CAN Obfuscation by Randomization (CANORa)

A technology to prevent large-scale malware attacks on driverless autonomous vehicles

Tobias Madl
MuSe – Munich
IT Security Research Group
Munich University of
Applied Sciences
Munich, Bavaria, Germany
tobias.madl@hm.edu

Jasmin Brückmann
MuSe – Munich
IT Security Research Group
Munich University
of Applied Sciences
Munich, Bavaria, Germany
brueckma@hm.edu

Hans-Joachim Hof
INSicherheit - Ingolstadt Research
Group Applied IT Security Technical
University of Ingolstadt
Ingolstadt, Bavaria, Germany
hof@thi.de

ABSTRACT

Driverless autonomous vehicles pose new challenges for security due to an increased attack surface and the missing "human in the loop". Future driverless autonomous vehicles could, for example, become targets for large-scale malware attacks. Such malware may spreads over V2X communication, infecting a large number of vehicles. Infected driverless autonomous vehicles may be manipulated in a way such that they drive to a place where it is easy for car thieves to collect the vehicles. This paper presents CANORa (CAN Obfuscation by Randomization), a security approach to mitigate large-scale malware attacks on driverless autonomous vehicles by increasing the effort for an attacker to send legitimate messages on the CAN bus. The CAN bus is of high importance for the security of a vehicle as it links many safety critical ECUs with each other. With full control of the CAN bus, an attacker can drive a vehicle. Hence, a successful attack on the CAN bus must be considered to be the worst case for automotive security. CANORa has a very small memory and computational footprint. Hence, it can be efficiently implemented even on today's ECUs. A prototype implementation of CANORa demonstrates the practical feasibility of this approach.

CCS CONCEPTS

Computer systems organization → Embedded and cyber-physical systems → Embedded systems; Security and privacy→ Network security; Obfuscation;

KEYWORDS

CAN security; obfuscation;

CSCS 2018, September 2018, Munich, Germany

ACM ISBN 978-1-4503-6616-8/18/09...$15.00
https://doi.org/10.1145/3273946.3273953

1 INTRODUCTION

The CAN bus is a very common fieldbus in vehicles. It is of high importance for the security of a vehicle as it interconnects many safety critical ECUs, e.g., the ECU for braking. With full control of the CAN bus, an attacker can drive a vehicle. Hence, a successful attack on the CAN bus must be considered to be the worst case for automotive security. In May 2018, hackers of Keen Security Lab demonstrated this worst case. It was possible to gain local and remote access to the infotainment system of selected BMW vehicles and that this attack enables attackers to gain control of the CAN bus [1]. In most cases, the CAN bus of a vehicle is not accessible via external communication interfaces. An attacker must infiltrate several other systems of a vehicle to finally reach a hackable ECU that is connected to the CAN bus. However, the increasing connectivity of vehicles in combination with the tremendous additional amount of software in modern vehicles, e.g., for autonomous driving, leads to a situation where vehicles have many exposed interfaces (e.g., see [11]), hence offer a large attack surface. Current attacks like the BMW hack still need a lot of manual hacking and usually target only one distinct vehicle. However, with the advent of driverless autonomous vehicles, malware for vehicles may become possible and could be attractive, e.g., for car thieves. Just imagine a piece of malware that infiltrates a whole fleet of driverless autonomous vehicles. The malware gains control over the car and drives these vehicles to a location where car thieves can easily pick them up. Rising the bar to send legitimate messages on the CAN bus may mitigate such an attack scenario. CANORa applies obfuscation on CAN bus communication in vehicles. Obfuscation is a technique to conceal the meaning of data or communication while still keeping functionality.

The basic idea of CANORa is to obfuscate some CAN message fields using randomization and permutation of data and identifier fields. To send a legitimate message on the CAN bus, an attacker needs to know the obfuscation used for the attacked vehicle. At present, CAN communication is defined by a so-called CAN profile. To send legitimate messages on the CAN bus, an attacker must know the CAN profile of a vehicle. Usually, all the vehicles of a vehicle series share the same CAN profile.

