Log your car: The non-invasive vehicle forensics

Hafizah Mansor*, Konstantinos Markantonakis†, Raja Naeem Akram‡, Keith Mayes§ and Iakovos Gurulian¶
Information Security Group, Smart Card Centre,
Royal Holloway, University of London, United Kingdom
Email: {*Hafizah.Mansor.2011,‡RajaNaeem.Akram.2008,¶Iakovos.Gurulian.2014}@live.rhul.ac.uk
{†K.Markantonakis,§Keith.Mayes}@rhul.ac.uk

***Abstract*—Digital forensics is becoming an important feature for many embedded devices. In automotive systems, digital forensics involves multiple electronic control units (ECUs) used to support the connected and intelligent vehicle's technology. Digital evidence from these ECUs can be used in forensics investigation and analysis. Such a mechanism can potentially facilitate crash investigation, insurance claims and crime investigation. Issues related to forensics include the authenticity, integrity and privacy of the data. In this paper, the security of the forensic process and data in automotive systems is analysed. We propose an efficient, secure, privacy-preserving and reliable mechanism to provide a forensics data collection and storage process. A diagnostic application for smart phones, DiaLOG, is incorporated in the proposed process that uses a secure protocol to communicate the collected forensic data to a secure cloud storage. The proposed protocol for communicating forensic data is implemented to measure performance results and formally analysed using Scyther and CasperFDR with no known attack found.**

I. INTRODUCTION

Vehicle forensics is becoming an important feature in a vehicle's design and operational life cycle. Interested stakeholders include insurance claim investigators and law enforcement who are interested in crime and crash incident investigation. In recent years, the forensic feature has been further used by insurance providers and companies providing vehicles to their employees for business related activities.

An Electronic Control Unit (ECU) is a microcontroller that controls the operations of a car. In modern cars, there can be around seventy ECUs that control the overall operations of the vehicle [10]. Each ECU is responsible for different operations, such as body control, engine control and telematics. A telematics unit for example, provides connectivity (Wi-Fi or cellular network) to the car, through which the car is able to communicate with the outside world [34]. The different ECUs are connected within a car through networks such as Local Interconnect Network (LIN) [30], Controller Area Network (CAN) bus [13], FlexRay [15] and Media Oriented Systems Transport (MOST) [16]. The networks operate at different baud rates depending on the applications. The OBD-II (On-Board Diagnostic) port is a port that interfaces the outside world to the in-vehicle networks [31]. The port can be interfaced with a Wi-Fi, Bluetooth or serial connection using the ELM327 interface [6].

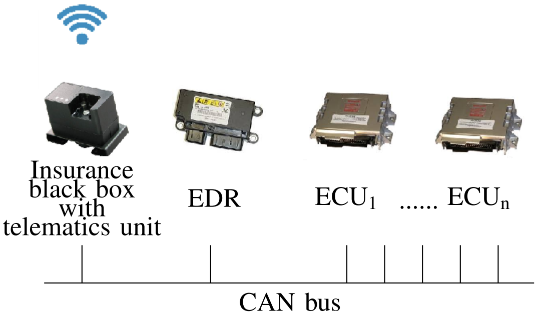

Fig. 1. Nodes on CAN bus

For digital automotive forensics, the two main and commonly used features are the Event Data Recorder (EDR) [1] and the insurance black box that works together with a telematics unit [34]. Fig. 1 shows the related nodes on CAN bus. An EDR is used to store data that is relevant to a crash incident. At least fifteen parameters are stored in order to be recovered during the forensic investigation [1], which include speed, seat belt status and airbag deployment state. The data is continuously stored and overwritten on the Random Access Memory (RAM) of the EDR, and the storage to its persistent memory (Electrically Erasable Programmable Read Only Memory (EEPROM) or flash memory) is triggered by a crash-like reading, for example a sudden change in the speed. The retrieval of data during the investigation is conducted by reading the content of the EDR, either through the OBD-II port or by physical extraction of the data memory of the EDR. There is however, a possibility that the data fails to be recorded due to a vehicle's electrical failure, which causes insufficient power to write to the EEPROM or flash memory of the EDR. The second possibility of storage failure is that the EDR module is defective. Whereas, an insurance black box continuously transfers the relevant parameters to the insurance company's server through the telematics unit, to monitor the driving style. The driving style is used to determine the insurance policy premium rate. A better and safer driving style will result in a lower premium rate. However, there are unresolved issues related to insurance black boxes, which include false data being transmitted to the telematics unit or the server and telematics data not being available.

A. Problem statement

(i) Current use of EDR and the insurance black box in forensic evidence provides limited features. The EDR gives a restricted number of parameters for analysis, whereas the insurance black box and telematics unit do not protect the privacy of the users. (ii) Although certain data is compulsory to obtain a service, users do not have control of the transmitted data and can not access it. (iii) The users are therefore unable to verify the correctness of the data being transmitted.

B. Contribution

In this paper, we propose DiaLOG, a mobile application that can be used as a forensic support feature. DiaLOG ensures only an authorised mobile device can be connected to the car, and provides integrity protected data that can be used for forensic investigation. It also protects the privacy of users and gives them control of the data being transmitted.

II. RELATED WORK ON AUTOMOTIVE FORENSICS

Nillson et. al discussed about performing forensics on in-vehicle networks [29]. They discussed an attacker model and requirements for detection, collection and event reconstruction. According to them, the features like diagnostic, firmware update, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications are desirable and could be achieved wirelessly, but these features introduced cyber attacks capability. In their forensic proposal design, their goals are to detect events in the vehicle (which require a device to detect, notify and store the forensic evidence); to answer the required questions in forensics (from the collected forensic data); and to obtain the current state of the firmware version. In order to perform this, a list of hashes of current firmware installed on all ECUs needs to be accessible. Their proposal described what to do, without providing any practical implementation. During the presentation phase of forensics, conclusions can be made from both the physical evidence through EDR data and also digital evidence through the network.

