request; SYN-related attack utilizes the time-out mechanism of the TCP session.

In this project, as DSRC communication is broadcasting on channels and without session set-up, we choose a packet flooding attack to jam the DSRC channel and result in legitimate packets dropping.

Researchers have already investigated some processes in the DoS attack on DSRC. Laurendeau et al. (2006) evaluate DoS as a major risk in his DSRC threats analysis model. The paper also points out that DSRC standards should enhance the security of the lowest possible layer to prevent DoS, such as providing link layer authentication. Islam et al. developed an application, CVGuard for DoS attack detection and prevention; the application is designed to monitor the context of DSRC communication and detect the attack based on road policies and rules.

2.4 Traditional Hidden Markov Model

Shalizi et al. (2002) proposed Causal State Splitting and Reconstruction (CSSR) algorithm to generate HMMs from discrete sequences of data. The algorithm makes no prior assumptions about the model structure. It infers the model structure (the number of hidden states and their transition structure) from the sequence of observation and a maximum state-space parameter. The derived HMMs from CSSR have predictive optimality properties *I*.

The CSSR algorithm assumes that the sequence of training data is enough to build the model that represents the underlying process. And it requires prior knowledge of maximum state space parameter.

2.5 Our Hidden Markov Model

Based on the approach of CSSR, Schwier et al. (2010) presented a method, zero-knowledge HMM inference, for automatically inferring the maximum data window size from training data as part of the model construction process. Thus, they proposed a method inferring HMMs only from the sequence of observations. This improved HMM inference approach has already been used in network traffic side-channel analysis.

Harakrishnan et al. (2011) proposed timing side-channel analysis for detecting protocol tunneling. They used a zero-knowledge approach (Schwier et al., 2009) to extract HMMs for extracted keystroke dynamics of languages. They then used the HMM for language detection. Zhong et al. (2015) proposed the side-channel analysis of the Phasor Measurement Unit (PMU) protocol used by the communications network of the smart grid. They isolated the packets of the target PMU sent through a VPN channel shared with other PMUs, followed Denial-of-Service (DoS) attacks that selectively drop packets from the target PMU.

CHAPTER 3

Method

3.1 Simple Traffic Simulation Model

To test our approach, we wrote a discrete event simulation program. This program was used to run a set of DSRC attack scenarios. As shown in Figure 2, the traffic intersection control is designed for an intersection of two-lane roadways. Vehicles running on the road are following:

1) No pedestrians are allowed at this crossroad.
2) Vehicles can come from one of the four directions North (N), South (S), West (W), or East (E), and go straight, left, or right. U-turn is not permitted at this crossroad.
3) All vehicles are connected autonomous vehicles (CAVs) and controlled by onboard unit (OBU) speed control. There is a central RSU (i.e., road side unit – an intelligent roadside unit with computational capabilities) in the center of the intersection. They change speeds only following the instruction from RSU.
4) We set the center of the intersection (0, 0) as the origin 'O'.
5) The roads are each 4 meters wide.
6) As shown in Figure 2, the exit points of each lane are A, B, C, and D. Coordinates are A (-4, -2), B (4, 2), C (-2, 4), D (2, -4). Vehicles would start to report information 50 meters away from the RSU (O in Figure 2).
7) As shown in Figure 2, the entry points are A', B', C', D'. Coordinates are A' (-4, 2), B (4, -2), C (2, 4), D (-2, -4).
8) The path of the vehicle in the intersection is calculated by linear distance from exit points to entry points. The distance a vehicle should drive in the intersection is calculated by the sum of path distance and vehicle length. For example, if a vehicle is moving from East to North, the path in the intersection should be line BC' (Figure 2).

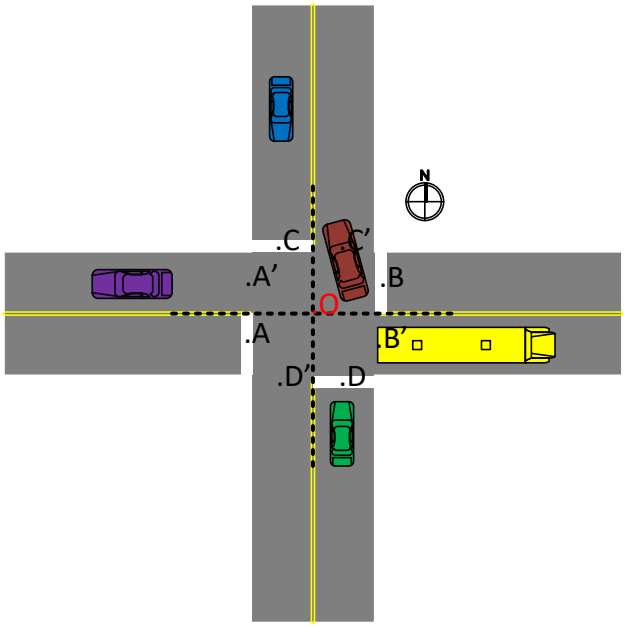

Figure 2: Crossroad graph

Each car sends its information to RSU when it's 50 meters away from the intersection. The RSU estimates vehicles' arrival time at the intersection and sends stop instructions if the intersection is busy at the estimated arrival time.

3.2 Protocol Implementation

Our research group member, Jon Oakley, implemented a reliable WAVE short message protocol (WSMP) communication application Wsm-channel. WSMP, IEEE 1609.3, is a DSRC based communication protocol that allows data rates parameters (Kenney, 2011). Wsm-channel could broadcast GSP information of host OBU on a WSMP channel. We implement the modes "FWDTX" and "FWDRX" on WSM-CHANNEL to forward packets through different protocols. FWDTX is forwarding received UDP packets to the WAVE protocol. FWDRX is forwarding received WAVE packets to the UDP protocol. Thus, processes on different OBUs can exchange data using this extended application.

The flowchart of communication is given in Figure 3. For example, if Process A on DSRC1 needs to send packet A to Process I on DSRC2, Process II receives packet A and sends packet B back to Process I. WSM-CHANNEL FWDTX mode, and FWDRX mode is running on DSRC1 and DSRC2. Process I and Process II are listening to UDP for receiving packets. The communication steps are as follows:

1a. DSRC1: Process I sends packet A to UDP.
1b. DSRC1: WSM-CHANNEL FWDTX thread receives packet A and sends it to WSMP at interface "wave-raw." Packet A is broadcasting at wave-raw.
2a. DSRC2: WSM-CHANNEL FWDRX thread receives packet A at wave-raw and sends it to UDP.
2b. DSRC2: Process II receives packet A.
3a. DSRC2: Process II generates packet B and sends it to UDP.
3b. DSRC2: WSM-CHANNEL FWDTX thread receives packet B and sends it to WSMP at interface "wave-raw." Packet B is broadcasting at wave-raw.
4a. DSRC1: WSM-CHANNEL FWDRX thread receives packet B at wave-raw and sends it to UDP.
2b. OBU1: Process I receives packet B.