Hence, an attacker can learn the CAN profile by reverse engineering CAN communication of one vehicle of a series and apply the reverse engineered CAN profile on all other vehicles of a vehicle series. In CANORa, obfuscation of data fields and identifier fields is used to create CAN profiles that are unique for each single vehicle. Hence, each vehicle of a vehicle series has a different CAN profile. Without the knowledge of the CAN profile, an attacker cannot send valid CAN messages, hence cannot use the CAN bus in the attack. By obfuscating CAN messages, a once far-reaching and damaging attack against a whole fleet of driverless autonomous vehicles can no longer be easily automated (e.g., by malware). Thus, large-scale malware attacks become more difficult for an attacker. The reason is that an attacker now has to analyze each single vehicle, in order to identify existing message formats and map their corresponding functions.

The rest of this paper is structured as follows: Section 2 reviews related work on CAN obfuscation. Section three presents the design of CANORa. Section four briefly describes a proof of concept implementation. Section 5 discusses the security of CANORa. Section 6 concludes the paper and gives an outlook on future work.

2 RELATED WORK ON CAN OBFUSCATION

There are several approaches to mitigate from large-scale malware attacks, e.g., [2-3] propose intrusion detection systems (IDS) to detect such attacks. However, an intrusion detection system does not protect against attacks but only reports ongoing attacks. Usually, intrusion detection systems need a human that evaluates attacks and reacts on them. Drivers are not capable of doing this. Hence the OEM or external partners must analyze possible attacks. IDS may benefit from CANORa as CANORa forces attacking malware to analyze CAN bus communication, hence slows down attacks, and hence provides OEMs with more time to analyze attacks.

The CAN bus can be protected using Message Authentication Codes (MACs) [4] or CAN encryption. In both cases, it is necessary to implement cryptographic functions and to manage keys. Also, additional data needs to be sent on the CAN bus, e.g., the MAC. Hence, Message Authentication Codes and CAN encryption have a moderate communication, computational, and memory overhead. The OEMs tend to be very cost sensitive, hence adding overhead is problematic in the automotive domain. CANORa has a very small communication, computational, and memory overhead for obfuscation. CAN profiles need to be integrated into ECUs anyway. MACs and CAN encryption are effective in message protection and can be combined with CANORa.

Another way of protecting the CAN bus against attacks is ID Hopping [5]. In this approach, a gateway is used to detect attacks. If an attack is detected, adding a well-defined offset to identifiers alters CAN messages. Then, these new identities are sent to ECUs by a secure channel. This approach could prevent denial of service attacks on ECUs as well as targeted attacks on specific ECUs. However, the approach has two weaknesses. First, it requires a mechanism to automatically detect ongoing attacks, which is hard to realize. And second, it needs a secure channel between ECUs. Establishing a secure channel usually has a moderate communication, computational, and memory overhead. In contrast, CANORa does not use secure channels.

The idea of using obfuscation in vehicle communication was first introduced in [6]. The paper presents an approach to calculate of random CAN identifier bounds using a Quadratically Constrained Quadratic Program. Data bytes of CAN messages get encrypted. Both, the complex calculation of the CAN identifiers and encryption have a high computational overhead. Also, the approach relies on backend services of the OEM, and the approach is limited on obfuscation of CAN identifiers. CANORa extends the idea of [6]. In contrast to [6], CANORa uses a simple randomization approach and a larger number of message parts get obfuscated. CANORa offers a mode of operation that does not need any OEM backend services.

3 DESIGN OF CANORA

One of the main goals of CANORa is to mitigate large-scale malware attacks against driverless autonomous vehicle. The design of CANORa assumes an active but static remote attacker. In attacker modeling, an active attacker is an attacker that actively attacks systems, e.g., the attacker exploits vulnerabilities in software. This is the default behavior of malware. A static attacker is an attacker that has a predefined behavior. This means that malware is not capable of doing in-depth analysis of the target system. A remote attacker is an attacker that has no local access to a vehicle. This is the case for a malware attack as it is assumed that malware spreads over remote communication. See [9] for more details on attacker modeling for autonomous driving.

