2020 International Conference on Computational Science and Computational Intelligence (CSCI) | 978-1-7281-7624-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/CSCI51800.2020.00034

Controller Area Network Security Requirements

Vinayak Tanksale
Electrical and Computer Engineering
Purdue University
West Lafayette, USA
vtanksal@purdue.edu
Full/Regular Research Paper, CSCI-ISMC

***Abstract*—Controller Area Network (CAN) is the dominant communication standard for intra-vehicle communications in automobiles. The CAN protocol is designed to be light-weight in order to increase speed and efficiency. However, this severely limits the protocol's ability to support any kind of security countermeasures. Some of the devices that utilize CAN lack the resources to perform the required cryptographic computations. The rapid increase in automobile communications has exacerbated the need for efficient security solutions. This paper addresses requirements for such CAN security solutions.**

***Index Terms*—Controller Area Network, vehicular security, long short-term memory, support vector machine**

I. Introduction

The rapid and omnipresent expansion of intra-vehicle networks has increased the number of vulnerabilities to such networks. Most modern vehicles implement various physical layer and data link layer technologies. Such networks not only interface among themselves but some interface with external networks. Vehicles are becoming increasingly smart, connected and part of the Internet. This has given rise to multiple attack surfaces and vectors. Miller and Valasek [1] demonstrated successful hacking into a car in motion on an interstate by jamming the transmission system and disabling the brakes at low speeds.

Controller Area Network (CAN) is one such serial bus system that is used to connect devices. The connected devices are commonly called Electronic Control Units (ECU) although there is a subtle distinction that we outline later in this paper. An electronic control unit controls an electrical subsystem in a vehicle. Most newer vehicles contain an average of 80 ECUs. ECUs are used in transmission control, engine control, speed control, airbag control, powertrain control, and many other vehicle subsystems.

II. Controller Area Network

CAN with flexible data-rate (CAN FD) is the latest communication standard that provides higher data rates. Classical CAN was introduced in 1986 and implemented in 1988 and CAN FD was launched in 2012 and internationally standardized in 2015 in ISO 11898-1. Figure 1 shows the format of a CAN data frame. A CAN frame is a sequence of dominant and recessive bits. Any device is allowed to access the CAN bus at any time. All devices transmitting a recessive level lose arbitration to devices transmitting a dominant level. Such devices then switch to listening mode. A 29-bit arbitration ID

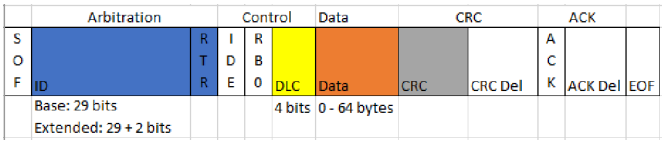

Fig. 1. CAN-FD Frame Format

is used to determine priority. If two nodes are transmitting simultaneously then the node with the lower arbitration ID maintains control of the bus. Classical CAN used an 11-bit message identifier. The 29-bit message identifier consists of the regular 11 bit base identifier and an 18 bit extension. A dominant IDE bit indicates an 11 bit message identifier and a recessive IDE bit indicates a 29 bit identifier. A distinction is made between high-speed CAN transceivers and low-speed CAN transceivers. High-speed CAN transceivers support data rates up to 1 Mbps. Low-speed CAN transceivers only support data rates up to 125 kbps. However, low-speed CAN transceivers ensure a fault-tolerant layout of the bus interface.

