Cyber Security of Vehicle CAN bus

Jouni Pöyhönen, Pyry Kotilainen, Janne Poikolainen, Janne Kalmari, Pekka Neittaanmäki
University of Jyväskylä, Finland
jouni.a.poyhonen@jyu.fi
pyry.kotilainen@jyu.fi
janne.poikolainen@jyu.fi
janne.kalmari@jyu.fi
pekka.neittaanmäki@jyu.fi

Abstract: There are currently many research projects underway concerning the intelligent transport system (ITS), with the intent to develop a variety of communication solutions between vehicles, roadside stations and services. In the near future, the roll-out of 5G networks will improve short-range vehicle-to-vehicle traffic and vehicle-to-infrastructure communications. More extensive services can be introduced due to almost non-delayed response time. Cyber security is central for the usability of the services and, most importantly, for car safety. The Controller Area Network (CAN) is an automation bus that was originally designed for real-time data transfer of distributed control systems to cars. Later, the CAN bus was developed as a universal automation system for many automation solutions. One of its characteristics is that bus traffic is not supervised in any way due to the lack of timing of control. In other words there are no authentication mechanism. This article highlights different approaches and their usability to reveal the car's CAN bus malfunctions. The study complements earlier studies on the safety of vehicles in the CAN bus. Based on the test results, practical methods can be evaluated to detect changes in CAN bus traffic, such as targeted cyber-attacks. The article is based on the results of a study on the cybersecurity of cars conducted at the University of Jyväskylä (AaTi study). Initially, the AaTi study attempted to identify the message content of the bus and to detect interferences via the Neural network solution. However, the problem with the neural network was the computational performance required and the lack of prediction accuracy. After that the study was focused on experiments that were based on the arrival times of control messages, that is, their timing-based intrusion detection. In this sense the research did concentrate on kernel density estimation, one-class support vector machine solution, absolute deviation method and categorization. Due to methodological challenges, a method for detecting intrusions based on statistical processing of message traffic was ultimately developed as an outcome of the study.

Keywords: cybersecurity, car, CAN bus, intrusion detection

1. Introduction

The term intelligent transport systems (ITS) refers to using roadside infrastructure and communication solutions for improving traffic flows and making traffic safer. In order to realize the perquisites for smart traffic, current national and international research projects are focusing on the development of platforms for weather, security and geolocational solutions. These include test environments for real-time road weather reports based on location data as well as for ITS cyber security. (Finnish Meteorological Institute, 2017)

Service usability is closely linked to cyber security, in which taking care of vehicle cyber security can be seen as a primary objective. CAN bus is a network solution originally developed for real-time communication in distributed automotive control systems, such as in engine control units, ABS brakes and drivetrains. (Alanen, 2000)

CAN bus later evolved as a general-purpose automation solution to accommodate other use cases in addition to automotive use. The real-time requirement makes minimizing network delays one of the main principles of CAN-bus functionality. This optimization also leads to design decisions that excluded many safety mechanisms, including authentication. These features make CAN bus implementations vulnerable to several types of attacks, including network traffic forgery, unauthorized access to data and denial of service attacks. As the growing use of automation means also the growing use of network connectivity, the attack surfaces in vehicles can be divided into two groups: surfaces that can be exploited remotely and surfaces requiring physical access. Because of development of intelligent transport systems and smart traffic the need of remote connections will grow even more in the future as ITS develops further. Vehicle network security research has emerged in past years, especially after the inherent vulnerabilities in commonly used technologies have been realized.

The purpose of this article is to present different approaches and their abilities to detect anomalies in vehicle CAN buses. Based on the results of this study, methods plausible for real-world scenarios are proposed. The ultimate goal of the study has been to develop real-time situational awareness methods for automation systems.

This report is based on a study (AaTi) conducted at the University of Jyväskylä, which concluded on 30 September 2018.

In addition to the chapters dealing with introduction and the CAN bus description, the paper includes a short description from other relevant vehicle studies and explanations from the methods used in the AaTi study to detect harmful bus traffic and the results obtained from their use. The conclusion chapter includes the summary of the AaTi study.

2. The CAN bus

2.1 The CAN Standard

The CAN communications protocol, ISO-11898: 2003, describes how information is passed between devices on a network and conforms to the Open Systems Interconnection (OSI) model, which is defined in terms of layers. Actual communication between devices connected by the physical medium is defined by the physical layer of the model. The ISO 11898 architecture defines the lowest two layers of the seven-layer OSI/ISO model as the data-link layer and the physical layer, shown in Figure 1 (Corrigan, 2016).