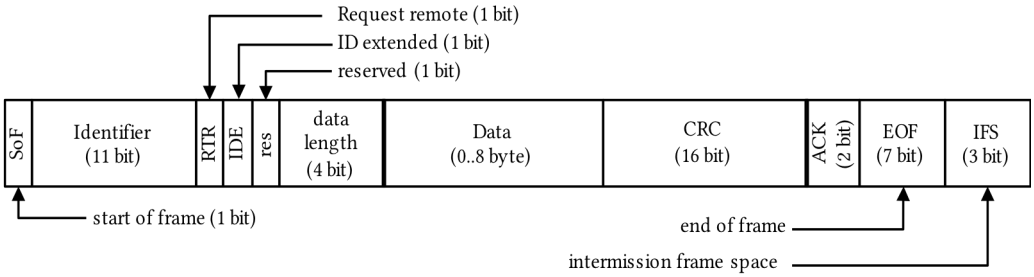
CANORa obfuscates CAN messages of a given CAN profile to produce a CAN profile that is individual for a single vehicle. On the CAN bus, one of four different frames is used for communication:

The *data frame* is the standard frame for data transmission and the most common frame in CAN communication. Data frames exist in two versions: base frame format and extended frame format. See Figure 1 for information on the different frame formats.

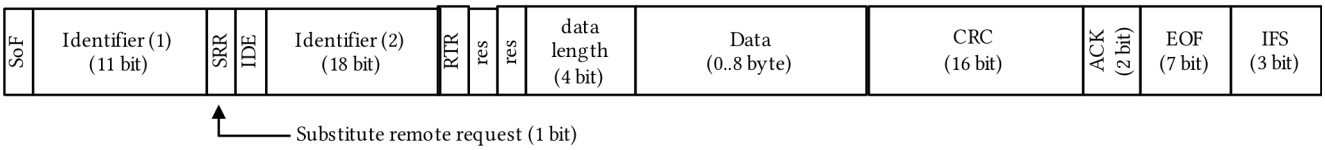
A *remote frame* is used to request data from other ECUs. Most ECUs send their data in autonomous or periodic data frames, but some ECUs may use remote frames to request a specific set of data based on the used identifier.

The *error frame* is used to signal errors on the CAN bus.

Overload frames are used to delay further data or remote frames to handle overload situations of a single ECU.



(a) CAN data frame in base frame format.



(b) CAN data frame in extended frame format.

Figure 1. CAN data frame.

CANORa focuses on obfuscation of the data frame, which is the most frequent frame of CAN. It should be noted that adversaries still could use the other three frames for attacks. These frames cannot be used for remote control of a car, but they may be useful in a Denial of Service attack on the CAN bus.

The CAN data frames consist of different parts, including identifier field, data field, CRC field, etc. See Figure 1 for details. Some fields are mandatory and cannot be changed, e.g., the start of frame (SoF), ACK slots, and all delimiter fields. Obviously, these fields are not available for obfuscation. Candidate fields for obfuscation include the identifier field(s) as well as the data field.

Obfuscation of data field

The data field can be obfuscated without any limitation. The obfuscation of the data field is a permutation of data bits or data bytes based on a unique random seed. The random seed is described in detail below. If the data field gets obfuscated, it is also necessary to adapt the CRC field.

Obfuscation of the data field is not very effective if most CAN messages of a CAN profile only use very small data fields, as the number of available permutations is limited in this case. For example, if most CAN messages only use one byte data fields, only 8! = 40320 permutations of this data field exist. With knowledge of the semantics of the data, an attacker may easily reverse engineer the obfuscation of the data field. To address this issue, dummy data may be used to extend the data field of CAN messages before obfuscation. Adding 1 single random dummy byte in the example above results in 16! = 20,922,789,888,000 available permutations – enough permutations that an attacker cannot reverse engineer the obfuscation, even if he knows the semantics of the data field. The receiver simply ignores dummy data, but the attacker does not know which part of the obfuscated data field is dummy data and which is not. Using dummy data, it is also possible to adapt the length of all data frames, making it more difficult for an attacker to use the length of a frame to analyze the type of message. It should be noted, that inserting dummy data increases the traffic on the CAN bus. Hence, this has an impact on performance.

