Intrusion Detection System Based on the Analysis of Time Intervals of CAN Messages for In-Vehicle Network

Hyun Min Song, Ha Rang Kim and Huy Kang Kim
Center for Information Security Technologies (CIST), Graduate School of Information Security
Korea University
Seoul, Republic of Korea
signos@korea.ac.kr, rang0708@korea.ac.kr, cenda@korea.ac.kr

***Abstract*—Controller Area Network (CAN) bus in the vehicles is a de facto standard for serial communication to provide an efficient, reliable and economical link between Electronic Control Units (ECU). However, CAN bus does not have enough security features to protect itself from inside or outside attacks. Intrusion Detection System (IDS) is one of the best ways to enhance the vehicle security level. Unlike the traditional IDS for network security, IDS for vehicle requires light-weight detection algorithm because of the limitations of the computing power of electronic devices reside in cars. In this paper, we propose a light-weight intrusion detection algorithm for in-vehicle network based on the analysis of time intervals of CAN messages. We captured CAN messages from the cars made by a famous manufacturer and performed three kinds of message injection attacks. As a result, we find the time interval is a meaningful feature to detect attacks in the CAN traffic. Also, our intrusion detection system detects all of message injection attacks without making false positive errors.**

Keywords—car seucirty; controller area network; intrusion detection system

I. INTRODUCTION

Today, modern vehicles become smart, intelligent and connected. The proportion of electronic equipment in a vehicle was only 1% at the 1980s and increased to about 50% nowadays. We might consider vehicles as an electronic device or Internet of Things (IoT) device not only physical or mechanical device. As vehicles adopt more electronic components and implement connectivity functions to the external network, security threats on electronic equipment of vehicles are highly rising. Most of the smart devices such as smartphone, tablet and laptop computers can have security or privacy problems when they are compromised by malicious attacks. Unlike the usual smart devices, smart car (or connected car) can have one more critical problem when hacked. That is the safety problem that can seriously threat human's daily life. Therefore, we need to develop detection and prevention algorithms to react the emerging threats on vehicles.

A. Vehicle Networks

Vehicle network can be categorized by logical network location. One is the external network, and the other is internal network called as in-vehicle network. Also, vehicle network can be categorized by communication type, Vehicle-to-Vehicle (V2V) also known as Vehicular Ad hoc Network (VANET) and Vehicle-to-Infrastructure (V2I). For convenience, these communications altogether are usually called as V2X. V2X communications are used for safety driving by notifying information on the road to drivers.

There are various protocols for in-vehicle networks. Table I shows three well-known protocols for the in-vehicle network, Controller Area Network (CAN), Local Interconnect Network (LIN) and FlexRay. CAN is a serial bus to provide an efficient, reliable and economical link between Electronic Control Units (ECUs). CAN is used for vehicle's core control systems like body systems, engine management, and transmission control. LIN is a serial network protocol like CAN. LIN is developed as an alternative of CAN where low-cost implementation is required. LIN is usually used in the environment where communication speed is not critical. LIN is now a complement of the CAN within vehicles. FlexRay is designed to support faster and more reliable communication than CAN. FlexRay supports two-channel communication where CAN supports only single-channel. The maximum speed of CAN bus is 1 Mbps, where the maximum speed of FlexRay is 10 Mbps. FlexRay also supports flexible topology configurations like a bus, star, or hybrid topology.

TABLE I. CLASSIFICATION OF THE IN-VEHICLE NETWORK

Protocol	Description	Applications
CAN	- Multi-master, asynchronous serial network - Fast and reliable	Critical real-time communication (body systems, engine management, transmission, airbags)
LIN	- Single-master, multiple-slave serial network - Cheap and slow	Body control (door locks, seat belts, lighting, window, mirror)
FlexRay	- Next generation protocol - Fast and but more expensive	Multimedia and X-by-wire (drive-by-wire, brake-by-wire, steering-by-wire)

B. Security Threats on Vehicles

There are many security threats on vehicle electronic systems via variety access points such as V2X communication, telematics service, Bluetooth connection of mobile devices, and On-Board Diagnostics (OBD) port. We described some security threats as its attack surface. As described in Table II, vehicle security problem is not just about information security or privacy leakage. These security threats can affect the safety of the drivers directly.