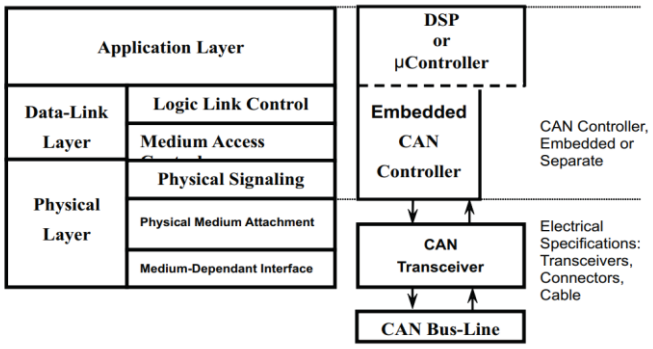

Figure 1: CAN bus in the OSI/ISO model

The application layer establishes the communication link to an upper-level application specific protocol such as the vendor-independent CANopen™ protocol. This protocol is supported by CAN in Automation (CiA), the international users and manufacturers group. Many protocols are dedicated to particular applications, such as industrial automation, diesel engines, or aviation. (Corrigan, 2016)

2.2 CAN message and frames

The four different message types, or frames (see Figure 2, CSS Electronics, 2018), that can be transmitted on a CAN bus are the data frame, the remote frame, the error frame, and the overload frame.

Figure 2: CAN bus message

CAN bus frames: (Corrigan, 2016)

The data frame

The data frame is the most common message type, and comprises the arbitration field, the data field, the CRC field, and the acknowledgment field. In Figure 2 the arbitration field contains a 29-bit identifier (or 11-bit identifier) and the RTR bit, which is dominant for data frames. Next is the data field, which contains zero to eight bytes of data, and the CRC field, which contains the 16-bit checksum used for error detection. The acknowledgment field is last.

The remote frame

The intended purpose of the remote frame is to solicit the transmission of data from another node. The remote frame is similar to the data frame, with two important differences. First, this type of message is explicitly marked as a remote frame by a RTR bit in the arbitration field, and second, there is no data.

The error frame

The error frame is a special message that violates the formatting rules of a CAN message. It is transmitted when a node detects an error in a message and causes all other nodes in the network to send an error frame as well. The original transmitter then automatically retransmits the message. An error mechanism in the CAN controller ensures that a node cannot tie up a bus by repeatedly transmitting error frames.

The overload frame

The overload frame is mentioned for completeness. It is similar to the error frame with regard to the format, and it is transmitted by a node that becomes too busy. It is primarily used to provide for an extra delay between messages.

2.3 CAN bus arbitration

Arbitration is a mechanism for conflict resolution between network nodes. When the network path is free, any of the nodes in the network can start the message send process. If another node also wishes to send at the same time, the order of the transmissions is decided using a bitwise arbitration mechanism. During arbitration, both nodes start their transmission. The transmission starts with a start bit, followed by an id field (identifier, CAN-ID). The sending order decision is made based on the value of the id field and the other node or nodes discontinue their transmissions. The messages are sent ordered by priority, where the zero value is dominant. In practice this means that if a node currently sending a bit with a value of one sees that another node is sending a zero bit, it backs off. In other words, it discontinues its own transmission, forfeiting its turn to the node sending the dominating bit. In practice the message with the smallest decimal id value has the highest priority. (Johansson et al., 2005)

From the viewpoint of attacks, this mechanism enables denial of service attacks. As an example, sending large quantities of forged messages having an id value of zero.

2.4 CAN bus pros and cons

CAN bus was designed for maximal speed and reliability. At the technical level this mean, among other aspects, that the network communication uses a provider–consumer model instead of the common sender–receiver model. The second feature aiming for performance gains was the lossless bus arbitration described above. (Voss and Comprehensible, 2005)

Improving the reliability of the data transmitted between the nodes was achieved with a mechanism that insures the integrity and timeliness of the messages. These mechanisms are based on bus arbitration, using checksums checking the payload and resending failed messages. (Voss and Comprehensible, 2005)

Based on these design decisions, CAN bus is effectively a broadcast network, where any node can send a message and all nodes are listening to the network and reacting to the messages they are interested in. The only thing the recipients check is the protocol correctness of the received message. (Voss and Comprehensible, 2005)

CAN bus speed is 1 Mbit/sec, which these days does not seem fast. Yet for transmitting short messages and having an effective collision avoidance mechanism, CAN bus is more suitable to be used in real-time applications than connected protocols such as TCP/IP, even if those would be using greater transmission speeds. (Voss and Comprehensible, 2005)