Hoppe et. al proposed a route reconstruction forensics in a hit-and-run scenario [11]. They proposed two methods, i.e manual and semi automated. Firstly, a Global Positioning System (GPS) receiver is installed in the car and connected to the ECU through the CAN bus. Any communication is logged in the data logger. The directions for navigation are displayed on the instrument panel cluster for safety and comfort. By having this feature, they propose to use it as forensic evidence by providing route reconstruction. In the manual method, the data from the data logger is manually analysed, and optimisation is conducted by filtering data that are potentially relevant to the incident. The semi automated method is by connecting a prober to the navigation unit and to a graphical user interface (GUI) to show the data read from the navigation unit. It is an invasive method since there is a physical connection to the unit required to get the relevant data. In order to log all the communication to the data logger, a large memory space is required. For example, according to [33], it shows there are 63 CAN IDs just to identify different operations of the car. A single operation is represented by a single CAN ID. As an example, just for the sideway acceleration sensor, there are 35 frames per second. For a car without a navigation system or a data logger, this might contribute to the cost of additional installation.

Kowalick discussed in detail about the unaddressed issues with automotive EDR [32] by the National Highway Traffic Safety Administrator (NHTSA), which is under the US Department of Transportation. Among the issues are the EDR data ownership, authenticity, the security at the time of crash and the chain of custody after the crash incident, tampering and manipulation, how data can be of use for civil/criminal proceedings, police's access authorisation on the data, possibility of developing EDR into a driver-monitoring tool, and third party's access authorisation.

III. AUTOMOTIVE FORENSICS

In this section, we discuss use cases, storage, retrieval, issues, security requirements and threat model of automotive forensics.

A. Automotive forensics use cases

Commonly, forensic data is used during crash incidents investigation. However, there are many other use cases when forensic evidence and data logging could be useful. As an example, a current trend adopted by insurance companies is to set the insurance premium rate depending on the driving style. The driving style of a driver is sent through a telematics unit to the insurance company's server. In another example, available technology like eCall service [3] is used during crash accidents, and can call emergency contacts. The service transmits location and time of accident to ensure rapid assistance. For criminal investigations, the GPS information could be used [2] to determine the location of a suspect.

Data logging can also be useful during car rental. Bob goes to the car rental company and is given a car. Similar to how the physical body checks are conducted, if Bob can use a diagnostic feature, he can verify the status of the ECUs, and the car as a whole (i.e. digital check). If the car is in a good condition, then Bob agrees to rent the car. Otherwise, Bob should notify the company about the current condition and he has the choice whether to rent the car or to choose a different car. If a crash or accident happens later, both parties, Bob and the car rental company, will have the evidence to prove the actual situation. Similar to the car rental use case, a potential buyer of a second hand car can conduct a diagnostic to ensure that all the units are working correctly as suggested by the seller.

For network intrusion, for example, an attacker, Mallory, is able to access the in-vehicle network, via the CAN bus network. He injects malicious messages to the CAN bus to cause denial of service (DoS), or to manipulate the operations of a car. The injection of malicious messages into the CAN bus network may cause a change in the normal frequency of messages [27], [28]. By having a data logging system, a driver/owner could be notified about the intrusion. Another example is in a valeting application, the car key is left to the valet service. Users do not want the valet to compromise their cars, either by trying to steal stored private information, or trying to trace their future locations by adding a malicious device (for example a USB stick or a malicious ECU) or application on the car.

The final example use case is diagnostic data. If there is any problem with the car, the car owner can conduct a first step diagnostic before visiting a workshop to get the problem solved. This will give a brief idea to the car owner of what should be fixed and its estimated cost. They can also benefit from assessing their driving styles from the stored data, and perhaps modify their style to reduce service costs and likelihood of accidents [12].

B. *Storage of forensic data*

The ECU could be the storage place for forensic data. Although, in case of total disruption to the car (and/or its units) caused by an accident, the data may not be retrievable. The data in the EDR could be corrupted or changed as a result of a bad retrieval technique [12], or it could be tampered with before the storage, e.g. by hacking the CAN bus and injecting malicious data. The cloud or a remote server could be a safe place to store forensic data, as long as the data is recoverable and protected from unauthorised access. The vehicle user should have control of what data is shared with third parties via the cloud or server. A mobile device could be another potential place to store the forensic data, however there are concerns on the tamper resistance of data since a mobile is easily accessible compared to an EDR. A mobile device would be a potential solution for a non-intrusive retrieval method. Compared to data retrieval directly from the car's black box, retrieving data from a mobile device or a cloud would only involve logical digital access rather than physical access. It is a user-friendly method and can be used as a first hand/first impression forensic result, and it reduces the possibility of causing any changes to the actual ECU that could invalidate the evidential data. The mobile could also be used as a backup or complementary unit for the EDR, as there are known problems related to the black box including failure to record, software and cable faults, OBD port retrieval issues, technical/training problems and permission issues [12]. Compared to an EDR, the mobile having a GUI through the application can help ensure that the data is always available. The owner or driver can also protect his interests by having the first hand forensic data. Forensic data are usually difficult to retrieve (requiring specialised tools and technical expertise if stored in the car). By having the data in the mobile device, the data is always available, where the owner or driver can have easy access, however, the data must be protected from any malicious tampering.