There are four types of frames - data, remote, error, and overload. The Start of Frame (SOF) field indicated the beginning of data and remote frames. The Arbitration field contains the message identifier and the Remote Transmission Request (RTR) bit. The RTR bit is used to distinguish between data and remote frames. Remote frames are used to solicit transmission of data from another node. SOF is used for synchronization. Bit stuffing is used to guarantee appropriate framing. ISO 11898-1 prescribes that senders must transmit a complementary bit at the latest after transmitting five homogeneous bits; a stuff bit is added even if a complementary bit followed the five homogeneous bits anyway. The Control Field is used to indicate the message identifier length and the size of the data. Data field contains data in a data frame and is empty in a remote frame. The Cyclic Redundancy Check (CRC) field contains a 15-bit frame check sequence which is computed over SOF to the Data field. The CRC delimiter bit is always recessive. The Acknowledgement field contains an ACK bit which is used to indicated a successful CRC check and a delimiter bit that is always recessive. The End of Frame (EOF) is a sequence of seven recessive bits and the intermission field (IMF) is 3-bits long. Frame transmission consists of arbitration, data transmission, and acknowledgement phases. The maximum data rate during the arbitration and acknowledgement phases

978-1-7281-7624-6/20/$31.00 ©2020 IEEE
DOI 10.1109/CSCI51800.2020.00034

is 1 Mbps whereas the data transmission phase can have a higher rate depending on end-device resources.

Error frames are transmitted when an error is detected. Cyclic Redundancy Check (CRC), Frame check, and ACK bits are used to perform error control. The CRC safeguards the information in the data frame by adding redundant check bits at the transmission end. At the receiver end, these bits are re-computed and tested against the received bits. There is no error correction mechanism other than retransmission. An error frame is transmitted when a node detects an error in a message. A node detecting an error condition sends an error flag and discards the currently transmitted frame. All nodes receiving an error flag discard the message. This results in all other devices in the network sending an error frame. All other nodes recognize the error frame sent by the node(s) that detected it and sent by themselves a second time, which results in an eventually overlapping error frame. The active Error Frame is made of six dominant bits and an 8-bit recessive delimiter followed by the IMF. The CAN standard uses bit stuffing when more than five consecutive recessive or dominant bits are present. All bit streams of more than five consecutive recessive or dominant bits signifies an error condition. The CAN error frame consists of at least six consecutive dominant bits. An overload frame is transmitted to perform flow-control. Although the overload frame is not currently used, it is designed to stem the flow of data when a device needs more processing time.

III. Attacks on CAN Communications

The following attack scenarios are possible:

- *Modification* - Malicious ECU sniffs frame and changes frame data
- *Interception* - Passively scan all traffic on CAN
- *Replay* - Lack of temporal information in the frame makes it easy to launch replay attacks
- *Fabrication* - Malicious ECU generates frame that is supposed to be generated by other ECU(s)
- *Interruption* - Denial of Service attack where malicious ECU continuously sends frames with lower IDs to thwart transmission of higher priority frames

CAN is extensively used to connect ECUs in vehicles. Miller and Valasek [1] were successful in their attacks by sending specific messages on the CAN bus. All communication in a CAN is broadcast. Hence, an analysis of the bus information can be used to determine all meta data for CAN messages. The transmission is neither confidential nor authenticated. Each CAN message has a priority which is used to resolve contention and thereby provide multiple access to ECUs. Priority is used to meet specific timing constraints for individual ECUs.

All vehicles are equipped with an On Board Diagnostic (OBD-II) port which is used run various diagnostics on the vehicle. OBD-II dongles have been used to connect cars to cellular networks. Such dongles can be exploited using SMS messages or hosted server software [2]

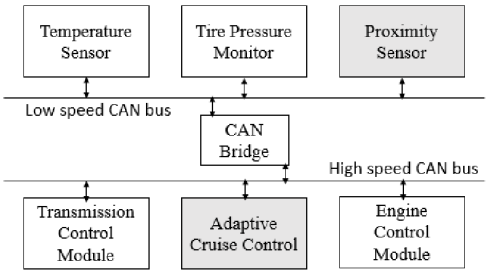

Fig. 2. High speed and low speed CAN buses

There are multiple wireless interfaces that operate over short ranges such as Bluetooth, Remote Keyless Entry, RFIDs, Tire Pressure Monitoring Systems, and WiFi. Bluetooth has become the defacto standard for supporting hands-free calling in automobiles and is standard in mainstream vehicles sold by all major automobile manufacturers. Class 2 devices used in automotive implementations have a range of 10 meters, but others have demonstrated that this range can be extended through amplifiers and directional antennas. A majority of automobiles use RF-based remote keyless entry systems to remotely open doors, activate alarms, flash lights and, in some cases, start the ignition. The adversary can place a wireless transmitter in proximity to the car's wireless receiver. For all of these channels, if there is a vulnerability in the ECU software responsible for parsing channel messages, then an adversary may compromise the ECU by transmitting a malicious input within the automobile's vicinity.