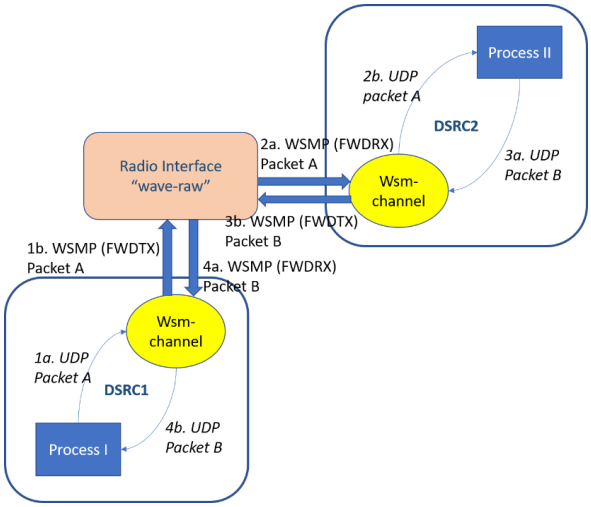

Figure 3: Flowchart of the packet within DSRC communication

In our simulation, we focused on the communications between an onboard unit (OBU) and a road side unit (RSU). The OBU stores the information of all the cars approaching the intersection, and the RSU serves as the road safety or control unit for the intersection. Thus, we observe communication between vehicles and RSU over the DSRC channel, wave-raw, as illustrated in Figure 4.

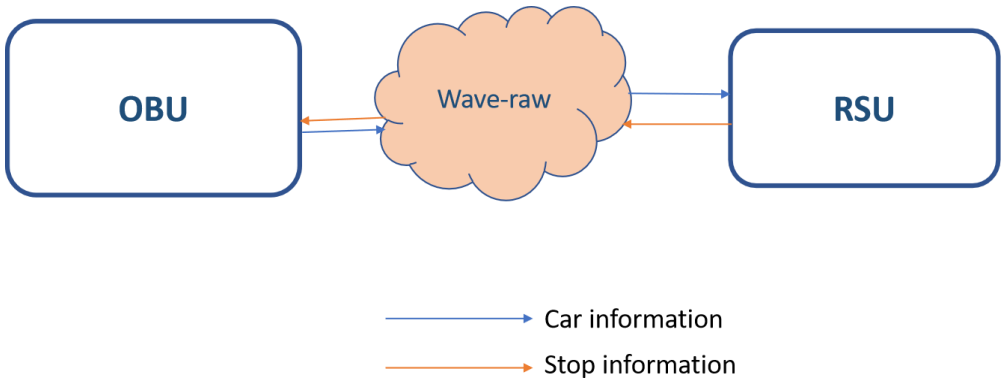

Figure 4: Communication between OBU and RSU

3.3 "Black Box" Analysis – Side-channel Analysis

As the IEEE1609.2 standard of the DSRC/WAVE stack defines the standard mechanisms for authenticating and encrypting messages, we consider the "black box" analysis. We look at the side-channel characteristics (packet size, packet inter-delay) of the WAVE short message protocol (WSMP). Even if encryption and authentication are implemented as specified in the IEEE 1609.2 standard, DSRC/WAVE may still be susceptible to a "black box" analysis that does not depend on the contents.

According to the sniffed traffic (Figure 5), where the "time" refers to the inter-packet time, the packets are not always arriving at the same rate, which means the protocol is not active all the time. If we perform the attack at an inactive time, we cannot cause any trouble. Moreover, since flooding traffic is easy to recognize, the devices may lose access to the channel.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	b8:ff:36:ff:36…		WSMP	176	WAVE Short Message Protocol IEEE P1609.3
2	0.004006	b8:ff:36:ff:36…		WSMP	178	WAVE Short Message Protocol IEEE P1609.3
3	0.270763	b8:ff:36:ff:36…		WSMP	177	WAVE Short Message Protocol IEEE P1609.3
4	0.025839	a8:ff:36:ff:36…		WSMP	234	WAVE Short Message Protocol IEEE P1609.3
5	6.853376	b8:ff:36:ff:36…		WSMP	177	WAVE Short Message Protocol IEEE P1609.3
6	0.027991	a6:ff:36:ff:36…		WSMP	233	WAVE Short Message Protocol IEEE P1609.3
7	0.212516	b8:ff:36:ff:36…		WSMP	177	WAVE Short Message Protocol IEEE P1609.3
8	0.230590	ba:ff:36:ff:3c…		WSMP	177	WAVE Short Message Protocol IEEE P1609.3
9	0.030930	a8:ff:30:ff:36…		WSMP	234	WAVE Short Message Protocol IEEE P1609.3
10	6.368706	b8:ff:36:ff:36…		WSMP	193	WAVE Short Message Protocol IEEE P1609.3
11	0.029879	a6:ff:36:ff:36…		WSMP	252	WAVE Short Message Protocol IEEE P1609.3
12	6.531107	b8:ff:36:ff:36…		WSMP	178	WAVE Short Message Protocol IEEE P1609.3

Figure 5: Sniffed DSRC traffic

We develop a network protocol analysis method based on the side channel and Hidden Markov Model (HMM). The overall process flow is shown in Figure 6. We build HMM for the system

protocol to understand the protocol regulations. We assume that the WAVE packets will be encrypted, so we apply size and timing side channels. We sniff traces of DSRC network protocols. We can identify network protocol states by using observed packet characteristics to associate each sniffed packet with a class. Protocol participants are known. Transitions between protocol states are given by their positions in the sequence. With the HMM, we successfully isolate the target packets of stop information sent by RSU, followed by DoS attacks that selectively drop packets from RSU. The goal is to side-channel vulnerabilities of WAVE protocol assuming all the security services are implemented.