TABLE II. SECURITY THREATS ON VEHICLE ELECTRONIC SYSTEMS

Attack surface	Security threats	Related to
Wireless communications (Telematics, Bluetooth, RF)	- Remotely vehicle control - Sensitive data leakage - Eavesdropping via microphone	Safety/Privacy/Security
Wireless communications (V2V/V2I)	- Abusing traffic signal control - Sending fake message - Polluting traffic information	Safety/Security(integrity)
Diagnostic interface (OBD, OBD-II)	- Execution of non-approval function - Injecting messages on CAN bus	Safety/Security(integrity, availability)
Infortainment system	- Unauthorized overall vehicle control	Safety/Security
Physical tampering	- Illegal tuning of engine - Odometer fraud - Usage of non-approval equipment	Safety/Security(integrity)

C. Organization of this paper

We introduced vehicle networks and security threats on vehicle electronic systems in Section 1. The rest of the paper is organized as follows. Section 2 presents the recent researches and projects about the vehicle security. We introduce our intrusion detection method for CAN bus traffic of vehicle network in Section 3. In Section 4, we describe the result of the experiment performed on the real vehicle. Finally, we discuss the experiment result and conclude the paper in Section 5.

II. RELATED WORKS

A. Recent Researches

Recently, Samy Kamal developed the hacking tool named Ownstar to hack GM's OnStar service. He successfully gained the system control authority of OnStar and controlled remote start, door, etc. [1]. Charlie Miller and Chris Valasek introduced their work on Jeep Cherokee via the wireless network. They took over full control of vehicle systems including steering, acceleration, brakes and turning off the engine at the remote side. [2]. They proved that an arbitrary vehicle can be controlled by remote attackers when attackers know the IP address assigned to the vehicle. Miller and Valasek also showed what hackers can do by injecting fake messages on CAN bus and suggested countermeasures of message injection attacks. They developed and publicly released the attack tool named as EcomCat, which helps to receive and transmit messages on CAN bus [3]. In fact, there were many attempts to hack a car before Miller and Valasek's work. Koscher et al. investigated practical security issues in vehicles on the road. They showed that they could take the control of vehicle systems like the engine, brakes, beating, and lights. A custom tool named as CARSHARK, which can analyze and inject messages on CAN bus, is used for experimental analysis [4]. Checkoway et al. categorized external attack surfaces of the vehicle. According to their category, there are four external attack surfaces of vehicle, OBD-II port as directly physical, CD and PassThru device as indirect physical, Bluetooth as short-range wireless, and Cellular as long-range wireless [5]. Verdult et al. found vulnerabilities in the Hitag2 transponders that enable to retrieve the secret key and can be abused to bypass immobilizer and start the vehicle [6]. Ishtiaq et al. introduced vulnerabilities of in-vehicle wireless networks through the case study of pressure monitoring system [7].

B. Research Projects on Vehicle Security

In this section, we summarized the recent research projects on vehicle security. Besides the listed projects below, many standards (e.g. ISO 26262 [21], a safety standard on road vehicles, and AUTOSAR [22], an open standard architecture) are continuously making efforts to enhance the vehicle security level. Well-known vehicle security projects are as follows.

- SeVeCom (Secure Vehicular Communication) defines the security architecture of inter-vehicular and vehicle-infrastructure communications, mechanism of security functions and cryptographic primitives required [8].
- While SeVeCom focused on attacks on external communication, EVITA (E-safety Vehicle Intrusion Protected Applications) focused on in-vehicle systems. EVITA developed an architecture and implemented Hardware Security Module (HSM) for automotive on-board networks to protect in-vehicle systems related to security and sensitive data [9].
- PRECIOSA (Privacy Enabled Capability in Co-operative Systems and Safety Applications) focused on privacy in V2X communication. They developed guidelines for Intelligent Transport System (ITS) privacy, trust models and ontologies for privacy, and privacy-verifiable architecture [10].
- OVERSEE (Open Vehicular Secure Platform) designed open platform that provides secure communication between in-vehicle network and applications. Secure Vehicle Access Service (SVAS) is used for secure communication. OVERSEE uses virtualization to isolate each workspace of applications and Security Policy Module to manage application's access to hardware [11].
- PRESERVE (Preparing Secure V2X Communication Systems) combines results from earlier research projects of European countries such as SeVeCom, PRECIOSA, EVITA and OVERSEE to provide a complete, scalable and cost-efficient solution for security problems related to communication systems connected to vehicles [12].