Obfuscation of identifier field

CAN uses CSMA/CR for CAN bus access. On the CAN bus, a logical “0” is dominant to a logical “1”, meaning that if sender A sends a “0” at the same moment sender B sends a “1”, the “0” is transmitted. All senders check during sending if the bit they receive is different to the bit they sent. If this is the case, they stop sending. As all CAN frames start with the start of frame delimiter followed by the identifier, messages with a low identifier have precedence over messages with a high identifier. This is called arbitration and it is used to make sure that messages with a high priority (low identifier) cannot be blocked by messages with low priority. Arbitration is very important for safety of the vehicle, hence CANORa may change the identifiers, but it is crucial that the ordering of identifiers stays the same to keep the priority ordering of CAN messages.

To obfuscate the identifier field of a CAN frame but still keep the original ordering of priorities, the identifiers of all the messages in the CAN profile are gather in an ordered list. The random number generator is seeded with a seed, see below for details. A second ordered list with the length of the list of all identifiers is created using randomly chosen identifiers from the available identifier space ($0..2^{11}$-1 for base frame format respectively $0..2^{29}$-1 for extended frame format). In the last step, each

identifier of the first list is mapped on the identifier of the second list that has the same index. This approach keeps the ordering of the priorities of CAN frames. Also, the approach uses the whole address space of the base frame format and the extended frame format for obfuscation. Also, no time-consuming calculations as in [6] have to be executed in order to generate a new CAN profile. Another advantage is that the OEM does not have to store any information about individual CAN profiles.

If most of the available identifiers are already used in the original CAN profile, an attacker may determine the randomized identifiers as identifiers can only be selected in a small range. To address this problem, a threshold is defined. If less than threshold percent of identifiers are available, the frame format is changed from the base frame format to the extended frame format, which offers 2^{29} instead of 2^{11} possible IDs. It should be noted that this may result in increased data traffic on the CAN bus, hence may influence the performance of the CAN bus. For the safety of the vehicle, it is important to thoroughly test the impact of obfuscation during development of a vehicle.

A weakness of CANORa is that it complicates CAN filtering. ECUs use CAN filtering to only process messages with IDs that are of interest for them. With the original CAN profile, a small number of simple bitmasks is sufficient to identify the messages the ECU is interested in. After obfuscation, the number of bitmasks may be much higher. Future versions of CANORa will address this issue, it is out of scope of this paper.

Security Profiles

CANORa offers two security profiles for obfuscation, one aiming for a low overhead and easy configuration (called standard security profile) and one aiming for a high security level (called high security profile).

The high security profile provides a high level of security but has higher complexity and higher costs compared with the standard security profile. The standard security profile provides a better balance between costs and benefits – it does not need additional hardware, but the security level is lower than the security level of the high security profile.

Both profiles use a seed value to create individual CAN profiles. The seed value is unique to a vehicle. It is used to seed the random number generator used in the obfuscation of data fields and identifiers. This paper does not elaborate details on suitable random number generators for CANORa. Several possible implementations will be evaluated in future work.

Neither seed value nor the complete individual CAN profile of a vehicle should be stored under any circumstances in the ECU software. Every ECU should just know the part of the individual CAN profile that is relevant for it. This keeps an attacker from learning the individual CAN profile of a vehicle by hacking one single ECU. If an attacker manages to compromise an ECU, he is able to identify receiving messages, but could not reverse engineer the whole traffic.

Standard security profile of CANORa

The main goal of the standard security profile is the deployment of individual CAN profiles without the need of additional hardware in the vehicle and in with as little effort as possible for the OEM. Figure 2 shows the basic setup of the standard security profile, which is a part of the reference model from [10], see there for more details. It has three main parts: the OEM, the vehicle itself, and possible external partner. The OEM produces the vehicles and its internal components. Every new vehicle gets produced with an individual CAN profile.