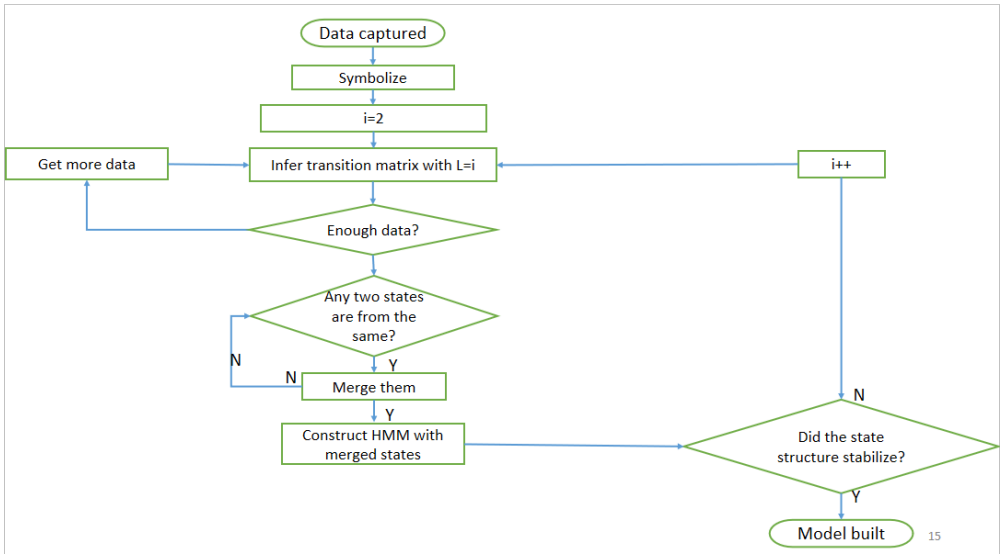

Figure 6: Flowchart of inferring HMM

3.3.1 Side Channel Symbolization

According to the sniffed traffic (see Figure 5), we can get two important side channels: timing and size. We use inter-packet time instead of receive time for analysis. The inter-packet time, also known as delta time, is calculated by subtracting the previous packet's receive time from the current packet's time. In other words, where it is the receive time of packet *i*. We start obtaining measure with *i=2* (e.g., inter-packet time $\Delta t_{i=2}=t_2-t_1$)

We have two side channels to build an HMM for each side channel. First, we develop the timing HMM. We group the data by plotting a histogram of timing and finding different ranges, Figure 7. We assign anything in a timing range a unique symbol, shown in Table 1. Finally, we can get a long sequence string from the data. Later, we will do the same with the size pattern when building the size HMM.

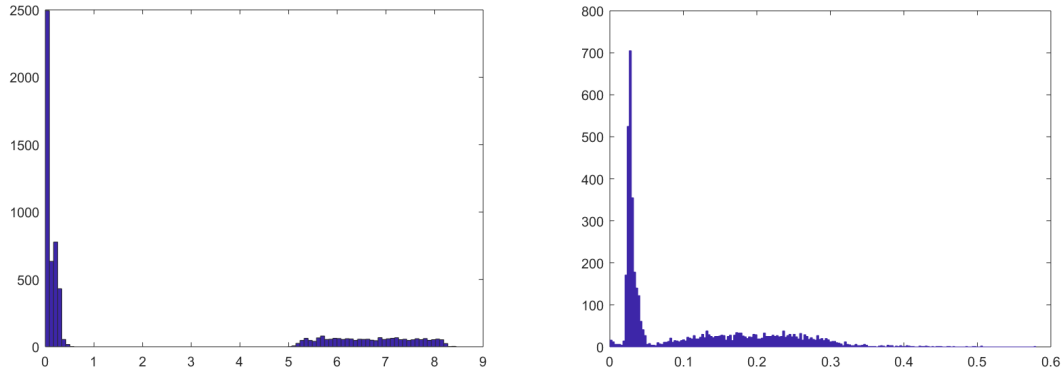

Figure 7: Histogram of timing (i) range (0,9) (ii) range (0,1)

Table 1: Timing observation ranges

Observation type	Timing range
A	$\Delta t \leq 0.06$
B	$0.06 < \Delta t \leq 1$
C	$\Delta t > 1$

3.3.2 Hidden Markov Model Inference

We use HMM to analyze side-channel information. We extend previous approaches (Ryan, 2010), adding hypothesis tests when determining the HMM. We apply z-test to HMMs to determine the statistical significance of the inferred model, which indicates data sufficiency (Yu et al., 2013). Pearson chi-square test proves the significance of evidence to merge two similar states (Ott et al., 2001). The confidence interval approach provides the level of acceptance for putting a string into an HMM (Schwier et al., 2011).

A standard HMM has two sets of random processes, one governing state transition and the other governing symbol outputs. In this paper, we use the representation of an HMM in (Yu et al., 2013), where output symbols are associated with transitions. The two approaches are equivalent (Schwier et al., 2009). This representation uses a tuple G = <A, V, E, P>, where A is a finite alphabet of observations, V is a finite set of nodes or states, E ⊆ V ×A×V is a transition relation, and P: E→[0, 1] is a probability function such that $\sum_{a \in A, v_j \in V} p(v_i, a, v_j) = 1$. Each element expresses the probability the process transitions to state once it is in state v_i . For each pair of (v_i, v_j) , $E(v_i, v_j) = a_i$. It should also meet the requirement that if $E(v_i, v_j) = a$, then $E(v_i, v_k) \neq a$, where $v_i, v_j, v_k \in V$.

Both state transition probability matrix P and state output probability matrix O can be constructed from G. The state output probability matrix refers to the matrix that described the probability distribution of the next observation for each state. We use the state transition probability matrix for steady-state probability calculation and figure plotting. We use the state output probability matrix for generating a string from the HMM and HMM acceptance checking. Following are some important variable calculations in an HMM.

1) Conditional probability $p_{i,j} = \Pr(v_j \mid v_i)$
2) Transition count $c_{i,j} = \#transition\_from\_i\_to\_j\_happened$
   State count $c_i = \sum_j c_{i,j} = \#state\_i\_is\_entered$
3) Asymptotic probability (steady-state probability) matrix $\vec{\pi} = (\pi_1, \pi_2, ..., \pi_n)'$ can be calculated from $\begin{cases} \vec{\pi} P = \vec{\pi} \\ \sum_i \pi_i = 1 \end{cases}$
4) Confidence interval for each transition $CI = Z_{\alpha/2}\sqrt{\frac{p_{i,j}(1-p_{i,j})}{n_i}}$ (Schwier et al., 2011), where $p_{i,j}$ is the conditional probability of the transition, $Z_{\alpha/2}$ is from either the Normal or t-distribution, α is the significance level of confidence, n_i is the times of state v_i.