CAN transmission data can be compressed by up to 81.06 percent [3]. CAN transmission data are further reduced by up to 22 percent with the method proposed in [3], compared to enhanced data reduction algorithm. Design of safe and optimized CAN-based communication system that accounts for message offsets is detailed in [4].

Using cellular communications, repair technicians can connect remotely with a vehicle. Tesla uses an over-the-air firmware update - systems in the vehicle can receive updates, without visiting service centers.

The CAN protocol has the following weaknesses due to its design:

- Broadcast: All nodes broadcast their messages on the CAN. A malicious node on the CAN can easily sniff all traffic.
- Low-latency requirement: CAN messages are supposed to be sent and received in real-time. Any security protocol will significantly add to the delay.
- Lack of authentication: There is no support for source and message authentication. This makes the CAN network vulnerable to integrity violations and replay attacks.

IV. Background Work

In this section we review work done in the areas of confidentiality and integrity of CAN communications. Kleberger,

Olovsson, and Jonsson [5] survey the current research related to securing the connected car, with a focus on the security of the in-vehicle network.

An attack model using a malicious smartphone application in the connected car environment is outlined in [6]. A smart phone was paired with using Bluetooth to an OBD scan tool and an app on the phone was used to launch the attack. CAN frames were collected using a passive attack and then the smartphone app sent malicious frames to distort the dash board, stop the engine, and control acceleration. The authors propose a protocol using a keyed-hash and symmetric encryption but does not sufficiently address the key-distribution problem as well as does not provide any protection against active replay attacks.

Kang and Kang propose an efficient intrusion detection system based on a deep neural network for the security of in-vehicular network [7]. DNN parameters are trained with probability-based feature vectors extracted from the in-vehicular network packets by using unsupervised pre-training method of deep belief networks, followed by the conventional stochastic gradient descent method. The DNN provides the probability of each class to discriminate normal and malicious packets, and, thus the system can identify any malicious attack to the vehicle as a result.

A clock-based intrusion detection system (CIDS) was proposed in [8] to detect intrusions by fingerprinting ECUs on CAN. CIDS derived the fingerprints by extracting the ECUs' clock skews from message arrival times. While the main objective of CIDS was to detect intrusions, the authors mentioned that the thus-derived fingerprints may also be used for attacker identification, but only when attack messages are injected periodically.

Song, Kim, and Kim propose a light-weight IDS based on analysis of time intervals of CAN messages for in-vehicle networks [9]. This system can successfully detect message injection attacks in a millisecond. Design, evaluation, verification and integration of the Lightweight Authentication for Secure Automotive Networks (LASAN) is presented in [10] . LASAN is designed to achieve high performance, even in environments with low computational power and network bandwidth.

The authors of [11] used the Mean Squared Error (MSE) of voltage measurements as fingerprints of ECUs. However, they were shown to be valid only for the voltages measured during the transmission of CAN message IDs, and more importantly when voltages were measured on a low-speed (10 kbps) CAN bus; this is far from contemporary vehicles that usually operate on a 500 kbps CAN bus.

There is very little real data available on the experimental evaluation of the bit error rate for a CAN bus. [12] presents a study on a vehicle in which the amount of error caused by electro-magnetic interference is qualitatively estimated based on the number of message retransmissions and the corresponding message delays because of errors (the actual bit or frame error rate is not provided). According to the author's conclusions, CAN is shown to be a robust communication protocol in the harsh, real-world vehicle environment with significant electromagnetic activity. In the presented experiments, the bus was able to recover quickly from all errors, with no loss of data or significant delay.

