VEHICULAR CAN TRAFFIC BASED MICROTRACKING FOR ACCIDENT RECONSTRUCTION

András Gazdag[1], Tamás Holczer[2], Levente Buttyán[3] and Zsolt Szalay[4]

[1,2,3,4] Budapest University of Technology and Economics, Budapest, Hungary,
[1,2,3] Department of Networked Systems and Services,
[1,2,3] Laboratory of Cryptography and System Security
[3] Ukatemi Technologies LLC
[4] Inventure LLC
[1]agazdag@crysys.hu, [2]holczer@crysys.hu, [3]buttyan@crysys.hu,
[4]zsolt.szalay@gjt.bme.hu

Abstract. Accident reconstruction is the process of reliably discovering what has happened before a serious event. We show how the most widely used intra vehicular network (namely the Controller Area Network, CAN) can be used in this process. We show how the actual velocity and steering wheel position transmitted on the CAN network can be used to reconstruct the trajectory of a vehicle. This trajectory is an essential input in the reconstruction process. In this paper, we show how the CAN traffic of an actual vehicle can be used to reconstruct the trajectory of the vehicle, and we evaluate our approach in several real life experiments including normal and pre-accident situations.

Keywords: Digital forensic, CAN network.

1 Introduction

Accident reconstruction is a standard process after any serious casualty. This process tries to clarify what has happened just before an accident. The base of the reconstruction are the tire marks on the road, the deformations on the vehicles, and the reports of the witnesses. In this paper, we propose another useful source, which is the logged traffic from the vehicle's CAN network.

CAN (Control Area Network) is the most widely used intra vehicular network, which connects different ECUs (Electronic Control Unit) and has broadcast traffic that describes the vehicle's state. For example, the vehicle's speed, the angle of the steering wheel, or the position of the brake pedal is also transmitted through the CAN. We argue that this information can be used to reconstruct the trajectory of a vehicle before an accident which leads to a more reliable reconstruction of the event.

Our approach is similar to the EDR (Event Data Recorder) required in the United States of America, but it can be retrofitted into already sold vehicles. The only requirement is an access to the CAN network, which can be realized by a CAN logger attached to the OBD port (which can be found in any vehicle sold after 2003 in Europe). The European equivalent of EDR is called eCALL. The main difference between EDR and eCALL is that eCALL tries to alert the emergency agencies in case of an accident, but does not store any data about the accident itself. Our approach is a good complementary for the eCALL system whose adoption is ongoing in several countries of Europe. The collected data can also be used to prove driving habits, which can be an essential data source for personalized insurances.

In this paper, our contribution is the following. (i) We show how a “black box” can be retrofitted into existing vehicles based on a CAN logger. (ii) We show how this device and the stored data can be used in accident reconstruction. (iii) We introduce several experiments to correct and to measure the accuracy of our approach in typical accident situations. The remainder of this paper is organized as follows: in section two, we show some related work, in section three we introduce the basic components of micro tracking, which are used and analyzed in section four. We conclude our paper with some acknowledgement in section five and six.

2 Related work

Vehicular accident reconstruction is important part of forensic investigations, and such has a wide literature. Some handbooks exist for police officers, attorneys, or insurance professionals, who do this job daily. Such books are mainly dealing with the simulated and real world physics of accidents, and only partly mentions other data sources, such as Event Data Recorder (EDR) [1,2,3].

Data recording devices that can capture information continuously or triggered by an event have existed in the transportation industry for decades. The best known such devices are probably the “black boxes” used in aviation to record data that can be used by investigators to reconstruct some of the circumstances of an airplane crash. Such recording devices now also exist in road vehicles: since September 2014, a so called Event Data Recorder (EDR) is mandatory for every new passenger car and new light commercial vehicle (LCV) in the US.

The purpose of EDR devices is to collect data about the vehicle dynamics and the vehicle status that enable better accident reconstruction. It helps in validating insurance claims, encourages safer driving behavior, and extends the scientific knowledge about real accidents. The importance of an EDR-like “black box” increases with the deployment of highly automated functions in road vehicles, as there must be some objective evidence proving who was in charge of control in the vehicle in a critical situation. It is, however, not clear what would be the minimum set of data that needs to be collected in case of automated or highly automated vehicles; accident researchers and automated vehicle experts are currently working together on new regulations in this field.