- VSCC (Vehicle Safety Communication Consortium) consists of 7 automobile manufacturers: BMW, DaimlerChrysler, Ford, GM, Nissan, Toyota, and VW. They developed vehicle safety service using Vehicle-to-Vehicle (V2V) communications and specified communication requirements of vehicle safety applications, including secure V2X communication [13].
- NoW (Network on Wheels) and CVIS (Cooperative Vehicle-Infrastructure Systems) designed communication protocols for V2X communications. While NoW focused on V2V and data security, CVIS focused on Vehicle-to-Infrastructure (V2I) and variety security issues such as user authentication and data privacy [14], [15].

C. *Intrusion Detection System (IDS) for Vehicle Network*

Traditional vehicles don't need to have a strong security system because they don't have a network interface to communicate with external networks. Therefore, CAN itself is like a closed network for a long time. Many components of the vehicles become computerized, and vehicles become connected to outside networks.

Vehicle security is closely related to safety. To detect and prevent the attacks is important to protect the safety of drivers and passengers. There have been several researches to detect attacks targeted on vehicles. Hoppe et al. [16] and Miller and Valasek [17] introduced a concept for in-vehicle intrusion detection based on the analysis of the rate of messages. Because the number of messages on CAN bus is the sum of numbers of normal messages and attack messages, they analyzed the distribution of rates of messages (messages per second) to detect anomalous message occurrences. Larson et al. proposed a specification-based attack detection method [18]. They detected the traffics not fit the protocol-level security specifications and ECU-behavior security specifications. Protocol-level security specifications define the individual fields, dependent fields, and inter-object fields of a message. ECU-behavior security specifications are about message transmission, message reception, and rates of message transmission and reception of each ECU. Muter and Asaj proposed an entropy-based anomaly detection method [19]. They defined the notion of entropy on CAN bus and detected the intrusion by comparing entropy to a reference set. Muter et al. [20] proposed a structured approach for anomaly detection. They use eight sensors to monitor variety aspects on CAN bus. Their method showed no false positive error. However, if adversary injects messages that do not violate CAN specification, then this attack cannot be detected by their algorithm.

Early researches about message rate based intrusion detection on CAN bus, need to collect enough amount of CAN bus messages to compute the distribution of a message. Thus, their detection methods need some time to detect anomalous messages. However, the current computerized devices in vehicles have limited computing power to detect and response in real-time.

To overcome this problem, we suggest a light-weight intrusion detection method. Our goal is simplifying detection algorithm to respond faster and to reduce the usage of computing power.

III. LIGHT WEIGHT IDS

A. *Threat model*

The proposed system is a hybrid IDS that can detect both of known attack signatures and anomalous events. The number of known attack signatures on a vehicle are relatively small; this signature-based detection module does not require high computing power. The proposed system is mainly designed to detect message injection attacks by analyzing traffic anomalies based on message frequency. As CAN is a broadcast network, messages sent by one of the nodes do not contain its source or target information. Also, these messages cannot be manipulated or eliminated easily. But, an attacker can still inject messages into CAN bus to control electronic devices such as ECU. Fig. 1 shows the conceptual diagram that describes the difference between the status under the message injection attack and a normal status.

In normal status, each message ID (0x1, 0x2, …) generated by ECUs has its own regular frequency or interval. When attackers try to inject messages to execute a command to an ECU, then this frequency or interval is unexpectedly changed. While messages being injected by attackers, ECUs still send their messages cyclically. Eventually, the rate of messages on the network can be increased more than two times (typically 20 – 100 times higher; it depends on the attacker's injection speed).

We select the message rate as a significant feature for the proposed detection method, and that is effective. But, there is a gap in time between the time of attack started and the time of detection. For example, if we set the time window as one second to observe and calculate the rate, there is always one-second gap at max. Even though attackers begin attacks at 0 seconds, we have to receive the attack packets until the minimum time window pass required to calculate the rate.

As the other statistical methods, small size of observational data can cause an error to make a decision. But attacks happened at anytime; this false-negative error can cause serious accident. To solve this problem, we simplify the process of detecting message injection to get the fast response while accuracy keeps high.