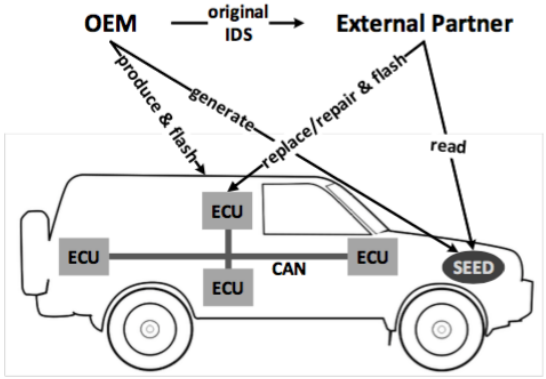


Figure 2. Setup of standard security profile

The only change necessary to the production process takes place during flashing of software on CAN connected components. At this process step, a randomization seed for CANORa is generated. Using this seed, the original CAN profile is randomized and the changed profile is flashed into the connected ECUs. The seed may be used to synchronize the random generator between different machines and platforms, but it is never actually stored on any ECU of the vehicle. As the OEM has to use a CAN profile anyway, picking a random seed and randomizing the CAN profile produces only negligible overhead.

The seed must be kept secret to keep CANORa secure. However, the seed may be needed during update, repair, or replacement of an ECU (either by the OEM or by external partners). In these cases, the individual CAN profile of the vehicle must be recreated to successfully flash the ECU. Repair shops may need access to the seed for diagnostic reason. This could be achieved by printing the seed on a physical label. The label has to be place at a "secret" place in the vehicle, where it is not readable from the outside. In the best case, even from the inside of the car it should not be easy to access this information, for example, the label could be placed behind some covering that has to be taken away with some screws. Thereby, two conditions have to be fulfilled to get access to the seed and resolve the randomization. First, the attacker needs physical access to the internals of the vehicle and cannot attack from remote, also he has to have full access to the vehicle in order to take it apart and read the seed. By this, even malicious passengers should not have the possibility to quickly read said seed. This is only some extra security, the main point is the only physical storage of the randomization seed in the vehicle. This hinders the attacker assumed for the design of CANORa (active, static, remote attacker) to disclose the seed.

External partners (like repair shops) also need the individual CAN profile when updating, repairing, or replacing an ECU. They can also access the "hidden" seed and input it in a flashing tool. The flashing tool uses the seed to initialize the random number generator. Next, this flashing device needs to know the list of all available CAN messages in this vehicle with their original identities. This information is either already stored inside the device or requested from the OEM via some interface. With these information, seed and original CAN profile, the flashing tool can regenerate the individual CAN profile for the vehicle. The flashing device has to be generic like the OBD diagnostic tool, or it should be provided by the OEMs. On the one hand, this is more work for the external partners. On the other hand, it is close to connecting a standard diagnostic system to the OBD II plug as it is common today in repair shops.

To sum it up, the main obfuscation by randomization step of the standard security profile takes place at the OEM. The OEM generates the individual CAN profiles and flashes it onto the ECUs. The seed can be accessed and is needed for replacing components.

The advantage of the standard security profile is that there is no need for additional hardware and only changes of production processes of the OEM are necessary. Obfuscated CAN communication undergoes the thorough testing during the production process. Hence, any safety issues should be detected. The seed, which is the most important secret in CANORa, is kept in a place where it is not accessible for a remote attacker. Legitimate users like the owner or a repair shop have access to the seed. Also, the standard security profile is easy to implement.

The disadvantages of the standard security profile is that the individual CAN profile is static, hence can be learned by an attacker that invests some time and listens on the CAN bus for ongoing communication. Considering the large-scale malware attacks described above, these attacks are at least slowed down as analysis needs time. Other security components may benefit from this slowdown, e.g., an OEM may have more time to analyze IDS alarms and react on them. Another disadvantage of the standard security profile is that if the seed is revealed, all ECUs must be reflashed and a new seed must be chosen. All those disadvantages are overcome by the high security profile of CANORa, however, at the price of increased costs.

High security profile of CANORa

The high security profile of CANORa is more complex than the standard security profile. It employs additional randomization hardware for the obfuscation. Figure 3 shows the setup of the high secure concept. The main parts are the OEM, the vehicle, and external partners like repair shops. The main difference to the standard security profile is the use of an additional piece of hardware for the obfuscation. This allows for a continuous re-obfuscation of the CAN profile.