With further development the CAN bus has became a dominant technology for the data transmission of vehicle basic functions. During the last two decades the number of electronic systems in vehicles has increased and at

the same time they have become more complex. CAN bus vulnerabilities can be traced back to design decisions described above, the most significant of these being the lack of authentication mechanism. The receiving entity does not have any mechanism to verify the origins of the received message or the validity of the data received. In other words, the control unit does not have a mechanism to detect message forgery. This characteristic makes vehicle CAN busses vulnerable to attacks, such as message forgery, unauthorized data use and denial of service. The DoS vulnerability can be exploited by sending a large number of high priority messages. These attacks can affect the vehicles systems in such a way as to cause loss of control, incorrect functionality, premature wear or rendering the vehicle unable to function at all. (Carsten et al., 2015)

2.5 Attack surfaces

The taxonomy of CAN bus attack surfaces is usually divided into two parts: remote exploits and exploits requiring physical access to the CAN bus. In addition to this, some researchers have expanded the use of physical connections by constructing experiments that enable man-in-the-middle type of attacks on the CAN bus (Lebrun and Demay, 2016).

Physical connection to a CAN bus is not technically complex to achieve. The simplest physical connection can be implemented through the vehicle's diagnostics port. This approach does not require any alterations to the vehicle in question. The limitation of this approach is the amount of network data observable at this point of entry, depending heavily on the make and model of the vehicle. CAN bus traffic seen through the diagnostic port is restricted by segmenting the network. These limitations can be avoided by choosing another point of entry from the desired segment. In most cases this approach requires alterations to the vehicle's wiring harnesses, because segment-specific connectors are rarely implemented in production vehicles.

Remotely exploitable attack surfaces that would have a direct effect on the vehicle's physical functionalities are usually more challenging to exploit. In practice, this normally means a multistage attack where the attacker first has to find a vulnerable and remotely accessible service to gain a foothold. As an example, this kind of service can be found from the vehicle telemetry or infotainment systems. After gaining a foothold on one of the connected systems, the attacker needs to find a way to gain access to another system that has connectivity to the more critical segments of the vehicle's CAN bus. This type of attack has been successfully conducted by some vehicle security researchers (see Miller and Valasek, 2013).

3. The AaTi study

3.1 Previous research

Wolf et al. (2004) found that vehicle networks are open and for this reason vulnerable on many levels. The attacker can exploit vehicle wireless connections and networks. Wolf et al. (2007) continued their work in an article where they were attempting to form a full picture of the current situation of automotive electronic systems. This article listed commonly used automotive systems and their properties, including details about communication and cryptography.

The possibility of cyber-attacks as a subject of scientific articles became more prevalent around a decade ago. At that time, the articles started to touch on the subject of, among other things, how to protect vehicles for possible attacks (Larson et al., 2008).

A research group consisting of researchers from the University of Washington and the University of California, San Diego conducted a system security analysis through experiments on a passenger vehicle. This article was aiming for a comprehensive security analysis of a vehicle system rather than an analysis of individual devices. The article also proposed a part threat model that identified the physical connection and wireless functionalities as individual attack vectors. (Koscher et al., 2010) This group continued their work the next year by publishing an article, focusing on a broader analysis of the vehicle attack surfaces. (Checkoway et al., 2011)

Vehicle cyber security research was brought to more common knowledge by Valasek and Miller, who published their first article on this subject in 2013. In this article they examined two vehicles from different manufacturers and got results on how vehicle functionality can be affected that were similar to what previous academic research efforts had shown. In addition to their results, they published most of the reverse engineered CAN messages they discovered, and the source code of the tools used in their research. The additional information was

published to encourage other groups to conduct similar research in the future (Miller and Valasek, 2013). Miller and Valasek (2015) continued their work and published an article that describes in detail how an unaltered vehicle can be taken into partial control without a physical connection. As a point of entry to the vehicle system they used a security flaw in the infotainment system of the vehicle in question.

3.2 Analysis methods

3.2.1 Introduction to the analysis methods used in this research

The anomaly detection methods proposed in previous academic articles can be divided into groups using several different taxonomies. The first example of such a taxonomy is dividing the methods based on the use of system specification. When system specification is available, detecting anomalies is based on detecting traffic that does not fit the given specification. This type of approach has been suggested in the method used by Larson et al. (2008), where anomaly detection is delegated to the network nodes. The nodes then inspect the traffic and sound an alarm if they see some else sending a message type only they are supposed to send. The second category of systems assumes that system and message specifications are present. In these systems the anomaly detection is based on features such as message timing, data semantics, entropy, repeating message sequences, protocol correctness and signal characteristics differing from normal network traffic.