There are two forms of CAN injection attacks. The one is injecting CAN diagnostic messages, and another one is injecting standard messages to intimate the messages from ECUs. In general, diagnostic messages should not appear when a car is on a road. If this diagnostic message happens on the road, then that is obviously attack or system malfunction case.

We divided message injection attacks into three types for experiments. Type 1 is injecting messages of single CAN ID, type 2 is injecting random or pre-ordered messages of multiple CAN IDs, and type 3 is massively message injection such as Denial of Service (DoS) attack. These attacks are basically similar but different on their purpose. Details of three types of injection attack and countermeasures are following.

- Type 1: Injecting specific messages of single CAN ID repeatedly to make vehicle operate according to injected messages. We could detect the type-1 attack by finding a message that have shortened time interval abnormally.
- Type 2: Injecting random or pre-ordered messages of multiple CAN IDs to cause a system malfunction on a vehicle. A replay attack is one of type-2 attack based on pre-ordered messages injection. We could detect the type-2 attack by finding multiple CAN IDs that have shortened time interval than normal.
- Type 3: Injecting massages massively to disrupt CAN communication. An attacker can easily generate the traffic to surpass the maximum capacity of CAN bus, only 1 Mbps. Each CAN message have 128 bits maximally, and there are three 1-bits called interframe space between messages. Thus, DoS attack can occur by sending about 8,000 messages per second.

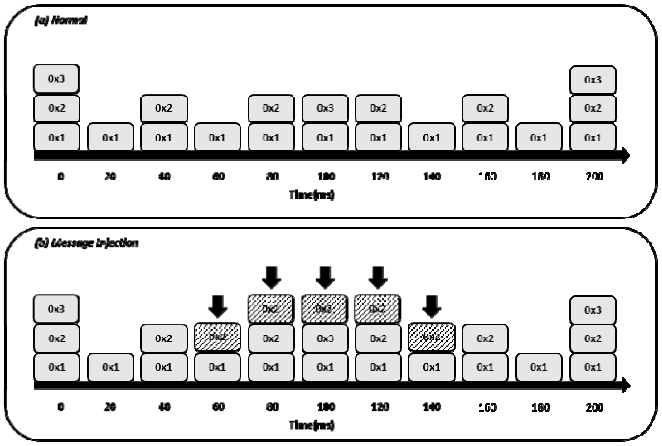

Fig. 1. Conceptual diagram about transmitted messages on CAN bus on (a) normal status and (b) under message injection attack. As shown in the figure, there are three CAN IDs, 0x01, 0x02, and 0x03. The time interval of 0x01 is 20, of 0x02 is 40, and of 0x03 is 100 milliseconds. There are five injected messages by attacker in (b) every 20 milliseconds from 60 to 140 millisecond The time interval of 0x02 falls rapidly lesst than 10 milliseconds from 20 milliseconds

B. *Intrusion Detection*

There is the unique time interval of each CAN ID because each ECU connected to CAN bus sends messages regularly. We focused on this fact and designed our IDS based on the analysis of time intervals of messages. The proposed system detects message injection attacks with the following procedure.

- When a new message appears on CAN bus, IDS checks the CAN ID and computes the time interval from the arrival time of the latest message.
- If time interval of a new message is shorter than normal, then IDS judges the message as an injected message. (In this experiment, we regard a message as an injected message when the time interval is below the half of the normal.)
- Especially if time intervals of latest messages in a row are less than 0.2 milliseconds, then DoS attack score increased by 1 per message.
- IDS classifies that event as a DoS attack when the score is larger than a given threshold.

The average time interval of messages on normal status is about 0.5 milliseconds and minimum time interval is about 0.14 milliseconds. Because there are some normal messages that have time intervals less than 0.2 milliseconds, a threshold is used in DoS attack detection to reduce the false positive ratio. We described the process of proposed system in Fig. 2.