While EDR devices collect data from the CAN bus [4], the recording of that data is not continuous in time, but triggered only by certain events that may indicate a forthcoming accident (e.g., events that trigger the airbag). In addition, the data recorded by EDR devices is limited to a short interval in time (typically a few seconds) surrounding the point in time of the accident. Some work on comparing the data on the CAN bus and the data stored by the EDR can be found in [5].

There exist data recording devices, such as tachographs, that perform continuous data collection in vehicles. Tachographs are mainly used on heavy trucks, buses, and emergency vehicles to continuously record certain parameters of the vehicle such as its speed, its engine RPM, and odometer values. Yet, the main purpose of tachographs is to monitor the duty status of the drivers of commercial vehicles, and they are not designed to record raw CAN traffic. They usually record only a few vehicle parameters with a certain recording frequency, and they are not available on all kinds of road vehicles. Hence, similar to EDRs, tachographs in their current form cannot really be used in investigations of cyber incidents affecting vehicles.

Hence, we can conclude that, although they have seemingly similar goals, existing data recording devices in road vehicles actually address a problem different from the one that we address in this paper, and they are not appropriate for incident investigations. However, some similar efforts with different goals exist, such as the work in [6], where accidents are detected by smartphones based on OBD-II messages.

In the next section, we introduce our approach and the basic elements of vehicular micro tracking.

3 Elements of micro tracking

If we want to reconstruct a traffic situation based on the CAN network traffic, we have to ensure the following:

1. we can collect and reliable store the CAN network traffic for a long period of time
2. we can understand the meaning of the most relevant CAN messages
3. we can reconstruct the movement of a vehicle based on the interpreted CAN messages

In the following subsections we show how the previous steps can be realized.

3.1 Collection of CAN messages

The Controller Area Network (CAN) is a broadcast network in nature, so the collection of sent messages can be implemented easily. It is enough to connect any CAN capable device to the network, and that device will receive any message sent on the network. Some vehicles use more than one CAN network for different purposes. In this case more than one log collector must be inserted into the network. The different collectors can save the data separately or to a central storage.

The log collector devices can be directly connected to the network by the manufacturer of the vehicle, or an external device can be connected to the On-board diagnostics (OBD) port of the vehicle. An OBD connector is mandatory for all vehicles sold in the European Union or in the United States of America for almost twenty years.

A CAN logger device can be bought from various vendors (such as VECTOR and many others), or can be constructed from general hardware. We decided to implement our own CAN logger because of the higher customizability. The following figure (

see Fig. 1) shows the prototype of the device used throughout the generation of this paper. This device consists of a Raspberry Pi 3 microcontroller and a SK Pang electronics PiCAN 2 shield [7]. This device can be directly connected to the OBD port of a vehicle, and our software automatically collects and stores every CAN message on the SD card. We verified our solution with an official CAN message logger and generator made by Inventure (WeCAN-USB) in generator mode. We found that in situations, where the bus load is below 50%, no message is lost by our device. We checked that in vehicles we used, the bus load rarely reaches the 30% level, thus our logger is capable to log every message accurately.

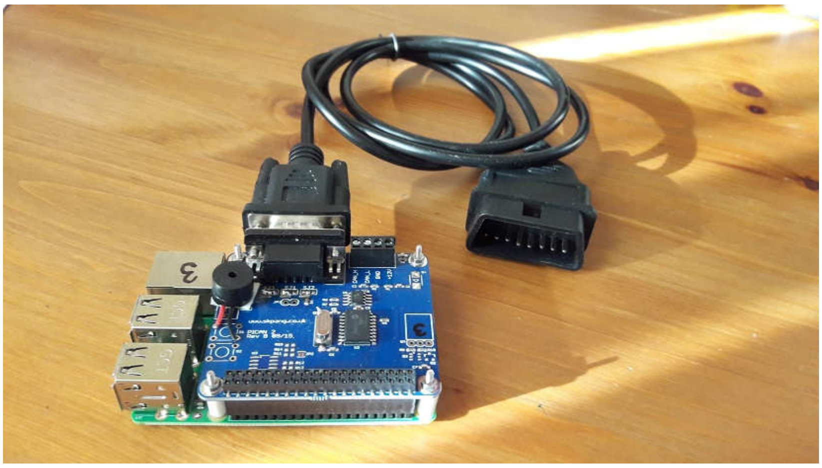

Fig. 1. Custom CAN logger based on a Raspberry Pi 3 microcontroller and an SK Pang electronics PiCAN2 shield.