3.3.2.1 Inferring an HMM from the sequence A with significance level α

1) i=2.
2) State space parameter L=i.
3) a) Infer G_n = <A, V, E, P> from the sequence A.
   b) Merge states in V using Algorithm 1 Pearson chi-square test.
   c) Do model confidence test for G_n. If it doesn't have enough, get more data and start over. Details of the Model confidence test are described in section 3.3.2.3.
4) Get output confidence interval matrix *CI* from G_n
5) a) Infer G_{n+1} = <A', V', E', P'> from the sequence A.
b) Merge states in V' using Pearson chi-square test in section 3.3.2.2.
c) Do model confidence test for G_{n+1}. If the training data doesn't have enough, get more data and start over.
6) Generate an extended sequence B from G_{n+1} whose length is longer than the result from the model confidence test. For the generation method see section 3.3.2.4.
7) Put sequence B into G_n (section 3.3.2.5). Get match probability matrix F.
8) Calculate $|P - F| - CI$, the elements less than zero in the result matrix donates the rejection proportion. Determine the rejection proportion by $P_{rj} = \sum_i d_{i,j} * p_i$, where $d_{i,j} = D_{i,j} - CI, D = \{D \in (P - F) \mid D > 0\}$, p_i is the probability of state i is entered.
9) If P_{rj} greater than α, i++. Repeat steps from 2).
10) Otherwise, quit with G_n as the correct HMM for sequence A.

3.3.2.2 State merging algorithm

We use the pairwise Pearson chi-square test for state merging. The test result shows whether two states are coming from the same state. We merge the pair of most likelihood at one time and update the merging in the output count matrix. We keep doing the pairwise test until all pairs reject the null hypothesis of two states from the same state. With the input of state

transition count matrix *M,* state output matrix *O*, and significant confidence level α, we do the state merging as:

1) Do Pearson pairwise chi-square test of independence (Ott et al., 2001) of rows in transition count matrix M as follows:
   a. Denote the population proportion (or probability) falling in row i, column j as π_{ij} . The total proportion for row i is $\pi_{i.}$. The total proportion for column j is $\pi_{.j}$. If the row and column proportions are independent, then $\pi_{ij}=\pi_{i.}\pi_{.j}$.
   b. The estimated expected value in row i, column j is $E_{ij} = n\pi_{ij} = n\frac{n_{i.}}{n}\frac{n_{.j}}{n} = \frac{(n_{i.})(n_{.j})}{n}$
   c. Test statistic: $\chi^2 = \sum_{i,j}[\frac{(n_{ij} - E_{ij})^2}{E_{ij}}]$
2) Determine the $\chi^2_{\alpha,df}$ statistic for the χ^2 test with significant level α and df = *(r - 1)(c - 1)* where *r* = number of rows, *c* = number of columns.
3) If $\chi^2 <= \chi^2_{\alpha,df}$ for any pairwise tests, the test accepts with significant level α the hypothesis that the two rows are from the same state. Find the minimum χ^2 value χ^2_{min}, and index *i, j* (*i<j*) of the pair of states it comes from.
   a. In the state transition count matrix M, add column j to column i, add row j to row i. Set zero of column j and row j.
   b. In the state output count matrix O, add row j to row i. Set zero of column j.
4) Repeat steps 1), 2), 3) until $\chi^2 > \chi^2_{\alpha,df}$ for all pairwise tests.
5) Remove zero columns and zero rows in M and O. Then quit with merged states transition count matrix and output count matrix.

3.3.2.3 Model confidence test

After deriving a model from the data, we need to know whether the data is enough to derive this model. If not enough, how much more data do we need. Thus, we take the model confidence test algorithm from (Lu et al., 2013) to check the model.

With the input of transition probability matrix *P*, transition count matrix *C*, and asymptotic probability matrix $\vec{\pi}$, we do the test as follows:

1) Null hypothesis: data is not enough for any transitions
   Alternative hypothesis: data is enough for any transitions
2) Test statistic: $z=\min(\frac{p_{i,j}}{\sqrt{\frac{p_{i,j}(1-p_{i,j})}{n_i}}})$, where $0 < p_{i,j} < 1$ is the conditional probability of the transition, $n_i = \sum_j c_{i,j}$ is the total counts of state *i*, $c_{i,j}$ is the element from transition count matrix *C*.

3) Rejection region: Reject *Ho* if $z > z_{\alpha}$ then we don't need to collect more data. Otherwise, we need to collect more data. Enough data $D = \max(\frac{z_{\alpha}^2(1-p_{i,j})}{p_{i,j}\pi_s})$, where $0 < p_{i,j} < 1$.

3.3.2.4 Generate a sequence of length l from an HMM G

We restrict our discussion to ergodic Markov processes, which for all states possible transitions to any state.

1) Choose an initial state $v_o = v_i$ from state set V where $v_i \in V$ and for $\forall v_i, v_j \in V, P(v_o = v_i) = P(v_o = v_j)$.
2) Using the probabilities of the outgoing transitions, select a transition $p_{i,j}$ to move from state v_i to state v_j .
3) Record the label $a_i = E(v_i, v_j)$, where a_i is associated with the chosen transition $p_{i,j}$.
4) Repeat steps 2) and 3) until *l* labels have been recorded.

3.3.2.5 Put sequence A into an HMM G (Schwier et al., 2009)

For every state v_i in V of G, we calculate the state transition probability F in sequence A. If there is no transition in G for a window in sequence A going to the next window, record it as a rejection and turn to the next window.

CHAPTER 4

Results

4.1 Road side unit Simulation

We ran the simulation on DSRC devices as shown in Figure 4 on the first DSRC device; we ran OBU processes; on the second DSRC device, we ran the RSU process.

First, we did the test for the simulation function. In continuous four hours simulation, the RSU works well to avoid crashes. We captured the traffic using tcpdump. According to Figure 8 of WSMP packet detail, the time shift for this packet approximately equals 0 seconds which shows the real-time data exchange.

```
Frame 3224: 253 bytes on wire (2024 bits), 253 bytes captured (2024 bits)
    Encapsulation type: Linux cooked-mode capture (25)
    Arrival Time: Jun 24, 2020 13:12:19.361308000 Eastern Daylight Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1593018739.361308000 seconds
    [Time delta from previous captured frame: 0.030123000 seconds]
    [Time delta from previous displayed frame: 0.030123000 seconds]
    [Time since reference or first frame: 7182.082375000 seconds]
    Frame Number: 3224
    Frame Length: 253 bytes (2024 bits)
    Capture Length: 253 bytes (2024 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: sll:ethertype:wsmp]
```

Figure 8: A WSMP packet

To test the denial of service flaw of the DSRC, we performed a flooding attack at the DSRC channel “wave-raw.” We kept the simulation running with unexpected GPS information broadcasting on OBU at a very high-speed rate to cause packet dropping on the RSU (Figure 9). Several crashes were immediately detected while the flooding attack was ongoing (Figure 10).