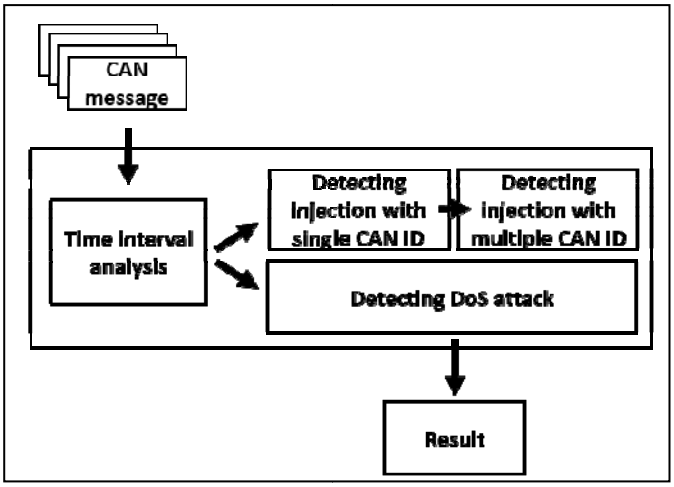

Fig. 2. Diagram of proposed IDS. After analysis of time interval of each message, there is two part of the detection module. The one is detecting injection of messages for controlling or malfunction. Another one is detecting DoS attack to disturb CAN communication.

IV. EXPERIMENT

A. *Dataset*

K-car (anonymized for protecting sensitive information) is used as a testing vehicle, and KVASER CAN interface is used to connect to CAN bus. Connecting the laptop computer to OBD-II port is shown as Fig. 3. OBD-II port of K-car is under the steering wheel.

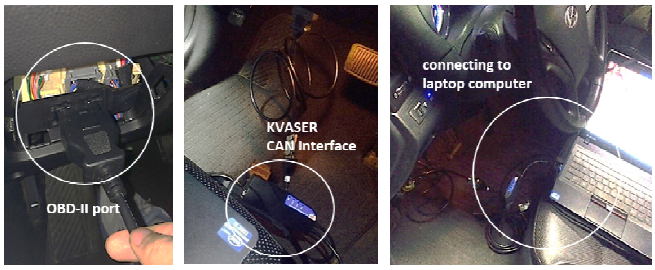

Fig. 3. Photos about connecting to OBD-II port of K-car with a laptop computer using KVASER CAN interface device.

We captured messages from CAN bus on normal speed driving for about 40 minutes. We injected messages 30 times for 5-10 seconds randomly for each attack. Types of attacks are injecting messages of a single CAN ID, multiple CAN IDs, and massively for DoS. After that, we performed random sampling to get a hundred 1-minute samples mixed with the under-attack

and normal status. We divided samples into two status which are containing injected messages (under-attack status), and the clean (normal status). Details about the dataset used in each experiment are described in the next section.

B. Experiment Result

First, we injected messages of a randomly selected single CAN ID with double, quintuple, and decuple faster than origin cycle. Previous research [17] also mentioned that an attacker should send messages 20-100 times faster than the original ECU to make the target ECU listens to the injected messages.

As described in subsection A, we created 43 attack samples and 57 normal samples in double speed injection, 39 attack samples and 61 normal samples in quintuple speed injection, and 35 attack samples and 65 normal samples in decuple speed injection. In all case, our IDS detected message injection attacks with 100% accuracy, and there is no false positive error.

Fig. 4 (a) and Fig. 4 (b) show the time intervals of the selected CAN ID at normal status and message injection status, respectively. Messages are injected decuple faster than the own cycle of the CAN ID. Therefore, the time interval of injected messages is less than 10% of the original interval. We injected message two times. The first injection started at 7 seconds continued for about 3 seconds. The second injection started at 16 seconds and ended at 17.6 seconds. There is a clear difference of time intervals between the normal status and under-attack status.

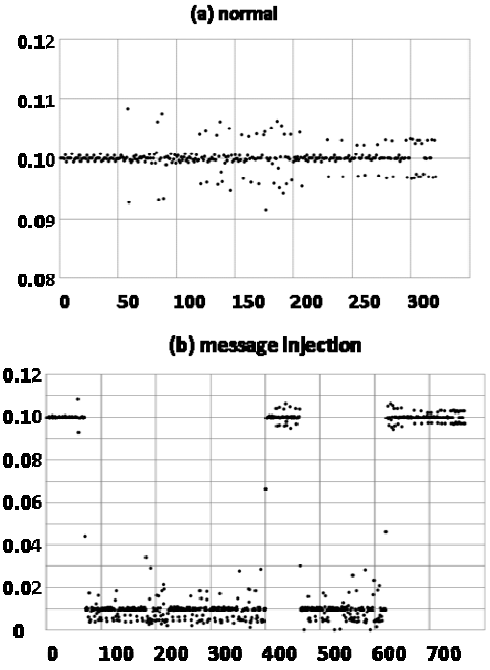

Fig. 4. Time intervals of messages of a certain CAN ID. Each point represents an order and time interval of a message. The X-axis is message generation number, and the Y-axis is a time interval of a message. For example, the first message is generated at 0.05715 seconds and the second message is generated at 0.15717 seconds. So, the second point in (a) is at (2, 0.10002)