Kang, Baek, *et. al.* [13] focus on how to authenticate electronic control units (ECUs) in real-time. They propose a lightweight authentication protocol with an attack-resilient tree algorithm, which is based on one-way hash chain. The protocol is deployed in CAN by performing an ECU firmware update. The protocol does not considerably add to the delay.

Sensors are authenticated and the integrity of data sent by sensors to a central server is maintained using elliptic curve digital signature algorithm (ECDSA) [14]

A security mechanism that can be used to retro-fit the CAN protocol to protect it from cyber-attacks such as masquerade and replay attacks has been proposed [15]. This mechanism has a low communication overhead and does not need to maintain global clock.

Knowledge-based intrusion detection approaches look for runtime features that match a specific pattern of misbehavior. One major advantage of this category is a low FPR. By definition, these approaches only react to known bad behavior; the basic idea is that a good node will not exhibit the attack signature. The key disadvantage of this category is that the techniques must look for a specific pattern.

Recently, a monitoring technique for detecting DOS attacks in Wireless Mesh Networks has been proposed [16]. The performance of the algorithm has been evaluated based on packet delivery ratio, average packet drops and delay metrics. It has been shown that proposed IDS successfully removes the malicious nodes and increases the packet delivery ratio while reducing the packet drop by integrating a priority mechanism into the system. However, the performance of the proposed approach is only tested for static mesh networks and its performance under mobile networks has not been analyzed yet.

Behavior-based intrusion detection approaches look for runtime features that are out of the ordinary. The basic idea is to construct models that characterize the expected/acceptable behavior of the entities. The key advantage of behavior-based approaches is they do not look for something specific. This eliminates the need to fully specify all known attack vectors and keep this attack dictionary current. An advantage of this approach is its potential for detecting unknown attacks. One major disadvantage of this category is the susceptibility to false positives.

The use of Support Vector Machines (SVM) for the detection of DoS attacks have been discussed in [17]. The performance of the proposed method has been validated experimentally and shown that proposed SVM-based detection approach achieves very high detection accuracy. However, the performance improvement of the network environment with the use of the suggested algorithm has not been analyzed.

Sekar et al. [18] combine specification-based with statistical anomaly detection techniques to ease the task of model construction and to reduce false alarm rate. Another major

disadvantage of this category is the training/profiling phase, during which the system is vulnerable.

SVM is a supervised machine learning model which is well-known for its great performance in pattern recognition and classification tasks with high dimensional data [19]. Nguyen et al. [20] describe ML techniques for Internet traffic classification. The techniques described therein do not rely on well-known port numbers but on statistical traffic characteristics.

Somwang et al. [21] propose a new intrusion detection technique by using hybrid methods of unsupervised/supervised learning. Their technique integrates the Principal Component Analysis (PCA) with SVM. The PCA is applied to reduce high dimensional data vectors and distance between vectors including its projection onto the subspace. SVM is then used to classify different groups of data, normal and anomalous.

Signature-based detection is a commonly used technique. As the number of attacks increases, the number of signatures also increases, making the usage of the whole set of signatures impractical for online detection.

Anomaly-based detection consists of creating a behavior model that is used to detect deviations from normal behavior. An anomaly-based classifier assigns a class to each event . This approach can often detect attack variations, but it tends to produce higher false-alarm rates than the signature-based approach [22].

Machine learning is often employed to implement anomaly-based intrusion detection. The network traffic is collected from the Network Interface Card or from a packet capture file containing previously captured network traffic. The packets are then filtered and sent to a feature extraction engine, which computes flow-based and header-based attributes. These attributes are assembled into a feature vector, which provides the input data for the training or classification phases of a classifier. Tavallaee M. et al. [23] proposed an anomaly detection scheme using the correlation information contained in groups of network traffic samples. The main idea is to compare the signs in the covariance matrix of a group of sequential samples with the signs in the covariance matrix of the normal data obtained during the training process. Machine learning techniques have been widely used in detecting network anomalies because machine learning can construct models automatically based on the given training data. Machine learning techniques have achieved good performance on anomaly-based detection systems. Some typical methods used in network traffic anomaly detection include Bayesian networks, support vector machine [24], fuzzy logical [25], genetic algorithm [26], and decision trees.