C. *Data retrieval of forensic data*

Parties interested in retrieving forensic data would include the law enforcement, lawyers, investigators (for insurance or police), and car manufacturers, as well as the vehicle owner/user. Law enforcement may be interested in the data to determine causes in accidents. Vehicle manufacturers may use this information in order to improve their vehicle designs and performance, and where possible to avoid or minimise the causes of accidents. State and highway officials maybe interested in the data to evaluate road conditions and safety. The driver or the car owner might be interested in the vehicle data, but may also attempt to modify or corrupt this data to conceal wrongdoing.

Data stored in a black box can be retrieved in three possible ways: firstly, through physical connection to the OBD-II port, then logically accessing the black box from the CAN bus; secondly, through physical connection to the ECU, then logically accessing the EEPROM or flash memory from the black box operating system and application; and finally, data can also be accessed through physical connection to the ECU's EEPROM or flash memory and conducting a memory dump. If the data is stored on a server, only authorised parties are able to retrieve the data.

D. *Issues related to automotive forensics*

A number of issues related to automotive forensics. One issue is on the privacy of data. The existing telematics unit provides connectivity to the car and sends the forensic data directly to a server (for example insurance black box) without the owner knowing what is transmitted. The access control authorisation of data should really be given to the owner although certain data is compulsory to obtain a service. There is a campaign to try and establish this right [7]. The owner also needs access to the stored and transferred data to verify that it is correct. It is also necessary to consider technical and security issues. The retrieval of data currently requires expensive specialised tools and expertise [12], although anyone can attempt to access private data via the vehicle network, accessible through the OBD-II port. Integrity and correctness of the stored data could not be verified in the existing system. The availability of data could be compromised if the automotive forensics is only relying on the availability of the EDR data [1].

E. *Security and privacy requirements*

From previous discussion, we conclude the security and privacy requirements for automotive forensics as follows:

(i) Integrity: It is crucial to determine if the data stored is not being tampered with or corrupted.
(ii) Authenticity: Data must be original and the person handling the data is authorised.
(iii) Availability: Ensuring all the required data is available for investigation, and updated all the time.
(iv) Reliability: Having a backup device for forensic data storage can increase the reliability of the forensic system.
(v) Privacy: It is important to protect the privacy of the car owner, especially when handling privacy-related data such as driving habits.

F. *Vehicle forensics threat model*

In automotive forensics, assets to protect are the read and write access authorisation and the authentication, integrity and privacy of the data. Potential attackers are untrustworthy workshops, owners, investigators and hackers with financial motivation. In our threat model, a malicious entity: (i) can access the CAN bus and manipulate the content before the storage (ii) can access and manipulate the content after the storage (iii) cannot break well-established cryptographic algorithms.

A number of possible attacks for automotive forensics can be conducted as follows:

(i) Denial of service (DoS) attack: To cause availability issue, where data stored is not able to be retrieved, or data not able to be stored. Denying access of data to an authorised party is also a method of DoS.

(ii) Impersonation attack: To impersonate an authorised party to conduct further attacks. For example, an attacker impersonating an authorised person to access the data during investigation to manipulate the content, or a device impersonating an authorised tool to access the data during the storage (i.e CAN bus manipulation). This will violate the authenticity requirement. Further attacks could lead to the violation of all other security requirements mentioned in section III-E.

(iii) Data manipulation attack: To change the content of the forensic data, either by changing the data before or after the storage, or during the retrieval process. This will violate the integrity property. Attacks can be performed mechanically by simply destroying the black box and its data contents. Attacks might also be mounted electrically which will cause disruption or change of data. Attacks through malicious CAN bus messages might distract the driver, or hack the engine or brake operation, or cause a crash and then erase all traces from the black box.

IV. PROPOSED SOLUTION

Commonly, during vehicle forensic investigation, the EDR, the infotainment unit and other ECUs are analysed [1]. In this proposal, the mobile application, DiaLOG, will be the backup data for the EDR and other related data that might be of interest for forensics.

Our proposal is based on the EVITA project [9], which proposed an embedded Hardware Security Module (HSM) in the ECU to ensure secure communications for on-board system. As proposed in the EVITA project, each ECU has its own HSM. This suggests that any node communicating through the CAN bus is required to have access authorisation in order to send or receive messages. In our proposal, the mobile device acts as a communicating node through the CAN bus, and so requires access authorisation.

To conduct a diagnostic on the car, the mobile device is connected to the OBD-II port via Wi-Fi or Bluetooth. Once connected, the mobile device will be authenticated, to determine whether it is authorised to retrieve the requested data. Once authenticated, the mobile device is connected to the CAN bus, and able to access the required data. The main idea of the DiaLOG application is to read the DTCs (Diagnostic Transmission Code) and log them securely. The DTCs can be read by the user of the DiaLOG application, and from there, the user is aware of the car's condition and state.

A. Assumptions

The assumptions of the proposed DiaLOG application are as follows: The mobile application is installed on a mobile device and the mobile device is available for investigation. The data is always automatically transmitted to the phone and later to the cloud. If data is not updated after a certain time, the owner will be notified. Finally, the cloud is securely managed. A user is authenticated to access the cloud server, and only authorised users have access to the data. However, even if an attacker is able to get access to the data in the cloud, our protocol protects the integrity and confidentiality of the data.