Second, we injected messages of randomly selected 2-5 CAN IDs with double, quintuple, and decuple than original speed. The difference with the first experiment is just the number of CAN IDs of the injected messages. We created 39 attack samples and 61 normal samples in double speed injection, 44 attack samples and 56 normal samples in quintuple speed injection, and 39 attack samples and 61 normal samples in decuple speed injection. Our IDS classifies the all attack status and normal status samples with 100% accuracy as well as the first experiment. Table III shows result of experiment I and II. As mentioned above, we successfully detected message injection attacks in all the cases.

TABLE III. DETECTION ACCURACY

Injection Type	Injection Speed	Attack samples	Normal samples	Detection accuracy
Single CAN ID	Double	43	57	100 %
	Quintuple	39	61	100 %
	Decuple	35	65	100 %
Multiple CAN ID	Double	39	61	100 %
	Quintuple	44	56	100 %
	Decuple	39	61	100 %

At last, we tested DoS attacks on CAN bus by injecting messages massively. There are about 2,000 messages per second at the normal status; attackers can do DoS attack on CAN bus by sending about 6,000 messages more per second. We set the cutoff of the time interval to 0.2 milliseconds for detecting DoS message. As mentioned in section 3, there are messages that have time interval less than 0.2 milliseconds at the normal status but not often. We removed the false positive error by using a scoring method. We increased DoS attack score by 1 per message when the latest messages in a row which have time interval less than 0.2 milliseconds. Then we reset the score when the time interval of the latest message is larger than 0.2 milliseconds.

We created 36 DoS attack samples and 64 normal samples. We used 1, 2, 3 and five as the threshold value to measure the detection accuracy for each case. Fig. 5 shows the results according to the threshold value. When the threshold value is 1, detection accuracy is only 36 percent. Because there are messages with time interval less than 0.2 milliseconds even though there was no DoS attack, so all samples regarded as attack sample. However, detection accuracy is increased as threshold becomes larger, especially at 3 to 93 percent and 100 percent over 3. Our IDS just requires less than one millisecond to detect the DoS attack since DoS attack begins. It is fast enough to avoid an accident caused by DoS attack.

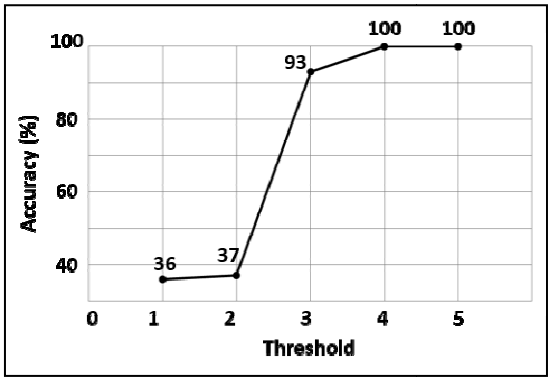

Fig. 5. Detection accuracy of DoS attacks according as threshold values. Only 36 percent when the threshold value was 1. Increased to 100 percent using threshold value over 4.

V. Conclusion

We showed that there was a clear difference between time intervals of messages in the normal status and under-attack status in section 4. Time intervals of specific CAN ID in normal were about 0.1 seconds. In contrast, time intervals under injection attack status became short (almost 10% of the normal time interval).

Therefore, we propose the light-weight IDS based on analysis of time intervals of CAN messages for in-vehicle networks. This system can successfully detect message injection attacks in a millisecond.

In spite of simplicity of detection algorithm, our IDS shows the improved performance than previous intrusion detection methods such as message rate based IDS. We significantly reduce the delay of detection that can cause a big accident when a vehicle is driving on a road with high speed. Also, the proposed IDS shows 100 percent detection accuracy without false positive errors in three kinds of message injection experiment.