All the obfuscation takes place in the vehicle. The use of the randomization hardware allows minimizing the needed interactions of the OEM or external partner. Also, the randomization hardware allows overcoming the disadvantages of the standard security profile of CANORa: the static individual CAN profile. Attackers may eavesdrop on communication on the CAN bus and try to reverse engineer the obfuscated communication. In contrast, the high security profile of CANORa allows to change the obfuscation from time to time, hence lowers the window of opportunity for an attack. This becomes possible, because the randomization hardware does all the obfuscation, so no interaction with the OEM is needed. The high security profile allows creating dynamic, individual CAN profiles for cars. For example, the randomization hardware may generate a new seed every time the vehicle is parked for a certain amount of time. It transmits to every ECU the corresponding new identifiers and data field obfuscation. To do so, the randomization hardware needs to know the original CAN profile of the vehicle and needs to be recognizable for the ECUs. Of course, the randomization hardware must take into consideration the state of the vehicle. Especially, it should not be active while the vehicle is moving.

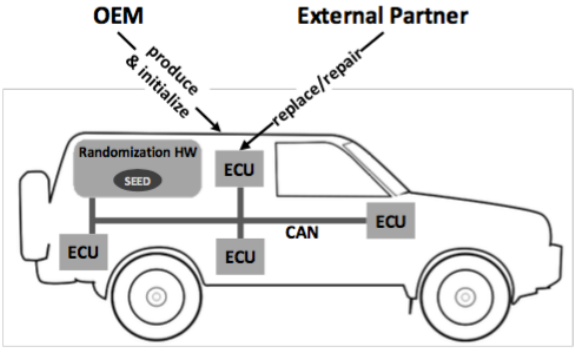


Figure 3. Setup of high security profile

The randomization hardware generates the individual CAN profiles and installs them on the ECUs. Every connected ECU listens to an identifier where it gets its new randomized values, the exact protocol for this has yet to be defined. Only known ECUs will be able to connect to this CAN bus. No other external or new component will be integrated in this randomization process. Neither the seed value nor the randomized values are accessible from outside or by unauthorized parties. This approach eases the work for external partners: there are no extra steps for CANORa involved during the replacement of an ECU. When the new ECU gets connected to the CAN bus, it sends a registration message, including an authentication token. The randomization hardware checks the authentication token and replies with the individual CAN profile for this ECU. The seed does never leave the randomization hardware. For legitimate diagnostics of CAN bus communication, e.g., in repair shops, a special device is needed. This device must authenticate itself to the randomization hardware to get integrated in the CAN communication. If an ECU should be integrated during a repair, the randomization hardware configures it automatically after an authentication. If the ECU is authorized, it gets the current individual CAN profile from the hardware module. Completely new ECUs are ignored to prevent attacks with unauthorized devices.

With the high security profile of CANORa, an attacker only has a very short window of opportunity to learn the current individual CAN profile of a vehicle by eavesdropping before the

individual CAN profile gets changed. However, all information on generation of the CAN profiles is stored in the randomization hardware, hence it is one of the most valuable components in the vehicle and should be protected accordingly, both from unauthorized software and physical access.

The advantage of the high security profile of CANORa is the dynamic nature of individual CAN profiles that makes it even harder for an attacker to reverse engineer obfuscated CAN communication. With this feature, a persistent malware capable of CAN bus analysis and even a local attacker will have only a very short window of opportunity to map identifiers. By this, not only attacks against whole vehicle fleets are prevented, but also attacks against a single vehicle get more difficult. Also, with the high security profile of CANORa, the process of flashing and configuring an ECU is simplified for OEM and external partners compared to the standard security profile of CANORa. Another advantage is the self-organizing system. It could be extended later to implement even more security mechanisms, because this component will be designed to be the most secured one in the vehicle.