Anomaly detection methods can also be grouped according to their method of detection. The majority of normal CAN bus traffic is cyclic by nature. This means that a series of messages repeat cyclically after very predictable intervals. Based on this property many proposed methods use message timing as a basis for anomaly detection. Time-based detection methods can also be divided into two main groups: those that measure message frequency and those that measure interval. Methods that fall into the first group have been proposed by Hoppe et al. (2008-2009), Münter et al. (2010) and Miller and Valasek (2014). Miller and Valasek (2014) proposed a substantial rise in the frequency of sent messages for a detection method. Taylor et al. (2015) proposed a more advanced method where the frequency monitoring is based on Hamming distance.

However, methods based on message frequency have their weaknesses. For example, when only message frequency is monitored short-term anomalies are not necessarily detected. However, if instead of frequency the detection method is based on message intervals, even short-term changes in network traffic can be detected more accurately. The use of interval analysis has been proposed by Son et al. (2016) and Moore et al. (2017). The latter article proposes a method based on absolute time deviation.

Several methods based on correctness of data carried by the messages have been proposed. Hoppe et al. (2008) described a method where only gross abuse of messages is detected. They continued their work in 2009 by proposing a method where consecutive messages are monitored for semantic correctness.

Münter and Asaj (2011) described a method based on data entropy, where changes in entropy are detected on the binary level. In addition, their method monitored communication entropy in protocol and signal levels.

Methods based on monitoring repeating message sequences in a specified time window has been proposed in several articles. Narayanan et al. (2015) based their method on a hidden Markov model and Marchetti et al. (2017) observed the order and changes in repeating messages.

The AaTi project used a test vehicle (Toyota Prius Hybrid). The data used in the study was first recorded from a test vehicle. It could then be inspected in laboratory environment. The vehicle-specific CAN bus interference messages were first generated under laboratory conditions and then verified on a test vehicle. The first goal of AaTI study was the ability to distinguish between normal and abnormal network traffic in real-time using recorded samples from a vehicle. In this research, a system of message specifications was not used, so anomaly detection of data payloads proposed a challenge. Mainly for this reason most of the methods that were researched were time based. This has also been the primary approach for previous research.

The only method during this research that was based on anomaly detection in data payloads was a neural network that could learn to predict incoming message payloads based on previous data it had inspected.

3.2.2 Neural network

For inspecting message data payloads, a neural network was built based the method presented in Taylor et al. (2016). In this method the neural network builds a model based on normal data traffic by inspecting network traffic. The method described in the article has produced promising results. Different metrics for identifying and measuring deviations in the data streams have also been proposed in multiple articles (e.g., Taylor et al., 2015).

The Long Short-Term Memory (LSTM) network architecture used in this research consisted of layers of nodes with an adjustable feedback loop. This architecture enables the network to have a "memory" as well as the ability to "forget". The majority of data in CAN bus traffic is regular by nature, in other words the data changes gradually and follows distinct trends. Therefore, predicting should be viable for at least some parts of the network traffic. In the experimental design the neural network was constructed to predict the data bits of an incoming message based on data bits of previously observed messages.

The biggest problems using the described neural network was its resource demand and probable difficulties in making the predictions more accurate. In addition, reasonable accuracy can only be achieved in regular dataflows, so some parts of the CAN bus traffic cannot be inspected using this method.

3.2.3 Kernel density estimation

The first method implemented for interval analysis was kernel density estimation. This method can model interval distribution characteristics for each message identifier. The distribution characteristics can then be compared to incoming message distribution to detect anomalies.

Modeled distribution gives the intervals a density function that can be used to calculate reliability values for new messages. If the calculated reliability drops too low, the situation is declared an anomaly and an alarm can be given.

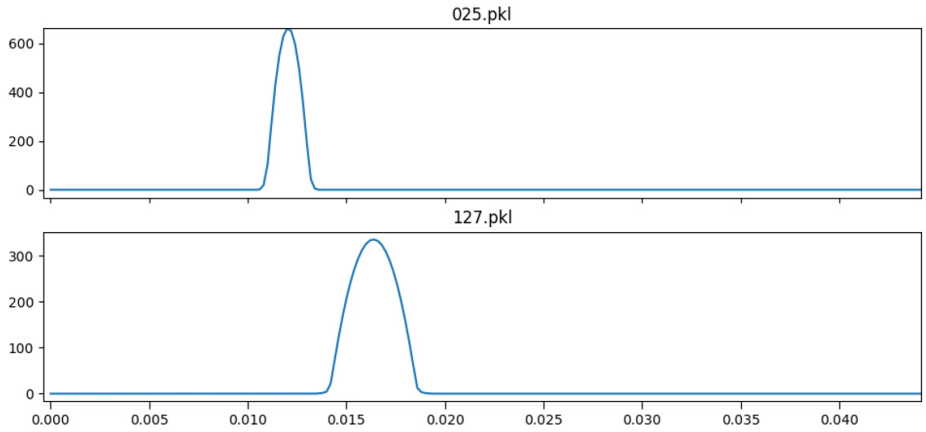

Figure 3: Arrival interval distributions of two different message identifiers.