V. CAN Security Requirements

CAN messages are broadcast and do not contain the sender's address. All frames are received by all ECUs and each ECU determines whether to accept the frame based on the message identifier. An inherent flaw in any broadcast transmission is that malicious nodes can easily eavesdrop on all the frames transmitted by other nodes.

Data on the CAN is not encrypted. It is easy to perform traffic analysis on CAN traffic. This allows attackers to passively monitor and collect detailed metrics about CAN traffic. At present, it is not possible to verify that a message was indeed sent by an ECU claiming to send it. Any node can potentially respond to a remote frame. Such malicious frames can contain data that might disable critical control systems.

Regardless of sender integrity being maintained, there is no mechanism to verify data integrity. The message space is limited thereby complicating the data integrity verification. A majority of ECUs send very similar messages with only minor changes to the content of the message. This makes it easier to replay messages. Our proposed security requirements aim to provide *sender integrity, data integrity, and protection against replay attacks*.

There are multiple interfaces into the CAN. The OBD-II port provides direct physical access to the CAN. The OBD-II port only provides wired access to the CAN. A majority of modern automobiles are equipped with a multi-functional telematics system, which supports GPS, media entertainment, Bluetooth, cellular among others. All such interfaces are potential vulnerabilities that can be used in any of the aforementioned attack scenarios.

ECUs can be broken down into three separate sets based on available resources. *Electronic Control Modules (ECMs)* are high-powered control units. The next set of ECUs is composed of *medium-powered control units (MPCUs)*. The last set of ECUs is composed of *low-powered sensors (LPSs)*. CAN messages are sent and received within and across ECU sets. The differing computing resources require different schemes for integrity checks.

For example, Delphi MT88 Engine Control Module consists of two separate 32-bit, 80 MHz RISC microprocessors with up to 1.5 MB flash memory for independent engine management and transmission control. This enables high-speed processing and in-vehicle memory updates. Delphi MT05 Engine Control Module consists of a 16-bit, 40 MHz microprocessor with up to 256K flash memory. It is capable of high-speed data processing.

The Mass Airflow Sensor reports the amount of air entering the engine to the Powertrain Control Module (PCM). The PCM uses this input to calculate engine load. The sensor is one of the LPSs and does not have a processor. It has an interface the CAN that is uses to report data.

Cross-domain communications occur frequently. CAN messages are sent and received within and across ECU domains. ECM to ECM communications occurs on a high-speed bus and ECMs possess required resources to perform authentication. Figure 3 illustrates the three types of cross-domain communications that require consideration when designing an authentication scheme. When an LPS is sending a message to an ECM, it is not computationally feasible for the LPS to perform classical cryptographic operations. For example, a proximity sensor does not have the resources to encrypt and decrypt data that is exchanged with the adaptive cruise control module. The resource limitations on LPSs and MPCUs sensors

will increase the authentication delay. This makes classical integrity enforcement mechanisms unfeasible for communications shown in Figure 3.