Disadvantages include possible high costs for the randomization hardware. Also, the periodic transmission of CAN profiles increases the traffic on the CAN bus, hence may impact the performance of the CAN bus. Another disadvantage is the inflexibility of adding completely new ECUs. As the randomization system only accepts well-known components that were already in place during production of the vehicle additional ECUs would mean an exchange of the randomization hardware. However, such a retrofitting of vehicles is not a common use case, future work will focus on this aspect. Also, an authentication function is needed for this approach to work. Additional effort needs to be invested to get this approach running with current hardware. It should be noted that the dynamic nature of the individual CAN profile may have safety impacts that should be evaluated in future work.

4 PROOF-OF-CONCEPT IMPLEMENTATION

To show the feasibility of CANORa, a proof-of-concept implementation is used. The main goals for the implementation are:

- Minimal performance overhead on ECUs
- Real-time requirements must be met (safety requirement)
- Prioritizing of CAN messages must be preserved (safety requirements)
- It should be possible for authorized personnel to replace broken ECUs.

The target platform for the prototype implementation is a Beagle Bone Black. A basic CAN bus with only two CAN nodes was implemented by connecting two Beagle Bones via CAN. Both Beagle Bones run an embedded Debian Linux. The proof-of-concept is written in Python, using the library “python-can” [7]. It implements the case study described in [8], which represents a basic, randomly generated CAN setup. The case study consists of five CAN buses that define clusters and a central gateway that interconnects these clusters. Note that the functions in each cluster are not solely mapped to ECUs of this cluster such that it is not possible to determine the priorities for each cluster separately.

cluster (CAN bus)	ECUs	tasks	functions / periods
body	6	71	40,40,50,80,80,80
driver assistance systems	3	74	5,10,20,40,80,80
chassis	4	97	10,20,40,40,50,50,100
infotainment	4	38	5,80,100
powertrain	8	52	5,10,40,50,50,80
system	25	332	

Figure 4: Clusters (buses), ECUs, number of tasks and periods of the functions of the case study [8]

The details of the case study are summarized in Figure 4, which also illustrates the periods of the functions in each cluster. Overall, the case study consists of 25 ECUs and 27 functions that result in 332 tasks. In summary, 149 tasks are implemented on the ECUs that require a feasible priority assignment while 137 priorities have to be assigned to the existing message tasks that are routed via CAN bus. The implementation on two Beagle Bones was run for 10 minutes. Then, the CAN profile was automatically randomized and the tests were rerun for the same amount of time. Afterwards, the results were compared to the accomplishments of above stated goals verified. The implementation is still an ongoing activity.

5 SECURITY EVALUATION

This subsection discusses the security of CANORa based on an attacker model for a large-scale malware attack on driverless autonomous vehicles. The attacker model follows the description in [9], see this paper for more examples on attacks on autonomous driving. Large-scale malware attacks are modeled by assuming an active but static remote attacker. In attacker modeling, an active attacker is an attacker that actively attacks systems, e.g., the attacker exploits vulnerabilities in software. This is the default behavior of malware. A static attacker is an attacker that has a predefined behavior. This means that malware is not capable of doing in-depth analysis of the target system. A remote attacker is an attacker that has no local access to a vehicle. This is the case for a malware attack as it is assumed that malware spreads over remote communication. The attacker model assumes that the goal of the active, static remote attacker is to get full control of the CAN bus.

As the attacker is a remote attacker, he usually cannot get direct access to the CAN bus but must first infiltrate a device that has an external communication interface. The attacker likely needs to infiltrate more than one device before the attacker gains control of an ECU connected to the CAN bus. With this ECU, the attacker only has very limited access as the ECU only knows the correct obfuscation of CAN message it uses itself. It is very likely that this is only a small subset of all CAN messages. If the attacker ignores CANORa and just sends CAN messages, these messages may be detected by a safety mechanism for detection of faulty ECUs that excludes these devices from further commu-

nication. Hence, in short time, the hacked ECU could become isolated from the CAN bus.

Considering the computational and memory resources of typical ECUs, it is challenging for an attacker to implement a CAN analysis to revert the obfuscation.

An attacker could try to revert the obfuscation of identifiers by eavesdropping on CAN communication, listing all identifiers, sorting identifiers, and mapping the identifiers on the original CAN profile or the individual CAN profile of another vehicle. This analysis takes some time, hence changing the obfuscation from time to time as suggested for the high security profile of CANORa helps to hinder this attack.