The strength of the proposed detection algorithm is simple and efficient to use. So, our IDS is well fit the most vehicles that have limitations of computing power.

A. Limitations and Future Works

In future work, we will analyze the CAN message sequence to detect irregular message incomings. This sequence analysis requires more computing power, but it can improve the detection accuracy by using the known message sequence patterns as a white-list.

Acknowledgment

This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-TB1403-00.

References

[1] S. Kamal, OwnStar: Locates, Unlocks, Remote Starts GM/OnStar Cars.. 2015.

[2] C. Miller and C. Valasek, “Remote exploitation of an unaltered passenger vehicle” in BlackHat USA, 2015.

[3] C. Miller and C. Valasek, “Demo: Adventures in automotive networks and control units,” in DEFCON, 2013.

[4] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental Security Analysis of a Modern Automobile," in Proc. of the 31st IEEE Symposium on Security and Privacy, 2010, pp. 447-462.

[5] S. Checkoway , D. McCoy , B. Kantor , D. Anderson , H. Shacham , S. Savage , K. Koscher , A. Czeskis , F. Roesner and T. Kohno "Comprehensive experimental analyses of automotive attacksurfaces", Proc. 20th USENIX SEC, pp.6 -6 2011

[6] Roel Verdult, Flavio D. Garcia, and Josep Balasch. Gone in 360 seconds: Hijacking with Hitag2. In 21st USENIX Security Symposium (USENIX Security 2012). USENIX Association, 2012.

[7] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study," in Proc. of the 19th USENIX Security Symposium, Aug. 2010.

[8] Sevecom.org, ‘Secure of future vehicle communication networks', [Online]. Available: http://www.sevecom.org/. Accessed on: Sep 12, 2015.

[9] Evita-project.org, 'EVITA', 2008. [Online]. Available: http://www.evita-project.org/. Accessed on: Sep 12, 2015.

[10] Preciosa-project.org, [Online]. Available: http://www.preciosa-project.org/. Accessed on: Sep 12, 2015.

[11] Oversee-project.com, 'Oversee'. [Online]. Available: https://www.oversee-project.com/. Accessed on: Sep 12, 2015.

[12] Preserve-project.eu, ‘Preparing Secure V2X Communication Systems'. [Online]. Available: https://www.preserve-project.eu/. Accessed on: Sep 12, 2015.

[13] H. Krishnan, Vehicle Safety Communications Project Vehicle Safety Communications Project., 2006. [Online]. Available: http://www.sae.org/events/ads/krishnan.pdf. Accessed on: Sep 12, 2015.

[14] A. Festag , G. Noecker , M. Strassberger , A. Lübke , B. Bochow , M. Torrent-Moreno , S. Schnaufer , R. Eigner , C. Catrinescu and J. Kunisch "‘NoW—Network on Wheels’: Project objectives, technology and achievements", Proc. WIT, pp.123 -128 2008 [online] Available: http://www.network-on-wheels.de.

[15] Cvisproject.org. [Online]. Available: http://www.cvisproject.org/. Accessed on: Sep 12, 2015.

[16] T. Hoppe, S. Kiltz, and J. Dittmann. Security threats to automotive CAN networks - practical examples and selected short-term countermeasures. In SAFECOMP, 2008.

[17] Charlie Miller and Chris Valasek, A Survey of Remote Automotive Attack Surfaces, BlactHat USA, 2014.

[18] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An Approach to Specification-based Attack Detection for In-Vehicle Networks," in Proc. of the IEEE Intelligent Vehicles Symposium, 2008, pp. 220-225.

[19] M. Muter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks, " in Intelligent Vehicles Symposium (IV). Baden Baden, Germany: IEEE, 2011, pp. 1110-1115.

[20] M. Muter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks, " in 6th Int, Conf. Information Assurance and Security (IAS). Atlanta, GA: IEEE, 2010, pp. 92-98.

[21] ISO 26262, “Road Vehicles – Functional Safety”, [Online]. Available: http://www.iso.org/iso/catalogue_detail?csnumber=43464. Accessed on: Sep 12, 2015.

[22] AUTOSAR, [Online]. Available: http://www.autosar.org. Accessed on: Sep 12, 2015.