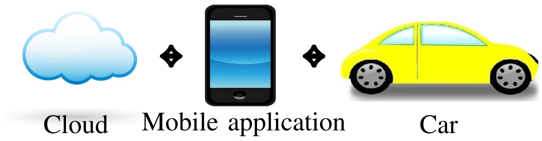

Fig. 2. Architecture of the proposed framework

B. The DiaLOG application

The architecture of the proposed framework with mobile application and cloud based backup storage is shown in Fig. 2. A mobile device with the DiaLOG application can log the latest vehicle operations. This data is uploaded to the cloud when a suitable network connection is available. From this framework, the forensic investigators have the options to get the data from three different sources: the car EDR, the mobile device or the cloud. The car owner (having the DiaLOG application and access to the forensic data) has control of what data to share with third parties. Certain data is compulsory to obtain a service and the car owner will need to give access to the service provider. However, he/she has the control of the transmitted data and can verify the correctness of data. The proposed architecture safeguards the privacy of the car owner/driver, in a way that is not possible with the current system that transmits data via the telematics unit. The keys for the mobile device are stored in a secure memory for example on a secure element, or the mobile device could be supporting TrustZone. The keys for the CCU are stored in the HSM of the CCU.

1) Authentication phase: In order to use the DiaLOG with the mobile device, the mobile device is required to be authenticated to the car. Only the authorised mobile device is given permission to access the data from the car, and most importantly to connect with the car's internal network. An authorised device is divided into two different levels: basic or full authorisation. These levels will be further explained in Section IV-C.

2) Diagnostic phase: In this phase, the mobile device is connected to the vehicle through a Wi-Fi connection, via an on-board router. The mobile device needs to be authenticated to the vehicle to ensure only authorised mobile devices are permitted to acquire the vehicle's diagnostic data. If authentication is successful, the mobile device will send a diagnostic command to the ECU, and the mobile will receive the resulting data. This operation is automated once connection is established and authentication is verified. This way, the driver is always aware of his/her vehicle condition. Apart from that, the consistency of data can be maintained between the mobile application and the vehicle.

3) Data logging: Data to be logged in the DiaLOG application are as follows:

(i) DTCs: are the error codes associated with the components in the vehicle. The main function of a diagnostic is to read the DTCs and to resolve the associated problems of the vehicle according to the codes.

(ii) ECU content is the firmware, application and data available in each ECU. To retrieve all the data in all the ECUs would be time consuming and require a large memory to store it all. The (concatenated) hashed value of each ECU can be stored to provide an integrity check. Using the architecture as proposed in the EVITA project [9], the master ECU contains all the hashed values of all the ECUs. Any changes in the content of the ECUs, i.e. any write operation to the flash, will change the hashed value stored in the master ECU. Hence, the master ECU is also alerted of the changes. The DiaLOG application data is also being updated accordingly.

(iii) Interface connection: to the vehicle is logged in the DiaLOG application. The authentication process from the interface connection will also be logged by DiaLOG to get the identity of the entity involved. The identification, authentication method, interface, time and location, which is related to the communication events (for example firmware update) are among the relevant data to be logged.

(iv) Crash-like data: There are two different ways the EDRs record the crash incident data. The first option is by continuously recording and overwriting the data on the EDR EEPROM/flash. The second option is by recording the data only when there is crash-like data. Similar to an EDR, crash-like parameters such as a sudden change in the velocity, could be a triggering factor for the DiaLOG application to start recording the required parameters for crash incidents. Triggering uses less mobile battery than continuous polling of data.

(v) Change in the frequency of messages through the CAN bus: could be an indicator of a potential remote attack being conducted. DiaLOG will record the normal frequency of messages and compare to the current operational frequency.

4) Storage to cloud: The user can transfer the data from the mobile device to the cloud. For example, after each driving cycle, all data is transferred to the cloud as a storage backup. At any required time, the data can be retrieved and analysed.

5) During forensics: Requirements during forensics include (i) Availability of mobile application data, (ii) Availability of ECU data, (iii) Authenticity, integrity and correctness of data. The latest diagnostic data stored in the mobile device is read during data collection. The ECU data, including the EDR is also read. During the forensic analysis process, this data is compared to ensure its consistency.

C. Protocols

There will be two levels of mobile device authorisation with different data access: basic and full, as shown in Table I. The protocol notations are shown in Table II.

1) Protocol description: There are two protocols depending on the different access authorisation.

TABLE I
CREDENTIALS FOR READ ACCESS

Data	User			
	Car owner	Car rental	Potential buyer	Investigator
DTC	✔	✔	✔	✔
Hash chains	✔	✔	✔	✔
External device	✔	✘	✘	✔
Crash data	✔	✘	✘	✔
Bus attack	✔	✘	✘	✔
Crash history	✔	✔	✔	✔
Authorisation	Full	Basic	Basic	Full

TABLE II
PROTOCOL NOTATIONS

CCU	Central Communication Unit
Mo	Mobile device of car owner (full authorisation)
Mt	Mobile device for temporary access (basic authorisation)
k_{mc}	Symmetric key for CCU (shared between Mo and CCU)
k_{temp}	Symmetric key shared between Mt and Mo
id_{Mo}	Identification number of mobile device of car owner
id_{Mt}	Identification number of mobile device for temporary access
id_{ccu}	Identification number of CCU
k_s	Temporary symmetric key shared between mobile device and CCU
pk_{ccu}	Public key of CCU used to verify signature
sk_{ccu}	Private key of CCU used to sign
n_c, n_{Mo}, n_{Mt}	Nonces
$reqdtc$	Request to access DTC
dtc	Diagnostic Transmission Code
ENC	Encryption using AES128
$sign$	Signature using RSA1024

(i) Full authorisation: For a full authorisation, the intended entity (e.g. the car owner) is required to be registered with the car manufacturer. After the installation of the DiaLOG application, a registration process is proposed. By registering, the mobile device, Mo, will have access to the car via a set of keys (k_{mc} and pk_{ccu}) provided by the car manufacturer. Car owners and law enforcement are given full authorisation provided they are registered with the car manufacturer. The key is a symmetric key shared between the mobile device and the car, k_{mc}. CCU is the central unit that interfaces the communication of the in-car ECUs to the outside world. The symmetric key, k_{mc}, is a medium term key that requires an update from the car manufacturer. It will be used to authenticate the mobile device to the car's CCU. Referring to Table III, the mobile device Mo will start the protocol by sending its ID, concatenated with an encrypted message using the pre-shared key k_{mc} containing ID of CCU, request to access the data and a generated nonce n_{Mo}. The CCU will then reply with its ID, concatenated with an encrypted message using the pre-shared key k_{mc} containing ID of CCU, nonce from Mo from previous message, n_{Mo} and a session key k_s. After a mutual authentication and freshness verification (Step 1-2), the mobile device will request the DTC from CCU. The dtc transmitted from CCU will then be encrypted to provide confidentiality and signed to ensure its integrity.