An attacker could try to revert the obfuscation of data fields by using knowledge about the sensor values - often, sensor values have only a very gradual change, so only the least significant bits change with every CAN message. An attacker can identify these bits in the obfuscated data fields. To prevent this attack, data fields may be padded with dummy data before obfuscation. See section 3 for details. The high security profile of CANORa changes the obfuscation of identifiers and data fields on a regular basis, hence hinders this attack.

Another way to get full control of the CAN bus would be to learn the seed used for CANORa. However, the seed is well protected – in the standard profile, the seed is not available from remote as it is printed on a physical label inside of the vehicle. In the high security profile, the seed is protected by the randomization hardware.

Hence, CANORa at least mitigates attacks on the CAN bus in both, the standard security profile as well as in the high security profile.

6 CONCLUSION AND OUTLOOK

This paper presents an approach to mitigate large-scale malware attacks on driverless autonomous vehicles: CAN obfuscation by randomization (CANORa). CANORa obfuscates CAN communication by randomization of identifiers and permutation of data fields of CAN data frames. By doing so, CANORa provides a CAN profile that is individual for each vehicle. An attacker must learn the individual CAN Profile of a vehicle before it could send any valid CAN frames on the CAN bus. CANORa provides a standard security profile that has a very low memory and computational overhead, hence can be implemented even on today's ECUs and does not result in a significant increase in costs. The high security profile of CANORa requires additional hardware per vehicle. The high security profile offers dynamic CAN profiles that are individual for each single vehicle and are changed from time to time. The high security profile of CANORa renders attacks useless that try to reverse engineer obfuscated CAN communication by eavesdropping on the CAN Bus. A prototype implementation so far shows the general feasibility of the approach. Future work will include a demo implementation on a target platform of a major OEM. It is also planned to transfer the idea of obfuscation on other vehicular communication technologies, e.g., automotive Ethernet or FlexRey.

REFERENCES

[1] Keen security lab. Experimental Security Assessment of BMW Cars: A Summary Report, https://keenlab.tencent.com/en/Experimental_ Security_Assessment_of_BMW_Cars_by_KeenLab.pdf [last access 23.05.2018]
[2] C. Miller and C. Valasek. Adventures in automotive networks and control units. DEF CON, vol. 21., pp. 260-264, 2013
[3] C. Miller, C. Valasek. A survey of remote automotive attack surfaces, Black Hat USA, vol. 2014, 2014
[4] R.Zalmanand and A.Mayer. A secure but still safe and low cost automotive communication technique," in Proceedings of the 51st Annual Design Automation Conference. ACM, pp. 1–5., 2014
[5] A. Humayed and B. Luo. Using id-hopping to defend against targeted dos on can, in Proceedings of the 1st International Workshop on Safe Control of Connected and Autonomous Vehicles. ACM, pp. 19– 26, 2017
[6] M. Lukasiewycz, P. Mundhenk, and S. Steinhorst. Security-aware obfuscated priority assignment for automotive can platforms, ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 21, no. 2, p. 32, 2016.
[7] B. Thorne. [Online]. Available: https://github.com/hardbyte/python-can
[8] M. Lukasiewycz, S. Steinhorst, and S. Chakraborty. Priority assignment for event-triggered systems using mathematical programming, in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013. IEEE, pp. 982–987, 2013
[9] C. Ponikwar and H.-J. Hof. Beyond the Dolev-Yao Model: Realistic Application-Specific Attacker Models for Applications Using Vehicular Communication. The Tenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2016), Nice, France, 2016.
[10] J. Brückmann, T. Madl, H.-J. Hof. An Analysis of Automotive Security Based on a Reference Model for Automotive Cyber Systems. The Eleventh International Conference on Emerging Security Information, Systems and Technolgies (SECURWARE 2017), Special Track "Secure Automotive Cyber Systems", Rome, Italy, 2017
[11] C. Ponikwar, H.-J. Hof, L. Wischhof. Towards a High-Level Security Model for Decision Making in Autonomous Driving. ACM Computer Science in Cars Symposium 2017 (CSCS 2017), Munich, Germany, 2017