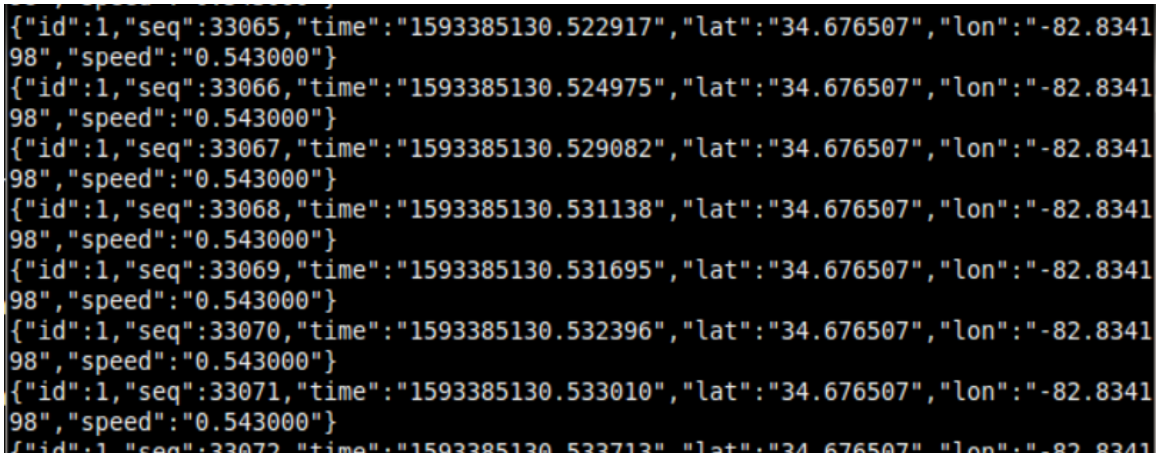

Figure 9: Flooding GSP information

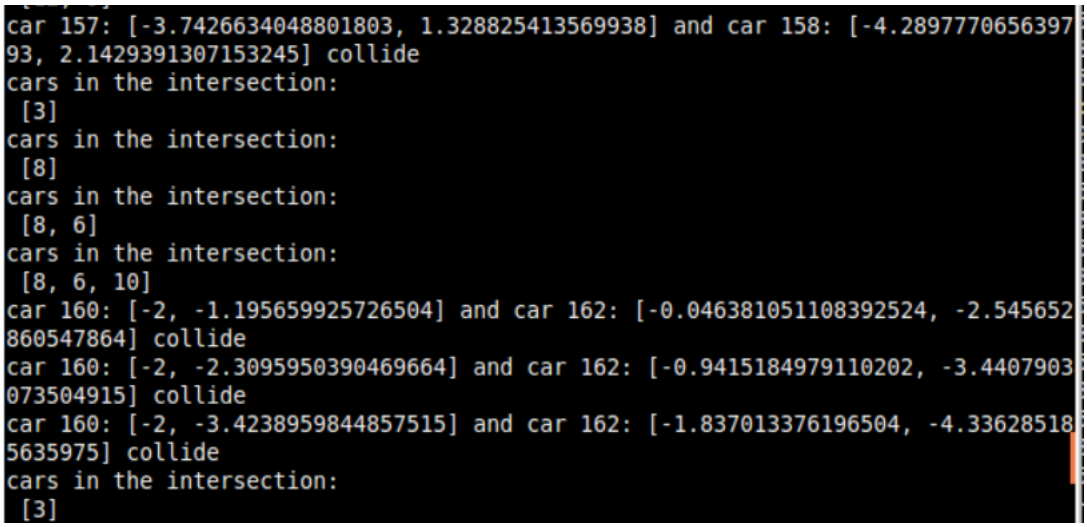

Figure 10: Collision detected while drop stop instructions

4.2 Hidden Markov Models

With the method described in section 3.3, we obtained Timing HMM and Size HMM. From the HMM, we want to predict the arrival of critical packets, which are the instruction packets sent from RSU, so that we can create the attack aiming at critical packets.

4.2.1 Timing HMM

Timing HMM with state-space parameter *l=2* is shown in Figure 11. The symbolization is described in Table 1. Table 2 includes the state details and state transition matrix. From the clear-text detail of the instruction packet (Figure 8), it is recognized as a type ‘a’ packet. So we consider the prediction of ‘a’ packet in Timing HMM. According to Table 2, the packet leaving state ‘ab’ has the highest likelihood (0.8800) to be an ‘a’ packet. And, the packet leaving state ‘bb’ has the second-highest likelihood (0.7300) to be an ‘a’ packet.