(ii) Basic authorisation: Entities included in this group are for example, anyone interested to rent a car from a company, or a potential buyer when a car is being resold. In order to be given access authorisation, the person interested is required to acquire a set of keys from the car owner. The key is transmitted and

TABLE III
FULL AUTHORISATION DATA ACCESS

1.	$Mo \rightarrow CCU$	$: id_{Mo} \Vert ENC_{k_{mc}}\{id_{ccu} \Vert fullreq \Vert n_{Mo}\}$
2.	$CCU \rightarrow Mo$	$: id_{ccu} \Vert ENC_{k_{mc}}\{id_{Mo} \Vert k_s \Vert n_{Mo}\}$
3.	$Mo \rightarrow CCU$	$: id_{Mo} \Vert ENC_{k_s}\{id_{ccu} \Vert reqdtc\}$
4.	$CCU \rightarrow Mo$	$: ENC_{k_s}\{id_{ccu} \Vert dtc\} \Vert sign_{sk_{ccu}}\{ENC_{k_s}\{id_{ccu} \Vert dtc\}\}$

TABLE IV
BASIC AUTHORISATION DATA ACCESS

1.	$Mt \rightarrow Mo$	$: id_{Mt} \Vert id_{Mo} \Vert ENC_{k_{temp}}\{id_{ccu} \Vert basicreq \Vert n_{Mt}\}$
2.	$Mo \rightarrow CCU$	$: id_{Mo} \Vert id_{ccu} \Vert ENC_{k_{mc}}\{id_{Mt} \Vert basicreq \Vert n_{Mo} \Vert n_{Mt}\}$
3.	$CCU \rightarrow Mo$	$: id_{ccu} \Vert id_{Mo} \Vert ENC_{k_{mc}}\{id_{Mt} \Vert k_s \Vert n_{Mo} \Vert n_c \Vert n_{Mt}\}$
4.	$Mo \rightarrow Mt$	$: id_{Mo} \Vert id_{Mt} \Vert ENC_{k_{temp}}\{id_{Mt} \Vert id_{ccu} \Vert k_s \Vert n_c \Vert n_{Mo} \Vert n_{Mt}\}$
5.	$Mt \rightarrow CCU$	$: id_{Mt} \Vert id_{ccu} \Vert ENC_{k_s}\{id_{Mt} \Vert id_{Mo} \Vert id_{ccu} \Vert n_c \Vert reqdtc\}$
6.	$CCU \rightarrow Mt$	$: ENC_{k_s}\{id_{ccu} \Vert dtc\} \Vert sign_{sk_{ccu}}\{ENC_{k_s}\{id_{ccu} \Vert dtc\}\}$

stored in the mobile device of the interested party. It is then used to authenticate the mobile device to the car. Basic authorisation only gives limited data accessibility as described in Table I. The key, k_s is only valid per transaction, i.e. once communication is disconnected, a new key is required to access the data again.

Prior to the start of the protocol, both mobile devices (Mo and Mt) share a symmetric temporary key, k_{temp} and public key of CCU, pk_{ccu}. Referring to Table IV, the temporary mobile device, Mt, will request an access to the car from an authorised mobile device (car owner's), Mo. Mt will send its ID, concatenated with id_{Mo}, and encrypted message using the preshared k_{temp} containing the CCU's ID, the request and a nonce, n_{Mt}. The car owner's mobile device will then send a message to notify the CCU about the temporary device's request which contains its ID, CCU's ID concatenated with an encrypted message using k_{mc} containing the temporary mobile's ID, the request, nonces n_{Mo} and n_{Mt}. The CCU will then acknowledge this by sharing k_s to the owner's mobile device. The owner's mobile device will then reply to Mt with its ID and the temporary mobile's ID, concatenated with an encrypted message containing id_{Mt}, id_{ccu}, k_s and all the nonces n_c, n_{Mo}, n_{Mt}. Now, the temporary mobile device can communicate with the CCU using the k_s. It will request the DTC and the CCU will reply with an encrypted DTC plus a signature to provide confidentiality and integrity.

D. Security Analysis

1) Informal analysis: Based on the security requirements in Section III-E, the proposal addresses them as follows:

(i) Integrity: The DTCs being transmitted and stored are signed by CCU to ensure that the DTCs are integrity protected.

(ii) Authenticity: of the communicating parties are verified for every protocol transaction. They are given access depending on the different levels of permission.

(iii) Availability: of the data is ensured by the update of data every time the authorised mobile device is connected to the car. The data on the cloud will be automatically updated once connection is available, or whenever the owner is notified.

(iv) Reliability: of the forensic system is improved by having a backup data on the cloud as well as on the mobile phone. If any of the stored data in the three different components does not match to each other, it shows that the data might be potentially corrupted.

(v) Privacy: of the car owner and its driving related data is protected. The owner has the control over the data.