3.2 Interpretation of CAN messages

The interpretation of the CAN messages is not standardized in most cases. Some heavy trucks related messages are standardized in the J1939 standard by the Society of Automotive Engineers (SAE) but no such standard exists for cars.

There are different efforts in reverse engineering and understanding different vendors CAN formats, but they are incomplete and unofficial. Some approaches that were useful for our work are the following:

- work of Charlie Miller and Chris Valasek with different brands such as Toyota, Ford and Jeep (http://illmatics.com/carhacking.html)
- community efforts such as www.canhack.de

We performed our own reverse engineering steps to analyze the CAN traffic. Based on the information gathered from the previously mentioned sources we defined various test scenarios. We monitored the CAN traffic in real time with a WeCAN device from Inventure Automotive Electronics R&D, Inc. This allowed us to focus on one functionality of a car at a time and find the corresponding information in the CAN traffic.

The main targets of the tests were to find the velocity and direction signals. In the CAN traffic the ID field is the base for the channel arbitration therefore it also represents the importance of the message. Ordering the messages according to the ID field value is helpful because than the most important parameters are to be found in the first couple of messages. This approach allowed us to find the engine main parameters like the rpm value easily. During this first steps, it was already clear that understanding any information in the message data is a big challenge because the message structure is very dense. The parameters of the vehicle are concatenated together with no separator or alignment. This is perfectly reasonable from an efficiency point but renders the reverse engineering process much harder.

The steering wheel position could be found relatively easily. The idea is to perform small left and right maneuvers and find a corresponding CAN value that oscillates around the middle value. Luckily the center value is represented with a 0 in our test vehicle therefore the small positive and negative values could be spotted fast.

The velocity value is not among the most important vehicle parameters. It is sent on the CAN bus first only for the dashboard. The fastest approach to find this information is to stop the car than start driving with a slow speed than stop again. A couple of repeated processes allowed us to find the speed of vehicle as well. Unfortunately, driving in reverse gear didn't help us because the velocity in that case is also represented with a small positive number.

3.3 Movement reconstruction

If we can understand the most important state variables of a vehicle, then we can reconstruct its trajectory. The most important variables are the following:

- steering wheel position
- velocity
- brake pedal position (optional, not used in this research)

These information is sent through the CAN network regularly (approximately once in every 100 milliseconds depending on the actual manufacturer and actual version). Based on the velocity and the timestamps of the CAN messages we can calculate the distance the vehicle travelled between two consecutive timestamps.

We checked this approach with a test car (Opel Astra Model H manufactured in 2005). We calibrated the test results by logging the CAN trace when travelling on a flat and straight road for different distances. The real and calculated distances can be seen on the following table:

Table 1. Real and calculated distances based on CAN messages.

Real distance (m)	Calculated distance (m)	Ratio: real / calculated
30	38.46	0.780
60	78.66	0.762
90	118.28	0.761
120	157.56	0.762
150	196.78	0.762

It can be seen that a compensation multiplier of 0.762 can lead us to quite accurate results. The reason of the existence of the multiplier is that different velocities are broadcast in different CAN IDs (e.g higher speed for the instrument cluster) at the same time. Also the actual speed heavily depends on the actual tire size, which is not known by the vehicle.

The distance travelled by the vehicle is not enough to reconstruct its trajectory. The direction is also fundamental for a correct path. Unfortunately, the actual direction of the vehicle is not sent on the CAN, but the position of the steering wheel is encoded in a CAN message type. From the steering wheel angle the angle of the front wheels can be calculated. The easiest way to find out the relation between the steering wheel angle and the front wheels' angle is by measurement. Fortunately, in our case, we found a simple linear relation between the two angles.

If we assume that no wheels are skidding on the road, then the line of the front and back wheels defines a circle whose radius can be calculated by simple trigonometry. This circle and radius defines the trajectory of the vehicle when the steering wheel is not in the central position. The following figure clarifies this situation.