Figure 3 shows arrival interval distributions for two message identifiers that have been modeled using kernel density estimation. In the first graph, the interval deviates between 10 and 15 milliseconds. In the second graph, it deviates between 15 and 20. If incoming traffic shows interval deviations to be different than the peaks showed in the graphs, an indication of anomalous traffic can be given.

The advantage of kernel density estimation is that it can model systems that implement different sending speeds. For example, an engine control unit can send messages with different intervals when the engine is in idle or when the vehicle is moving. This kind of situation would show in the model as two distinct peaks. However, this kind of behavior was not observed in the test vehicle used in the study.

3.2.4 One-class support vector machine

One-class support vector machine (OCSVM) is a variation of a support vector machine, which is a popular machine learning method for classification. However, a normal support vector machine requires examples of each class that it should identify. An OCVSM, on the other hand, classifies the elements into two categories: normal

and abnormal. This means examples of normal behavior are sufficient and it does not require examples of abnormal behavior. The method defines a boundary around normal behavior and classifies all messages outside this boundary as abnormal.

Therefore, a one-class support vector machine is fit for detecting abnormal behavior, because it is challenging to find examples of all possible abnormal behaviors for training. Examples of normal behavior, on the other hand, are in most cases easily available. Normal data sets were recorded from test vehicle.

Taylor et al. (2015) presented an application of OCVSM, and the AaTi study implemented a variant of this machine. A moving window containing a set number of messages was used as a data element. From the data elements the characteristics were calculated that could be used to define the whole inspected window as either normal or abnormal. Characteristics calculation was based on mean interval and standard deviation of the messages within a window.

3.2.5 Absolute deviation

Because the kernel density estimation method described above (see 3.2.3) is resource intensive and no multiple peaks were observed in the gathered data, a decision was made to implement a simplified method using the same principles. This method attempted to model the interval deviation for each message identifier, which would give considerable gains in performance and, at the same time, maintain similar performance for detection.

A decision was made to model the intervals with a normal deviation, because this made it possible to describe the deviation using only mean and standard deviation values. In the training phase it was also decided to include upper and lower bound values for each message identifier for classification. The distinct upper and lower bound values were added because positive deviations were more common in the normal network traffic, possibly due to message collisions. In addition, doing these calculations in the training phase made the classification faster. The boundary values were chosen so that no deviations were classified from the training data and a small marginal was added.

Again, a moving window was used for data-element as in the one-class support vector case. The messages in the window were classified based on its average interval. If this value was smaller than the lower bound or greater than the upper bound value then the traffic within the window in question was defined as abnormal.

An almost identical sensor was implemented in an article by Moore et al. (2017). The difference being that they did not use a moving window as a data element (Müter et al., 2010). An alert caused by a single abnormal message would produce too many false positives, so in the implementation described in the article only three consecutive abnormalities will trigger an alarm. In testing this method showed similar performance with other methods tested and it was less resource intensive. A decision was made to do a proof of concept implementation of this method.

3.2.6 Categorization

Based on the previously described methods, it was observed that most false positives originate from control units that send their messages in irregular intervals. For this reason, the possibility of categorizing the messages by their send profiles was examined. Some of the control units send messages at regular intervals and others send irregular messages of events between regular status messages. A simple absolute deviation detection would categorize these messages as abnormal and initiate an alarm.

Based on the observations made during the research. It would be possible to reduce the number of false positives using categorization. But the number of send profiles would pose challenges for an implementation of such a categorization method. In addition, some messages that have the same message identifier can use multiple send profiles. In addition to these two drawbacks, there is uncertainty over what kind of send profiles exist in addition to the ones observed.

3.2.7 Method comparison

Time-based methods were compared by drawing a receiver operating characteristic (ROC) curve for each individual method using the same data recorder from a vehicle CAN bus that included a test attack. All methods

used the same window size of five messages per window so that comparability could be maintained. All methods were given a setting that if they detected even one message that was part of the attack, they should mark the whole window as abnormal.

The best threshold value in Figure 4 shows the threshold value for which the best accuracy without any false positives was achieved. The number of true positives is shown in parentheses labeled “TPR” (True Positive Rate). The main interest in this figure should be the threshold value, since a practical real-time sensor would require a minimal number of false positives.