Based on the threat model in Section III-F, the proposal addresses them as follows:

(i) Denial of service (DoS) attack: The data is always automatically transmitted to the phone and later to the cloud. If data is not updated after a certain time, the owner will be notified. Having a backup copy of the data can ensure that the data is available to be retrieved if the person is authorised. If the owner himself is the attacker, the data can always be accessible directly through the car's CCU or the EDR. If the investigator is able to get to the mobile device or the cloud data, then the data is always available.

(ii) Impersonation attack: Data can be retrieved by any entity having the correct authorisation, whether it is a full authorisation or a basic authorisation. Instead of using specialised tools, a mobile application provides easy data access without sacrificing authenticity of the person/tool in use.

(iii) Data manipulation attack: The content uploaded on the mobile device and cloud are integrity-protected by the use of signature by the CCU. Furthermore, since this proposal uses a mobile device, the cloud and also the ECU as the storage device to store the required data, there are three different components to verify the consistency of the data. All three components (ECU, mobile device and cloud) should have the same content of data. However, if content is manipulated by injecting malicious data through the CAN bus, all three components would have the same falsified data. However, our proposal is based on the EVITA project [9], where each ECU contains its own HSM and the communication through the CAN bus requires authentication. Therefore, any nodes communicating through the CAN bus are authorised.

2) Formal Analysis: The protocol is analysed using formal analysis tools to attain indicative results regarding its security. CasperFDR [14] and Scyther [5] tools are used to verify the protocol. The required security requirements include confidentiality of the secret keys, k_{mc} and k_s, and the authentication properties, which include aliveness, agreement and synchronisation. The full scripts can be found in the link: CasperFDR and Scyther input scripts.

For CasperFDR, the security properties verified are the secrecy, aliveness and agreement. The confidentiality property is to verify the secrecy of the k_s and k_{mc}, that are shared between the mobile device and the CCU. The aliveness property is to verify the aliveness between mobile device and CCU. The agreement property is to ensure the agreement of k_s shared between mobile device and CCU. The intruder has the knowledge of all the entities (CCU, Mo and Mt) and the request messages to access the data (fullreq, basicreq and reqdtc). Referring to CasperFDR full authorisation script, the script starts with #Free variables declaration, which declares all the variables used in the protocol. It is followed with the #Protocol description. This describes the messages being transmitted (in sequence) during the authentication and diagnostics, which starts with a request from MD to the CCU (i.e *1.a->b:a,*$\{$*b,fullreq,nmo*$\}\{$*kab*$\}$). In *5.a->b:a,reqdtc,*$\{$*b,reqdtc*$\}\{$*ks*$\}$, the MD requests the dtc after being authenticated and verified the freshness in messages 1-4. In the #Processes, all the involved entities in the protocol and their knowledge are declared. For example *INITIATOR(a,b,kab,nmo,fullreq,reqdtc)*, where a is the MD, b is the CCU and nmo is the random nonce generated by MD. MD knows kab which is preshared. The #Specification declares all the assertions made to verify the security properties. The confidentiality of k_{mc} and k_s are declared as *Secret(a,kab,[a,b])* and *Secret(b,ks,[a,b])*. As an authentication verification, the aliveness property and the agreement property between MD-CCU are verified. The #Actual variables section describes the names of the actual agents, and the actual variables such as MD and CCU. Nothing is declared in the #Functions section. The #System section again declares all the involved entities in the protocol and their knowledge, but with their actual names. For example, *INITIATOR(MD,CCU,KAB,Nmo,FULLREQ,REQDTC)*. The #Intruder Information declares the intruder X who has the knowledge of all the entities involved, i.e *IntruderKnowledge=*$\{$*MD,CCU,X,FULLREQ,REQDTC*$\}$. All the specifications made are verified and no attack is found for all the assertions.

For Scyther, the security properties verified are the non-injective synchronisation, non-injective agreement, weak agreement, aliveness and secrecy. The default verification setup was used (i.e. five as the maximum number of runs, type matching and to find the best attack with ten maximum patterns per claim). The secrecy property is to verify the confidentiality of the k_s and k_{mc}, that are shared between the mobile device and the CCU. Non-injective synchronisation property is to verify that parties know who they are communicating with, agree on the content of the messages and the order of the messages. The non-injective agreement is to verify that parties agreed on the content of the variables. In Scyther, all the security properties are modeled as role-based. Each entity is considered as one role. The properties are viewed from the local view of each role. Referring to Scyther basic authorisation script, the script starts with functions declarations. Then, we have macros of messages to make the script neat and easy to be followed. Next, the events and claims are made for each role (starts with MT, followed by MO and CCU). For example, for MT role, the events are *send_1 (mt,mo,m1)*, *recv_4 (mo,mt,m4)*, *send_5 (mt,ccu,m5)* and *recv_6 (ccu,mt,m6)*, which means MT sends the macro $m1$ to MO and later, receives macro $m4$ from MO and sends macro $m5$ to CCU to then receive macro $m6$ from CCU. Claims are the security properties to be verified. For example, for the MT role, *claim_R4 (mt, Secret, ks)* and *claim_R6 (mt,Secret, k(mo,mt))* are for confidentiality. Authentication properties are verified through Agreement (such as *claim_x3 (mt, Weakagree)*, *claim_x5 (mt, Niagree)*), Synchronisation (*claim_x4 (mt, Nisynch)*), and Aliveness (*claim_x6 (mt, Alive)*). The results for all the claims made are verified as “Ok” in the “Status” with “Verified” and “No attacks” in the “Comments”. This means that no attack was found within the bounded or unbounded statespace; the security property has been successfully verified [4].

E. Implementation

The proposed protocol was then implemented on a PIC Microchip microcontroller (PIC32MZ2048ECM144) and an Android device to obtain indicative performance results.