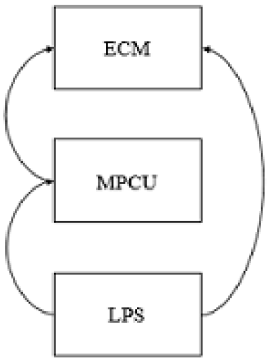

Fig. 3. Critical Cross-Domain Communication

One of the goals is to address the issues of entity authentication of ECUs and data integrity of CAN messages. We use the term delay to indicate time spent performing an operation. The delays that need to considered are transmission delay (t_t), queuing delay (t_q), propagation delay (t_p), processing delay (t_r), and authentication delay (ϵ). At present, there is no authentication of CAN messages. Any proposed authentication solution should be designed by limiting the delay that will be introduced by such a solution. Transmission delay is the amount of time it takes for an entire CAN message to be put on the bus. Transmission delay is a function of the size of the message. Propagation delay is the amount of time of time it take for the message to travel from source to destination. Propagation delay is a function of the transmission speed. Queuing delay is the amount of time that a message has to wait until all higher priority messages and successfully transmitted. Processing delay is the amount of time spent a node to process the CAN frame. Authentication delay is the sum of the amounts of time spent by the sender to generate a digest and by the receiver to complete verification. Let t be the maximum allowable response time for a CAN message. Any proposed authentication solution should satisfy the following:

$$t_t + t_q + t_p + t_t + \epsilon = t \tag{1}$$

There is a need for an authentication scheme for LPSs. Such devices possess limited computing power. Their memory and storage are limited. The absence of granular clocks is a significant challenge for any type of synchronization.

The absence of a source address in a CAN frame requires that each ECU be assigned a unique identifier. Such an identifier is not secret but can be used as one of the inputs to a hash function. As the average number of ECUs continues to increase dramatically, a 16-bit identification number will be sufficient to support the expansion of ECUs in the long-term future.

To enforce source integrity, the presence of a secret key is required. This key should be known to (or be able to be generated by) only the sender and the receiver. Key generation and distribution as well as re-keying are challenges that will need to be addressed.

CAN Message space is limited. Traffic on CAN is not encrypted. It is fairly easy to capture CAN traffic and analyze it for traffic and message patterns. All of this makes a replay attack fairly easy to execute. To thwart a replay attack, a nonce needs to be used to ensure freshness of the message. Such a nonce can be computed a function of the last sent message and current time.

VI. Conclusion and Future Work

Based on the architecture of the CAN, properties of devices that connect to the CAN, and functional and safety requirements of the vehicle, we conclude that the most important requirements for any security solution for the CAN are minimal latency, combination of offline and real-time computation, low resource requirements, temporal nonce as a function of prior messages, and hardware-based operations. We believe an intrusion detection system using support vector machines or recurrent neural networks that can be trained offline and used in real-time is one such possible solution. Our support vector machine based solution [27] has demonstrated successful results with certain limitations. To overcome these limitations, we are working on a solution that utilizes Long Short-Term Memory (LSTM). Our LSTM-based approach has demonstrated promising results in our initial experiments.

References

[1] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," in *Proceedings of the Black Hat USA 2015*, 2015.

[2] I. Foster, A. Prudhomme, K. Koscher, and S. Savage, "Fast and vulnerable: A story of telematic failures," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. Washington, D.C.: USENIX Association, 2015. [Online]. Available: https://www.usenix.org/conference/woot15/workshop-program/presentation/foster

[3] Y.-j. Wu and J.-G. Chung, "Efficient controller area network data compression for automobile applications," *Frontiers of Information Technology & Electronic Engineering*, vol. 16, no. 1, pp. 70–78, 2015. [Online]. Available: http://dx.doi.org/10.1631/FITEE.1400136

[4] P. M. Yomsi, D. Bertrand, N. Navet, and R. I. Davis, "Controller area network (can): Response time analysis with offsets," in *2012 9th IEEE International Workshop on Factory Communication Systems*, May 2012, pp. 43–52.

[5] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, June 2011, pp. 528–533.

[6] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993–1006, April 2015.

[7] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLOS ONE*, vol. 11, no. 6, pp. 1–17, 06 2016. [Online]. Available: https://doi.org/10.1371/journal.pone.0155781

[8] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *Proceedings of the 25th USENIX Conference on Security Symposium*, ser. SEC'16. Berkeley, CA, USA: USENIX Association, 2016, pp. 911–927. [Online]. Available: http://dl.acm.org/citation.cfm?id=3241094.3241165

[9] H. M. Song, H. R. Kim, and H. K. Kim, "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)*, Jan 2016, pp. 63–68.