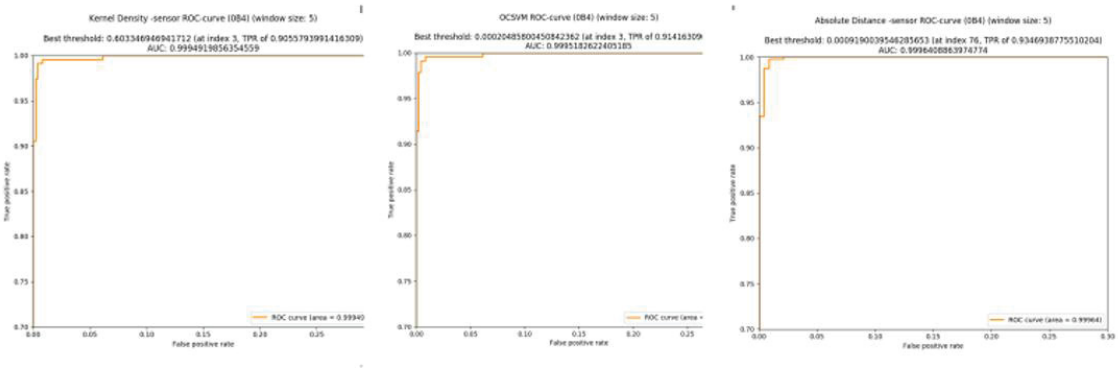

Figure 4: ROC graph (Vertical axis: true positive rate and horizontal axis: false positive rate)

Figure 4 left, kernel density estimation method; best achieved TPR without false positives: 0.9056. Figure 4 middle, OCSVM; best achieved TPR without false positives: 0.9142. Figure 4 right, absolute deviation method; best achieved TPR without false positives: 0.9347.

The comparison shows the analysis of a single message identifier attacked during the data recording. The results suggest that the performance of the methods shown is similar. At least with the test data ware used in the comparison. The absolute deviation, which is also the simplest method, achieved the most accurate results, maintaining a zero false positive rate.

Based on the observations, the methods described above show that most false positives originate from electronic controller units with irregular send profiles. Message send profile categorization could be used to improve the result in these cases, but it has its drawbacks, as described in section 3.2.6.

4. Conclusions

The focus of the AaTi study was to survey anomaly detection methods applicable to vehicle networks. This research complements previous research and patents by understanding network-traffic characteristics using recordings obtained from a test vehicle. The study shows that attacks against vehicle networks can be categorized into three groups. The network can be injected (a) with special messages such as diagnostics messages; (b) with normal messages that disturb vehicle functionality or (c) by sending normal messages after the real sender has been rendered unfunctional. The most common situation is probably when the real sender is still functional, and the attacker sends normal CAN messages. These kinds of attacks can be detected by observing message send intervals, since in a normal situation the intervals should remain regular.

In the first phase of research a neural network implementation was tested for its ability to detect abnormalities in message data payloads. The aim of this implementation was to provide technical means to learn different payload possibilities and predict the data incoming in the following messages. This would have created the possibility to detect abnormal data payloads. The problem with using neural networks arose from its resource intensiveness and lack of prediction accuracy. The next experiments focused on anomality detection methods based on message timing.

The first time-based method we tested was One-Class Support Vector Machine (OCSVM), which is a variation of the popular machine learning method. This method defines boundaries around normal behavior and classifies all other traffic as abnormal. In the implementation, a moving window with a set number of messages was used as a data-entity. The characteristics of the messages are then calculated using OCSVM and, based on the results,

the whole window is declared normal or abnormal. The characteristics used in this implementation were average interval and standard deviation.

After this fist experiment, other methods based on message interval were surveyed. Kernel density estimation models interval deviation for each message identifier. This value can then be compared to incoming messages in order to detect abnormalities. Modeled deviation provides a density function for the interval that can be used for likelihood value calculation for incoming messages. A drop in the calculated likelihood that exceeds a predetermined threshold can be detected as an anomaly and an alarm can be triggered. Because kernel density estimation is also a resource-intensive method and the observed test data did not show multipeak properties, a simplified version using the same principles of this method was implemented. This method aims to model message identifier deviation using key values. This implementation of absolute deviation achieves substantial gains in resource efficiency and without decline in the performance of the detection properties. The modeling was done using standard deviation in order to use the two key values: average and standard deviation. In the practical implementation training phase average, lower and upper bound values were calculated for each message identifier for classification purposes. A moving window was used as a data entity. If the values within the window went below the lower bound or exceed the upper bound, the whole window is declared an anomaly in the network traffic.

All of the above mentioned methods have their own challenges in either resource intensiveness, accompanied in some cases with inaccuracy of predictions.