1) Implementation platform: Our approach of implementation is to observe the computation time on both the CCU and the mobile device separately. The mobile device communicates via Wi-Fi, while the CCU communicates via CAN bus. There is a Wi-Fi module connected to the CCU to receive the Wi-Fi messages from the mobile device and convert these messages into UART messages. There is another interface module between the Wi-Fi module and the CCU to translate UART messages into CAN messages and vice versa. Figure 3 shows the connection setup for CCU's communication. The CCU is simulated using a microcontroller with all the functions required to be an actual ECU with cryptographic engines. PIC32MZ2048ECM144 [26] is chosen as the implementation platform for CCU. It is a 32 bit microcontroller with 2048 KB of flash and 512 KB of SRAM, and operates at 200 MHz clock. It supports CAN bus communication, as required in an ECU. The hardware cryptographic engines support the computation of cryptographic algorithms to produce faster performance. For the mobile device, the application protocol is loaded into a LG Nexus 5 with a Quad-core 2.3 GHz Krait 400 CPU running on Android 5.1. There are two platforms used as the interface module to translate UART-CAN messages to compare different platform performance. They are PIC18F4580 and PIC32MZ2048ECM144. PIC18F4580 [18] is an 8 bit microcontroller with 32 KB of flash and 256 bytes of RAM. It operates with a 16 MHz clock and supports CAN bus and UART communication. For the Wi-Fi module, the Wi-Fi G demo board [25] is used.

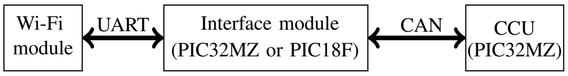

Fig. 3. CCU's setup for communication

2) Experiment setup: For the CCU setup, the simulation of the messages from and to the mobile application is using the Microchip CAN bus analyser tool [20]. The tool can be used to observe the messages sent from the PIC32MZ microcontroller and also send messages to it. On the PIC32MZ part, the PIC32MZ2048ECM144 starter kit [24] is connected to a CAN PICtail daughter board [21] through a starter kit adapter [23] and an I/O expansion board [19]. The CAN PICtail daughter board is then connected to the CAN bus analyser. The setup is shown in Fig. 4. The CCU's computation performance is measured based on cycle count given by MPLABX debugger.

The same setup is used for the interface module using PIC32MZ. For interface module using PIC18F4580, an additional CAN transceiver, MCP2551 [17], is connected to the PIC18. The interface module is then connected to MCP2200 breakout module [22] to observe the UART messages. The performance of communication is measured using oscilloscope. The performance of Wi-Fi communication is measured using "Inspector" feature from the internet browser.

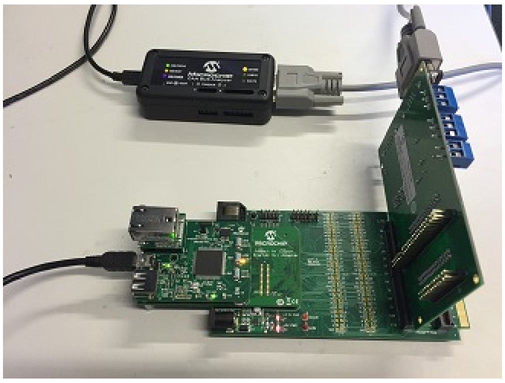

Fig. 4. Lab setup for CCU's CAN bus communication

3) Performance results: Based on the proposed protocol, the length of a message is more than eight bytes, hence, all the messages will need to be divided into more than one CAN message due to the limited number of bytes (8 bytes) of data per CAN message transmission. The messages are divided into three to eighteen messages to be transmitted via CAN. The computation and communication performance for full and basic authorisation protocols is as shown in Table V. The communication includes the transfer of data from Wi-Fi module to the middle interface module (via UART) and from middle interface module to the CCU (via CAN). Based on the results, it can be observed that the performance for the interface module is almost the same for the different platforms used. This is because both the devices use the same baud rates of communication, i.e for UART (at 9600 bps) and CAN (at 1 Mbps). The communication time can be further improved if CAN FD [8] is used, where one message can contain up to 64 bytes of data, instead of just 8 bytes.

V. CONCLUSION

As car operations are digitally controlled, security is now part of the main consideration in the automotive systems implementation. Our proposal is based on the new ECU architecture where a HSM is included in the ECU. By having a mobile application as a logging platform for the vehicle operation, it can help the forensic investigation to be more effective. More data options can be stored and thus increase the accuracy of forensic analysis. A secure framework for vehicle forensics is proposed to ensure the security of data and at the same time protect the users' privacy. The DiaLOG application proposed uses a new framework of automotive forensics, which provides usability and reliability. Our future work is to include more logging features on the DiaLOG application as discussed in Section IV-B3.

REFERENCES

[1] David Randall Peterman Bill Canis. "Black Boxes" in Passenger Vehicles: Policy Issues. Technical report, Congressional Research Service, 2014.

[2] Kangsuk Chae, Daihoon Kim, Seohyun Jung, Jaeduck Choi, and Souhwan Jung. Evidence Collecting System From Car Black Boxes. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–2. IEEE, 2010.

[3] European Commission. eCall Do You Have Any Concerns for Your Privacy? You Shouldn't. Technical report, Europian Commission, 2014.

[4] Cas Cremers. *Scyther User Manual*, draft edition, February 2014.

[5] Cas JF Cremers. The Scyther Tool: Verification, falsification, and analysis of security protocols. In *Computer Aided Verification*, pages 414–418. Springer, 2008.

[6] ELM Electronics. ELM327L. http://www.elmelectronics.com/DSheets/ELM327L_Data_Sheet.pdf/.

[7] FIA. My Car My Data. http://www.mycarmydata.eu/, 2015.