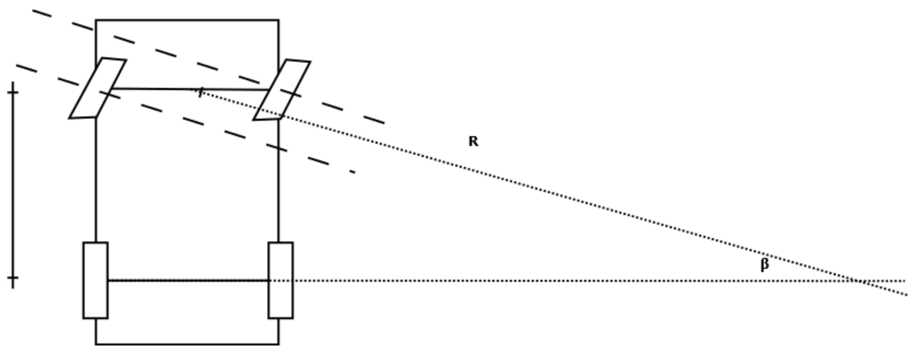

Fig. 2. Computation of the radius based on the vehicle length and the steering wheel position.

This is a simplified model of the vehicles movement, but we found it accurate enough to reconstruct trajectories. The accuracy of this approach was verified by going in circles by the test vehicle with a constant steering wheel position, and the diameter of the circles was measured. The measured and the reconstructed diameters was within a 5 percent range. The following diameters was used in the experiment: 8.6 m, 10.5 m, 12.7m, 18.9 m. From the actual velocity and the radius of the circle the test vehicle is going on, the actual position and direction of the vehicle can be computed by basic geometric computations.

Based on the actual position actual directions, the trajectory of a vehicle can be reconstructed. In the following we show how this information can be used to reconstruct real traffic situations.

4 Analysis and test results

For the analysis of the previously described method we searched for real life situations. As test case, we chose those scenarios that often occur in accidents. These emergencies usually contain sudden direction and speed changes or rapid maneuvering.

Among the most frequent cause of accidents are overtaking or obstacle avoidance. These two were the basis of most of our tests. Here, we present 3 of our measurements to demonstrate the effectiveness of our results. The first scenario is a lane changing maneuver, the second is an obstacle avoidance, and the third is a sudden direction change and intensive breaking.

4.1 First test case – lane change

In this test case the vehicle performed a simple maneuver: a lane change. After a short strait section a lane change was performed. The goal of the test was to clearly show the shift in the vehicle movement then the direction of the car remains unchanged.

Figure 3. shows that our approach is capable of effectively reconstructing this movement. The dots representing the position of the vehicle are shifted from one lane to the other.

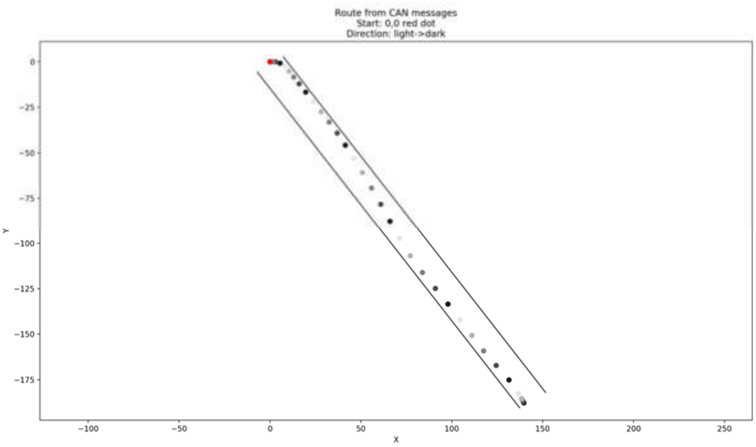

Fig. 3. Lane change maneuver

4.2 Second test case – obstacle avoidance

In the second test case the driver had to go around an obstacle and then continue further in the same lane. The goal of the test was to show direction changes during the maneuver which eventually realigns the movement with original direction.

Figure 4. shows that the reconstructed movement is exactly the same as the original one. After the driver passed the obstacle the direction of vehicle is the same again.