Based on the experience described in the method comparison chapter, a novel method for detecting CAN bus anomalies based on message arrival intervals was developed and a patent application for this method has been filed. The description of this method is part of the patent. The functionality of this method was verified in a computational environment.

As different digital platforms become ever more common in automated processes, the protection of different processes and the cyber security of the infrastructure is going to play a significant role in the overall safety of these platforms. For future researchers in this field, the group would like to recommend the usage of outcomes found in the AaTi study as well as the utilization of the patented method as a part of future CAN bus implementations in order to improve cyber security.

References

Alanen J. (2000). CAN ajoneuvojen ja koneiden sisäinen paikallisväylä. Tampere: VTT Automaatio, koneautomaatio.

Carsten P., Yampolskiy M., Andel T.R. and McDonald J.F. (2015). In-Vehicle Networks: Attacks, Vulnerabilities, and Proposed Solutions. CISR '15 Proceedings of the 10th Annual Cyber and Information Security Research Conference (apr 2015), 477–482

Checkoway S., McCoy D., Anderson D., Kantor B., Savage S., Koscher K., Czeskis A., Roesner F. and Kohno K. (2011). Comprehensive Experimental Analysis of Automotive Attack Surfaces, in Proceedings of the USENIX Security Symposium, San Francisco, CA.

Corrigan S. (2016). Introduction to the Controller Area Network (CAN). Texas Instruments. http://www.ti.com/lit/an/sloa101b/sloa101b.pdf

CSS Electronics (2018). A Simple Intro to CAN Bus. https://www.csselectronics.com/screen/page/simple-intro-to-can-bus/language/en

Finnish Meteorological Institute, (2017). Intelligent Transport. https://ilmatieteenlaitos.fi/alykas-liikenne

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

Hoppe T., Kiltz S. and Dittmann J. (2009). "Applying Intrusion Detection to Automotive It-Early Insights and Remaining Challenges." Journal of Information Assurance and Security (JIAS) 4 (6): 226–235.

Johansson K. H., Törngren M. and Nielsen L. (2005), Vehicle applications of controller area network, in Handbook of Networked and Embedded Control Systems, William S. Levine Dmiitris Hristu-Varsakelis, and,ed., Birkhauser.

Koscher K., Czeskis A., Roesner F., Patel S., Kohno T., Checkoway S., McCoy D., Kantor B., Anderson D., Shacham H. and Savage S. (2010). Experimental security analysis of a modern automobile. In D. Evans and G. Vigna, editors, IEEE Symposium on Security and Privacy. IEEE Computer Society.

Larson, U. E., Nilsson D. K. and Jonsson E. (2008). "An Approach to Specification-Based Attack Detection for in-Vehicle Networks." In 2008 IEEE Intelligent Vehicles Symposium, 220–25. doi:10.1109/IVS.2008.4621263.

Lebrun A. and Demay J. C. (2016). Canspy: a platform for auditing can devices. https://www.blackhat.com/docs/us-16/materials/us-16-Demay-CANSPY-A-Platorm-For-Auditing-CAN-Devices.pdf.

Marchetti M. and Stabili D. (2017). “Anomaly detection of can bus messages through analysis of id sequences,” in 28th IEEE Intelligent Vehicle Symposium (IV2017).

Miller C. and Valasek C. (2013). Adventures in automotive networks and control units, DEFCON 21, Las Vegas, NV.

Miller C. and Valasek C. (2014). A survey of remote automotive attack surfaces, BlackHat USA.

Miller C. and Valasek C. (2015). Remote exploitation of an unaltered passenger vehicle, Black Hat USA.

Moore M. R., Bridges R. A., Combs F. L., Starr M. S. and Prowell S. J. (2017). “Modeling inter-signal arrival times for accurate detection of can bus signal injection attacks,” in 12th CISRC. ACM.

Müter M., Groll A. and Freiling F. C. (2010). “A Structured Approach to Anomaly Detection for in-Vehicle Networks.” In 2010 Sixth International Conference on Information Assurance and Security (IAS), 92–98. doi:10.1109/ISIAS.2010.5604050.

Müter M. and Asaj N. (2011). "Entropy-based anomaly detection for in-vehicle networks." IEEE IVS.

Narayanan S. N., Mittal S. and Joshi A. (2015). “Using Data Analytics to Detect Anomalous States in Vehicles.” arXiv Preprint arXiv:1512.08048. http://arxiv.org/abs/1512.08048.

Song H. M., Kim H. R. and Kim H. K. (2016). "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," in 2016 International Conference on Information Networking (ICOIN), pp. 63-68

Taylor A., Japkowicz N. and Leblanc S. (2015). “Frequency-Based anomaly detection for the automotive CAN bus,” in Proc. of WCICSS, 2015, pp. 45–49.