[10] P. Mundhenk, A. Paverd, A. Mrowca, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, and S. Chakraborty, "Security in automotive networks: Lightweight authentication and authorization," *CoRR*, vol. abs/1703.03652, 2017. [Online]. Available: http://arxiv.org/abs/1703.03652

[11] P. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Processing Letters*, vol. 21, no. 4, pp. 395–399, April 2014.

[12] R. T. McLaughlin, "Emc susceptibility testing of a can car," in *SAE Technical Paper*. SAE International, 10 1993. [Online]. Available: https://doi.org/10.4271/932866

[13] K.-D. Kang, Y. Baek, S. Lee, and S. H. Son, "An attack-resilient source authentication protocol in controller area network," in *Proceedings of the Symposium on Architectures for Networking and Communications Systems*, ser. ANCS '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 109–118. [Online]. Available: https://doi.org/10.1109/ANCS.2017.25

[14] H. Mahkonen, T. Rinta-aho, T. Kauppinen, M. Sethi, J. Kjällman, P. Salmela, and T. Jokikyyny, "Secure m2m cloud testbed," in *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, ser. MobiCom '13. New York, NY, USA: ACM, 2013, pp. 135–138. [Online]. Available: http://doi.acm.org/10.1145/2500423.2505294

[15] C. W. Lin and A. Sangiovanni-Vincentelli, "Cyber-security for the controller area network (can) communication protocol," in *2012 International Conference on Cyber Security*, Dec 2012, pp. 1–7.

[16] G. Akilarasu and S. M. Shalinie, "Wormhole-free routing and dos attack defense in wireless mesh networks," *Wireless Networks*, vol. 23, no. 6, pp. 1709–1718, Aug 2017. [Online]. Available: https://doi.org/10.1007/s11276-016-1240-0

[17] S. Mukkamala and A. H. Sung, "Detecting denial of service attacks using support vector machines," in *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03.*, vol. 2, May 2003, pp. 1231–1236 vol.2.

[18] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, "Specification-based anomaly detection: A new approach for detecting network intrusions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ser. CCS '02. New York, NY, USA: ACM, 2002, pp. 265–274. [Online]. Available: http://doi.acm.org/10.1145/586110.586146

[19] S. Peng, Q. Hu, Y. Chen, and J. Dang, "Improved support vector machine algorithm for heterogeneous data," *Pattern Recognition*, vol. 48, no. 6, pp. 2072 – 2083, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320314005263

[20] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys Tutorials*, vol. 10, no. 4, pp. 56–76, Fourth 2008.

[21] P. Somwang and W. Lilakiatsakun, "Computer network security based on support vector machine approach," in *2011 11th International Conference on Control, Automation and Systems*, Oct 2011, pp. 155–160.

[22] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *2010 IEEE Symposium on Security and Privacy*, May 2010, pp. 305–316.

[23] M. Tavallaee, W. Lu, S. A. Iqbal, and A. A. Ghorbani, "A novel covariance matrix based approach for detecting network anomalies," in *6th Annual Communication Networks and Services Research Conference (cnsr 2008)*, May 2008, pp. 75–81.

[24] A. H. Sung and S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks," in *2003 Symposium on Applications and the Internet, 2003. Proceedings.*, Jan 2003, pp. 209–216.

[25] X. D. Hoang, J. Hu, and P. Bertok, "A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference," *Journal of Network and Computer Applications*, vol. 32, no. 6, pp. 1219 – 1228, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S108480450900071X

[26] W. Li, "Using genetic algorithm for network intrusion detection," in *In Proceedings of the United States Department of Energy Cyber Security Group 2004 Training Conference*, 2004, pp. 24–27.

[27] V. Tanksale, "Intrusion detection for controller area network using support vector machines," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, 2019, pp. 121–126.