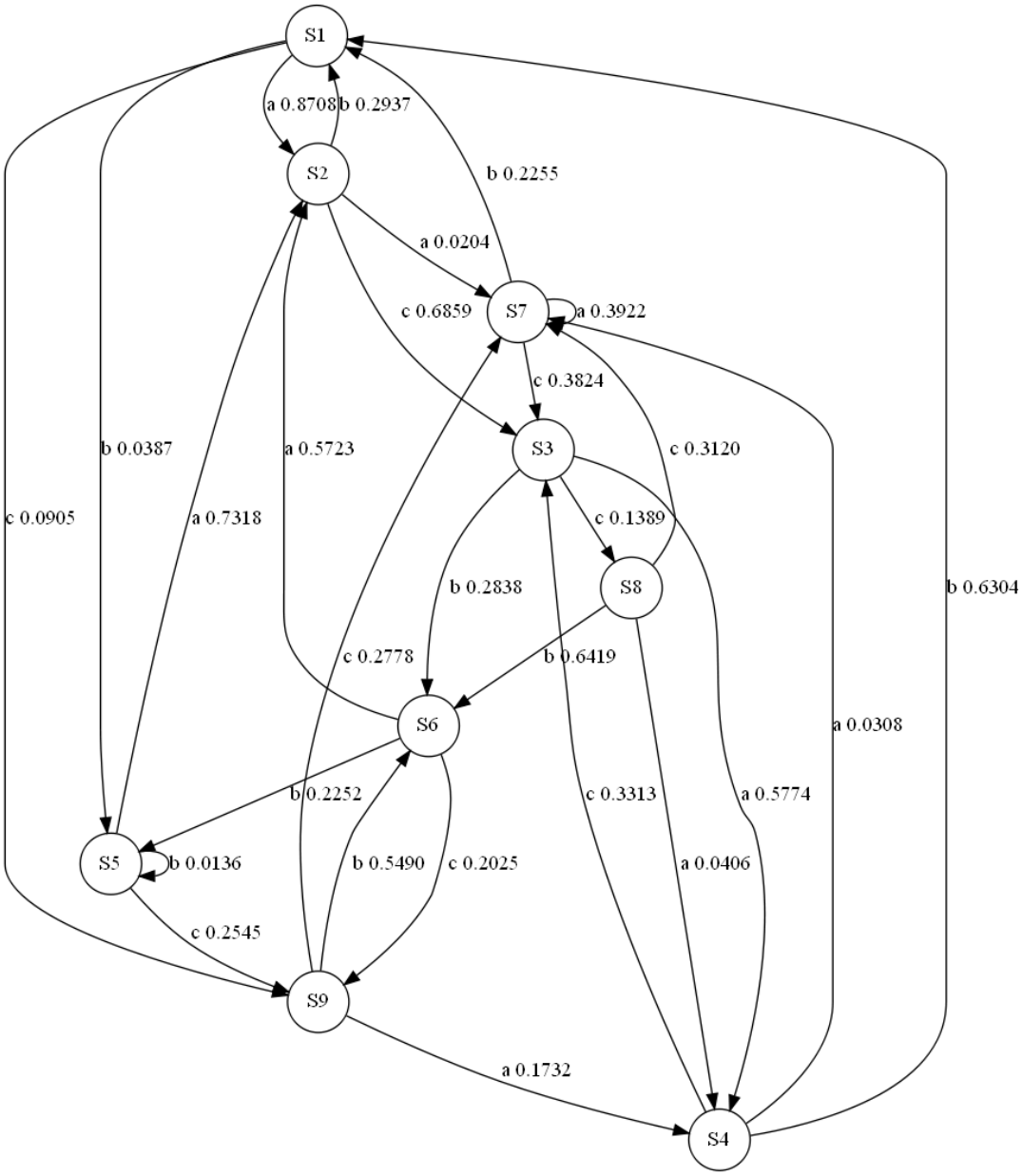

Figure 11: Timing HMM

Table 2: Timing HMM descriptions

State	State Detail	State Transition Matrix		
		a	b	c
1	ab	0.8708	0.0387	0.0905
2	ba	0.0204	0.2937	0.6859
3	ac	0.5774	0.2838	0.1389
4	ca	0.0308	0.6304	0.3313
5	bb	0.7318	0.0136	0.2545
6	cb	0.5723	0.2252	0.2025
7	aa	0.3922	0.2255	0.3824
8	cc	0.0406	0.6419	0.3120
9	bc	0.1732	0.5490	0.2778

4.2.2 Size HMM

Similarly, size HMM with state-space parameter *l=2* is shown in Figure 12. The symbolization is described in Table 3. Table 4 includes the state details and state transition matrix. From the clear-text detail of the instruction packet (Figure 8), it is recognized as a type 'y' packet. So we consider the prediction of the 'y' packet in Timing HMM. According to Table 4, the packet leaving state 'yx' has the highest likelihood (0.7101) to be a 'y' packet.

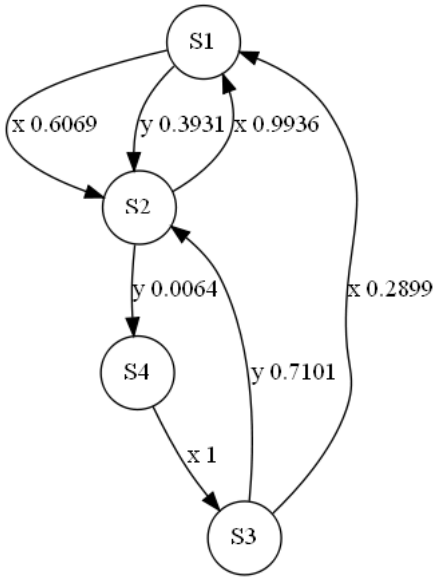

Figure 12: Size HMM

Table 3: Size classification

Observation type	size range
x	$s \leq 210$
y	$s > 210$

Table 4: Size HMM descriptions

State	State Detail	State Transition Matrix	
		x	y
S1	Xx	0.6069	0.3931
S2	Xy	0.9936	0.0064
S3	Yx	0.2899	0.7101
S4	Yy	1	0

4.3 Attack Simulation with HMM predictions

As shown in Table 5, we have three target states: timing state ‘ab’, timing state ‘bb’, and size state ‘yx’. We set up an attack simulation to test the HMMs prediction by dropping the packets leaving the target states.

Table 5: Target states

Target State	Category	Strings
1	Timing	ab
2	Timing	bb
3	Size	yx

We used the wave-raw process to simulate the DSRC/WAVE communication channel. The wave-raw process sends packets from the speed adjustment module and RSU to each other. For each received packet, the wave-raw process marks it with symbols as described in Table 1 and Table 3. We set up six scenarios of experiments. In each scenario, the wave-raw process would drop packets after different states.

The attack was simulated under six different scenarios. The first scenario is a control group to see the crash rate if all packets from RSU dropped. In this scenario, the wave-raw process didn’t forward any packets from RSU to OBU. In the second scenario, we dropped the packet after the first timing state ‘ab’ was observed. In the third scenario, we dropped the packet after either timing state ‘ab’ or ‘bb’ was seen. In the fourth scenario, we dropped the packet after the size state ‘yx’ occurred. In the fifth scenario, we dropped the packet after any defined target states in Table 5. In the sixth scenario, we dropped the packet after the state was recognized as a combination of a target timing state and a size state.

In these scenarios, a crash occurs when a car drives into the intersection while it should stop according to the information from RSU but the information dropped by an attack. The simulation tracks vehicle positions and velocities. The DSRC application logic sends a packet telling a vehicle to stop when the simulation believes that a crash would occur (i.e., the positions and velocities of the two vehicles would force them to be in the same place at the same time.) A simulated crash occurs when the denial-of-service attack succeeds in making the system unable to transmit the packet telling the vehicle to stop.

4.4 Data Sets

After six scenarios attack simulation experiments, we obtained the data in Table 6. For each scenario, a total of 2000 cars approach the intersection. We consider the crash in an intersection with normal traffic flow and only between vehicles from different directions. Thus, we set vehicles that come from four directions with Uniform inter-arrival times from *U(2.1, 4.0)*

seconds. The second column is the description of each scenario. The third column is the total packet number sent through DSRC/WAVE. The fourth column is the number of dropped packets. The fifth column shows the number of instructions in dropped packets. The sixth column is the number of crashes caused during each scenario[2].

We considered six attack strategies, shown by the columns in Table 6. Strategy 1 did not use the HMMs. Strategy 2 dropped packets when the timing HMM identified the state most likely to transmit a packet triggering a stop signal. Strategy 3 dropped packets when the timing HMM recognized any state that could transmit a packet triggering a stop signal; Strategy 4 dropped packets when the packet size HMM identified the state most likely to transmit a packet triggering a stop signal. Strategy 5 dropped packets when either the packet size or timing HMM identified a state likely to transmit a packet triggering a stop signal. Strategy 6 dropped packets when both the packet size and timing HMMs identified states likely to transmit a packet triggering a stop signal.