Taylor A., Leblanc S. and Japkowicz N. (2016). “Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks”. IEEE DSAA (2016).

Voss W, and Comprehensible A. (2005). Guide to Controller Area Network. Massachusetts, USA: Copperhill Media Corporation.

www.br-automation.com

Wolf M., Weimerskirch A. and Paar C. (2004). Security in automotive bus systems. In Proceedings of the Workshop on Embedded Security in Cars 2004.

Wolf M., Weimerskirch A. and Wollinger T. (2007). State of the art: Embedding security in vehicles. EURASIP Journal on Embedded Systems.

Marvin Newlin is a graduate student at the Air Force Institute of Technology studying to obtain an M.S. degree in Cyber Operations. Marvin graduated in 2014 from North Carolina State University with B.S. degrees in Mathematics and Computer Science.

Captain (Eng.), Dr. Juha-Pekka Nikkarila has a PhD in Physics (2008) and serves as a researcher and a special officer at the Finnish Defence Research Agency (FDRA). He obtained his MSc in Physics (2006) and MSc(Tech.) in Electrical Engineering (2016). He has served at FDRA since 2012 with research interests in operation analysis, electronic warfare and Cyber studies. His current research interests include modelling Cyber influencing, resilience and warfare. Earlier he served as a researcher in Marioff Corporation / United Technologies Corporation (2009-2012), Inspecta (2007-2009) and at the University of Jyvaskyla (2006-2007), as well as a physics lecturer in Metropolia University of Applied Sciences (2009-2014).

Dr Dave Ormrod is a cyber security professional. Over his 22 years in the military, Dave has served on operations in Iraq and worked with multi-national forces in Europe, the United Kingdom and the United States. Dave has extensive practical leadership, management, cyber security, wargaming and simulation experience. He has a PhD in computer science from the University of New South Wales (UNSW) in 2017. Dave is certified with the Australian Information Security Registered Assessors Program (IRAP), Certified Information Systems Security Professional (CISSP) and Certified Information Security Auditor (CISA). Dave is also a Project Management Professional (PMP), a member of the SAP Defence Interest Group Security Working Group, the UNSW Australian Centre for Cyber Security Cyber War and Peace Research Group and the United States Military Cyber Professionals Association.

Timea Pahi is an engineer in IT Security. She works currently as a Junior Scientist at the Austrian Institute of Technology on several research projects focusing on threat intelligence, cyber attribution and on establishing cyber range trainings. Her area of expertise includes national cyber security, the protection of critical infrastructures, and cyber situational awareness.

Dr. Pankaj Pandey is a Research Scientist at the Norwegian University of Science and Technology, Gjovik. Dr. Pandey's research is focussed on the economics of cyber security, risk management, etc. and he has contributed to risk assessment and application of blockchain technology for IoT under the ambit of GHOST project.

Youngjun Park received the B.S. degree in Computational and Systems Biology from University of California, Los Angeles in 2018. He is currently pursuing a masters degree on Cyber Operations at the Air Force Institute of Technology. His research focuses on investigating information leakage in Internet of Things networks.

Dr. Joon S. Park is a professor at Syracuse University, Syracuse, New York, USA. Over the past decades Dr. Park has been involved with theoretical/practical research and education in Cybersecurity. He is Syracuse University's point of contact (PoC) for the Center of Academic Excellence (CAE) in Cyber Defense (Education) and CAE-R (Research) programs, which are designated by the National Security Agency (NSA) and the Department of Homeland Security (DHS).

Pierre Jacobs is an Cybersecurity Operations Manager at Cisco. He received his Masters degree n 2016 at Rhodes University with specialisation in information security, and is currently a PhD candidate at the University of Johannesburg. He specialises in Security Operation Centers (SOCs), and have implemented SOCs at both national and organisational level. His main areas of research are cybersecurity frameworks and models.

Karine Pontbriand is a PhD Candidate in Cyber Security and a member of the Research Group on Cyber War and Peace at UNSW Canberra Cyber. She is a former policy analyst at Global Affairs Canada. She holds a B.A. in International Relations and International Law and a M.A. in International and Intercultural Communication (with Distinction).

Jouni Pöyhönen, MSc. (Industrial Development and Management), Col (ret.) is PhD-student in Cybersecurity at the Faculty of Information Technology in the University of Jyväskylä. He has over 30 years' experience as developer and leader of C4ISR Systems in Finnish Air Forces. Now he is also a project researcher of the Cyber Security programs. He has in all a few research reports and articles on areas of cyber security.