[8] Florian Hartwich. CAN with flexible data rate, 2012.

[9] Olaf Henniger. EVITA:E-Safety Vehicle Intrusion Protected Applications. Technical report, EVITA, 2011.

[10] Olaf Henniger, Ludovic Apvrille, Andreas Fuchs, Yves Roudier, Alastair Ruddle, and Benjamin Weyl. Security Requirements for Automotive On-board Networks. In *Intelligent Transport Systems Telecommunications,(ITST), 2009 9th International Conference on*, pages 641–646. IEEE, 2009.

[11] Tobias Hoppe, Sven Kuhlmann, Stefan Kiltz, and Jana Dittmann. IT-forensic automotive investigations on the example of route reconstruction on automotive system and communication data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7612 LNCS:125–136, 2012.

[12] David Hynd and Mike McCarthy. *Final report: Study on the Benefits Resulting From the Installation of Event Data Recorders*. 2014.

[13] Road vehicles – Controller Area Network (CAN) – Part 1: Data link layer and physical signalling. Standard, International Organization for Standardization, February 2013.

[14] Gavin Lowe. Casper: A compiler for the analysis of security protocols. *Journal of computer security*, 6(1):53–84, 1998.

[15] Rainer Makowitz and Christopher Temple. FlexRay- A Communication Network for Automotive Control Systems. In *2006 IEEE International Workshop on Factory Communication Systems*, pages 207–212, 2006.

[16] Media Oriented Systems Transport Specifications, 2006.

[17] Microchip. High-Speed CAN Transceiver. http://ww1.microchip.com/downloads/en/devicedoc/21667d.pdf, 2003.

[18] Microchip. PIC18F2480/2580/4480/4580 Data Sheet. http://ww1.microchip.com/downloads/en/DeviceDoc/39637c.pdf, 2007.

[19] Microchip. Starter Kit I/O Expansion Board Information Sheet. http://ww1.microchip.com/downloads/en/DeviceDoc/51950A.pdf/, 2010.

[20] Microchip. CAN BUS Analyzer Users Guide. http://ww1.microchip.com/downloads/en/DeviceDoc/51848B.pdf/, 2011.

[21] Microchip. CAN/LIN/J2602 PICtail (Plus) Daughter Board Users Guide. http://ww1.microchip.com/downloads/en/DeviceDoc/70319B.pdf/, 2011.

[22] Microchip. MCP2200 Breakout Module User's Guide. http://ww1.microchip.com/downloads/en/DeviceDoc/52064A.pdf, 2012.

[23] Microchip. PIC32MZ Embedded Connectivity (EC) Adapter Board Information Sheet. http://ww1.microchip.com/downloads/en/DeviceDoc/50002199A.pdf/, 2013.

[24] Microchip. PIC32MZ Embedded Connectivity (EC) Starter Kit Users Guide. http://ww1.microchip.com/downloads/en/DeviceDoc/70005147A.pdf/, 2013.

[25] Microchip. Wi-Fi G Demo Board Users Guide. http://ww1.microchip.com/downloads/en/DeviceDoc/50002147A.pdf, 2013.

[26] Microchip. PIC32MZ Embedded Connectivity (EC) Family. http://ww1.microchip.com/downloads/en/DeviceDoc/60001191E.pdf/, 2015.

[27] Charlie Miller and Chris Valasek. Adventures in Automotive Networks and Control Units. In *DEF CON 21 Hacking Conference. Las Vegas, NV: DEF CON*, 2013.

[28] Charlie Miller and Chris Valasek. A Survey of Remote Automotive Attack Surfaces. *Black Hat USA*, 2014.

[29] Dennis K Nilsson and Ulf E Larson. Conducting forensic investigations of cyber attacks on automobile in-vehicle networks. In *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information,*

TABLE V
FULL AND BASIC AUTHORISATION PERFORMANCE ON SAMSUNG GALAXY S5 MINI AND PIC32MZ

		Time(ms)						
		Computation			Communication		Total	
Protocol	Message	Mo	Mt	CCU	PIC32MZ	PIC18F	with PIC32MZ	with PIC18F
Full	1	1.972	-	0.055	57.686	57.605	59.713	59.632
	2	0.506	-	0.082	57.816	57.991	58.404	58.579
	3	0.458	-	0.037	42.472	42.423	42.966	42.918
	4	1.069	-	39.646	157.047	157.676	197.762	198.391
Basic	1	0.724	1.861	-	19.650	19.650	22.235	22.235
	2	1.956	-	0.059	34.864	34.832	36.879	36.847
	3	0.660	-	0.149	80.715	80.995	81.525	81.804
	4	0.922	0.500	-	19.650	19.650	21.072	21.072
	5	-	0.752	0.041	72.900	72.787	73.694	73.580
	6	-	0.994	39.651	157.047	157.676	197.692	198.322

and multimedia and workshop, page 8. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[30] Matthew Ruff. Evolution of Local Interconnect Network (LIN) solutions. In *Vehicular Technology Conference, VTC-Fall. IEEE 58th*, volume 5, pages 3382–3389. IEEE, 2003.

[31] SAE J1962 Revised APR2002. Standard, SAE Vehicle Electrical and Electronics Diagnostics Systems Standards Committee, April 2002.

[32] Kowalick Thomas Michael. Motor Vehicle 'EDR' Global Standardisation and Related Issues. http://onlinepubs.trb.org/onlinepubs/UA/111610Kowalick.pdf.

[33] Szia Vilg. Toyota PRIUS CAN ID. http://www.vassfamily.net/ToyotaPrius/CAN/cindex.html, 2008.

[34] Yilin Zhao. Telematics: Safe and Fun Driving. *IEEE Intelligent Systems*, (1):10–14, 2002.