Table 6: Attack simulation

Scenario	Attack target state	total packet #	drop #	stop packet #	drop instruction #	crash #
1	Control group	3130	1130	1130	1130	257
2	Maximum likelihood timing state	3051	349	1050	299	40
3	Any target timing states	3147	471	1145	410	76
4	Target size state	2904	546	903	357	75
5	Any defined target state	2918	599	916	413	100
6	Combination of a target timing state and a size state.	3032	330	1032	292	32

4.5 Statistical Analysis

We mark packets from RSU as positive packets, packets from OBU as negative packets. The true positive (TP), true negative (TN), false positive (FP), and false-negative (FN) of our attack as defined as:

- TP is the attack that drops a packet sent from RSU.
- TN is the attack doesn't drop a packet sent from OBU.
- FP is the attack that drops a packet sent from OBU.
- FN is the attack doesn't drop a packet sent from RSU.

We use false positive rate (FPR) to evaluate the reliability of the attack, where $FPR = \frac{FP}{FP + TN}$;

True positive rate (TPR) is used to evaluate the sensitivity of the attack, where $TPR = \frac{TP}{TP + FN}$;

[2] The attacker is able to prevent the DSRC application from telling a vehicle to stop. This causes the vehicle to run into the victim,

The effect of the attack is presented by the crash proportion $p_c = \frac{crash\#}{drop\#}$. The rates are shown in Table 7.

Table 7: An attack simulation analysis

Scenario	Attack target state	Crash proportion (%)	FPR (%)	TPR (%)
1	Control group	22.74	0	100
2	Max. likelihood timing state	11.46	2.50	28.48
3	Any target timing state	16.14	3.05	35.81
4	Target size state	13.74	9.45	39.53
5	Any defined target state	16.69	9.29	45.09
6	Combination of a target timing state and a size state.	9.70	1.9	28.29

To evaluate different attack scenarios, we plotted a column chart of true positive rate (TPR), false-positive rate (FPR), and crash proportion with the confidence interval (CI) for each scenario. The confidence interval is calculated by $p \pm Z_{1-\alpha}\sqrt{\frac{p(1-p)}{N}}$, where $Z_{.95} = 1.96$. The results are shown in Figure 13.

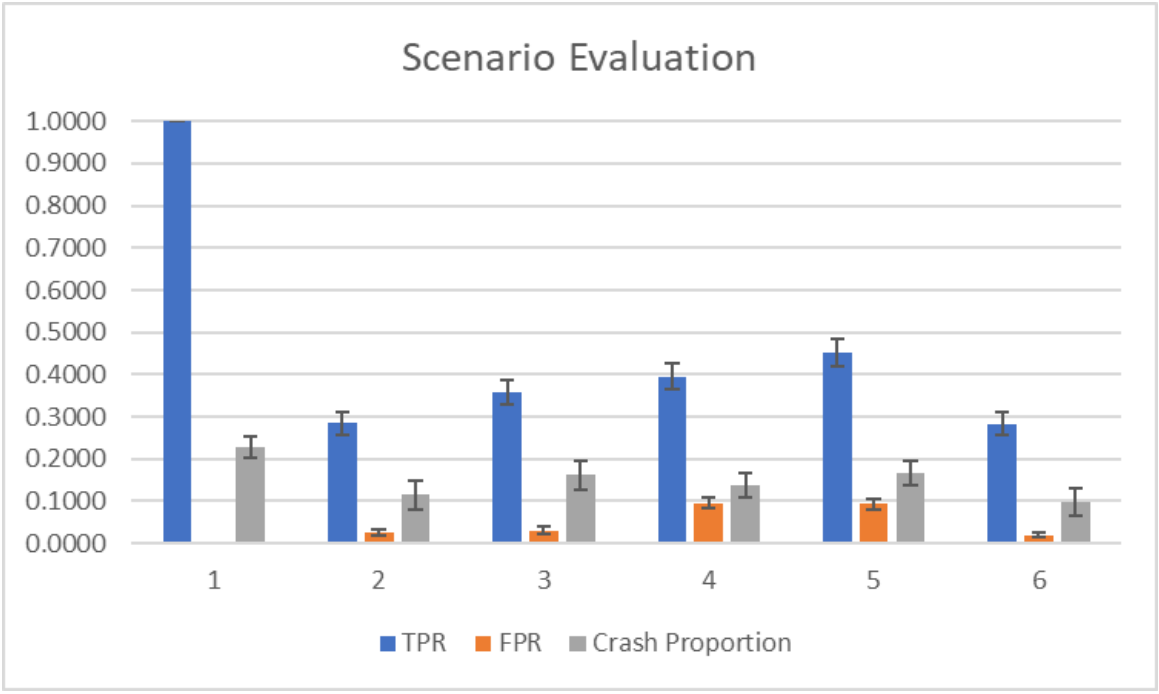

Figure 13: Comparison of attack strategies

As shown in Figure 13, the control group is the first group of bars where the TPR is 100%, FPR is 0%, and crash proportion is 22.74%. This means that each RSU packet dropping has about 22.74% of a crash if the attacker only drops RSU packets and does not drop all RSU packets. The goal of the side-channel analysis is to cause vehicle crashes with fewer unnecessary packets dropping. Each scenario made a targeting attack and caused crashes. The FPR is less

than 10% for all scenarios. The effectiveness of side-channel analysis is proved. We evaluate the attack scenarios based on crash proportion and FPR.

We first compare the attacks based on one type of side-channel information: timing side-channel attack for the most likely state, timing side-channel attack for two most likely states, and size side-channel attack. As shown in Figure 13, the 2nd and 3rd scenarios have the lowest FPR in the 2nd, 3rd, and 4th scenarios. With the windows of a confidence interval, there's no significant difference of FPR between the 2nd and 3rd scenarios. Moreover, the 3rd scenario also has the highest crash proportion. So we can conclude the 3rd scenario of timing side-channel attack for two most likely states is best in side-channel analysis based on one type of information.

Then, we compare all attack scenarios to find the best attack method for this application. As shown in Figure 13, the 3rd scenario and the 5th scenario have the highest value of the crash proportion, while the third scenario has a much lower FPR than the fifth scenario. So the 3rd scenario is the best in five attack scenarios.
In conclusion, timing side-channel analysis has better performance on predicted states. The attack targeting the packet leaving two most likely timing states worked best to cause crashes while avoiding dropping unnecessary packets.

CHAPTER 5

Conclusions

This project focuses on the evaluation analysis of DSRC/WAVE applications. To do this, we set up a DSRC stop light application based on a developed WSMP implementation. We sniffed the data through WSMP. The sniffed result of clear-text WSMP data content shows the current implementation is insecure. Lack of security services, such as content encryption, makes it easy for attackers knowing critical car/road information with DSRC equipped devices. Then we performed a DoS attack and successfully dropped packets at the communication channel and caused crashes.