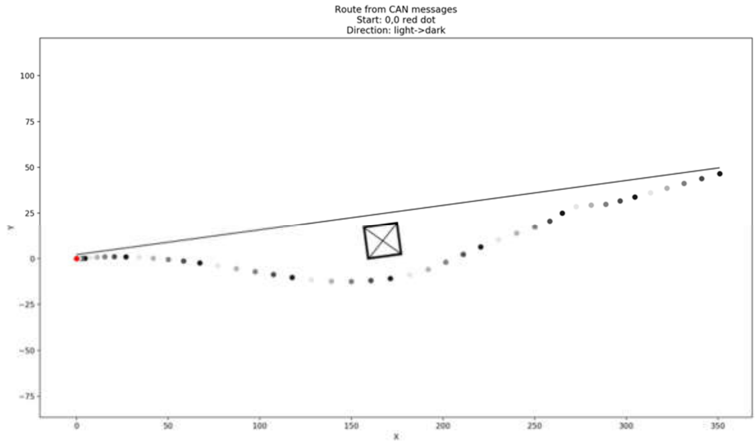

Fig. 4. Obstacle avoidance

4.3 Third test case – Sudden direction change

In the third test case the driver performed both a sudden direction and a sudden speed change. At the start of the test the driver achieved the speed of about 50 km/h. After about 150 meters there was a small curve in the road. Finally, at a previously defined point the vehicle was stopped with an intensive breaking and 90 degrees turn.

Figure 5. shows that the test movement can be accurately reconstructed in this case as well. During the strait sections the speed growth can be tracked based on the density of the dots. Furthermore, the angle of the curves is proportional to the real angle of the road.

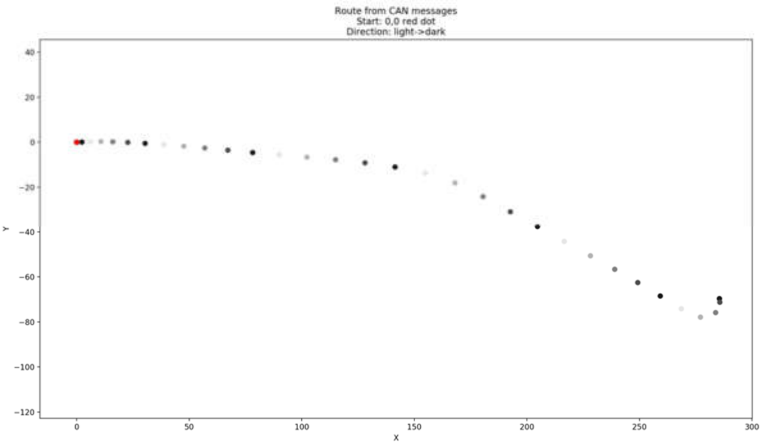

Fig. 5. Sudden direction and speed change

5 Acknowledgement

The work presented in this paper was partially supported from the grant GINOP-2.1.1-15. The project has been supported by the European Union, co-financed by the European Social Fund. EFOP-3.6.2-16-2017-00002.

6 Conclusion and future work

In this paper, we have shown how CAN network traffic can be used in accident reconstruction, and we argue that this kind of information can be a valuable source of data in forensic investigations after accidents. We showed our approach in reverse engineering CAN messages, the geometric base of trajectory reconstruction, and the effectiveness of our approach in real world situations.

In the future, we would like to test our approach with different vehicles from different manufacturers, and we want to reverse engineer different CAN messages such as brake pedal position or ABS signals to incorporate them into our approach to achieve even more reliable and accurate event reconstruction.

References

1. Brach, Raymond M. *Vehicle accident analysis and reconstruction methods*. SAE, 2011.
2. Van Kirk, Donald J. *Vehicular accident investigation and reconstruction*. Crc Press, 2000.
3. Searle, John A. *The physics of throw distance in accident reconstruction*. No. 930659. SAE Technical Paper, 1993.
4. Bosch, Controller Area Network Version 2.0 Protocol Standard, 1998
5. Mueller, C., Daily, J., and Papa, M., "Assessing the Accuracy of Vehicle Event Data Based on CAN Messages," SAE Technical Paper 2012-01-1000, 2012, https://doi.org/10.4271/2012-01-1000.
6. J. Zaldivar, C. T. Calafate, J. C. Cano and P. Manzoni, "Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones," *2011 IEEE 36th Conference on Local Computer Networks*, Bonn, 2011, pp. 813-819.
7. http://skpang.co.uk/catalog/pican2-canbus-board-for-raspberry-pi-2-p-1475.html