Assuming all the security services will be implemented in the future, we completed a “black box” attack. Hidden Markov Models (HMM) are constructed using sniffed inter-packet timing and packet size side channels since operational packets would be encrypted. We set up an attack simulation to test the HMM predictions of important packet arrival. The simulation results show the effect of the side-channel analysis. Timing side-channel analysis worked better than the packet size side-channel analysis in the attack experiments.

The DoS result of packet dropping shows neither the application nor WSMP has a detection or prevention mechanism for DoS attacks. In DSRC communication, entropy-based DoS detection could be a good tool against DoS attacks. In DoS attack detection, entropy measures the amount of disorder in the observed data. For example, in this application, the road side unit (RSU) system could calculate the entropy value of packet rate and packet size. The RSU can also detect abnormal network traffic from the vehicle by cooperating with other RSU nearby. The vehicle volume could be estimated according to the information from other RSU. DSRC should add the authentication mechanism to the standard to prevent a DoS attack.

To prevent side-channel attacks, the WSMP of DSRC should improve the packet formatting. For example, it could define the length of a packet through WSMP to prevent packet size side-channel attack.

In the future, we will do the following work:

1. Collect more data and conduct the joint side channels analysis;
2. Apply this evaluation approach on more DSRC applications;
3. Test other attack methods, e.g., radio signal jamming.

REFERENCES

Al-Kahtani, M.S., 2012, December. Survey on security attacks in vehicular ad hoc networks (VANETs). In 2012 6th International Conference on Signal Processing and Communication Systems (pp. 1-9). IEEE.

Bettisworth, C., Burt, M., Chachich, A., Harrington, R., Hassol, J., Kim, A., Lamoureux, K., LaFrance-Linden, D., Maloney, C., Perlman, D. and Ritter, G., 2015. Status of the dedicated short-range communications technology and applications: report to Congress (No. FHWA-JPO-15-218). United States Department of Transportation. Intelligent Transportation Systems Joint Program Office.

Boukhtouta, A., Mokhov, S.A., Lakhdari, N.E., Debbabi, M. and Paquet, J., 2016. Network malware classification comparison using DPI and flow packet headers. Journal of Computer Virology and Hacking Techniques, 12(2), pp.69-100.

Brooks, R.R., Schwier, J.M. and Griffin, C., 2009. Behavior detection using confidence intervals of hidden Markov models. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(6), pp.1484-1492.

Brooks, R.R., Sander, S., Deng, J. and Taiber, J., 2009. Automobile security concerns. IEEE Vehicular Technology Magazine, 4(2), pp.52-64.

Brooks, R.R., 2013. Introduction to computer and network security: navigating shades of gray. CRC Press.

Craven, R., 2010. Traffic analysis of anonymity systems. All Theses. 837. https://tigerprints.clemson.edu/all_theses/837

Harakrishnan, B., Jason, S., Ryan, C., Richard R, B., Kathryn, H., Daniele, G., and Christopher, G., 2011. Side-channel analysis for detecting protocol tunneling. Advances in Internet of Things, 2011.

Islam, M., Chowdhury, M., Li, H., and Hu, H., 2018. Cybersecurity attacks in vehicle-to-infrastructure applications and their prevention. Transportation research record, 2672(19), pp.66-78.

Kenney, J.B., 2011. Dedicated short-range communications (DSRC) standards in the United States. Proceedings of the IEEE, 99(7), pp.1162-1182.

Laurendeau, C. and Barbeau, M., 2006, August. Threats to Security in DSRC/WAVE. In International Conference on Ad-Hoc Networks and Wireless (pp. 266-279). Springer, Berlin, Heidelberg.

Lee, H., Choi, K., Chung, K., Kim, J., and Yim, K., 2015, March. Fuzzing can packets into automobiles. In 2015 IEEE 29th International Conference on Advanced Information Networking and Applications (pp. 817-821). IEEE.

Lu, C., Schwier, J.M., Craven, R.M., Yu, L., Brooks, R.R. and Griffin, C., 2013. A normalized statistical metric space for hidden Markov models. IEEE transactions on cybernetics, 43(3), pp.806-819.

Ott, R.L. and Longnecker, M., 2001. Multiple comparisons, An introduction to statistical methods and data analysis (pp. 438–440). Australia: Duxbury Thomson Learning.

Ryan, P., Schneider, S.A., Goldsmith, M., Lowe, G. and Roscoe, B., 2001. The modeling and analysis of security protocols: the CSP approach. Addison-Wesley Professional.

Schwier, J.M., Brooks, R.R., Griffin, C. and Bukkapatnam, S., 2009. Zero knowledge hidden Markov model inference. Pattern Recognition Letters, 30(14), pp.1273-1280.

Schwier, J.M., Brooks, R.R. and Griffin, C., 2010. Methods to window data to differentiate between Markov models. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 41(3), pp.650-663.

Shalizi, C.R., Shalizi, K.L. and Crutchfield, J.P., 2002. An algorithm for pattern discovery in time series. arXiv preprint cs/0210025.

Song, Y., Stolfo, S. and Jebara, T., 2011. Markov models for network-behavior modeling and anonymization.

Sun, F., Brooks, R.R., Comert, G., and Tusing, N., 2022, Side-Channel Security Analysis of Connected Vehicle Communications Using Hidden Markov Models (Accepted). IEEE Transactions on Intelligent Transportation Systems.

Yang, D., Jiang, K., Zhao, D., Yu, C., Cao, Z., Xie, S., Xiao, Z., Jiao, X., Wang, S., and Zhang, K., 2018. Intelligent and connected vehicles: Current status and future perspectives. Science China Technological Sciences, 61(10), pp.1446-1471.

Yu, L., Schwier, J.M., Craven, R.M., Brooks, R.R. and Griffin, C., 2012. Inferring statistically significant hidden Markov models. IEEE Transactions on Knowledge and Data Engineering, 25(7), pp.1548-1558.

Zhong, X., Ahmadi, A., Brooks, R., Venayagamoorthy, G.K., Yu, L. and Fu, Y., 2015, March. Side-channel analysis of multiple pmu data in electric power systems. In 2015, Clemson University Power Systems Conference (PSC) (pp. 1-6). IEEE.

Zhong, X., Fu, Y., Yu, L., Brooks, R. and Venayagamoorthy, G.K., 2015, October. Stealthy malware traffic- not as innocent as it looks. In 2015 10th International Conference on Malicious and Unwanted Software (MALWARE) (pp. 110-116). IEEE.