A Security Architecture for Remote Diagnosis of Vehicle Defects

Kevin Daimi
Computer Science and Software Engineering
University of Detroit Mercy
Detroit, USA
email: daimikj@udmercy.edu

***Abstract*—Remote vehicle diagnostics within the auto industry will soon become a reality. Currently, all maintenance work including diagnostics is being performed by dealership. With the new setting of remote vehicle diagnostics, manufacturers will take the lead in the diagnostics process to improve their products and customer satisfaction. This paper proposes a high-level architecture for the remote diagnosis of vehicle defects. It then sets the ground for securing such an architecture due to the fact that safety and privacy of drivers and passengers are extremely challenging with the manifestation of security breaches.**

Keywords—Remote Vehicle Diagnostics; DTC, ECUs; Telematics; Security Architecture; Security Policy

I. INTRODUCTION

Modern vehicles utilize a number of buses in the in-vehicle networks. These buses include Local Interconnect Network (LIN), Controller Area Network (CAN), Media-Oriented System Transport (MOST), and FlexRay. LIN handles the lowest data-rate functions, such as door locks, climate control, and mirror control. CAN fits medium speed applications including body systems, engine management, and transmission. High-speed data rates are dealt with by MOST, and therefore, it is convenient for multimedia and infotainment. Finally, safety-critical applications, such as steer-by-wire, stability control, and brake-by-wire are managed by the FlexRay [1]-[5]. Connected to these buses are various Electronic Control Units (ECUs). Modern-day vehicles are furnished with over 80 embedded electronic control units (ECUs), which oversee an enormous part of their functionality. This functionality spans a broad collection of tasks including overseeing door looks, climate, sunroof, body systems, transmission, advanced safety and collision avoidance systems, and pressure monitoring systems. On each ECU, a dedicated and independent firmware runs to control these tasks. ECUs acknowledge signals from various sensors located at various parts and in different components of the vehicle. Using these signals, ECUs control various critical units in the vehicle [6]-[10].

The entire network, including the buses and the ECUs, demands protection against security attacks. Some analyses of the buses, especially the CAN bus, have spotted various vulnerabilities in the available in-vehicle network protocols [11] [12]. All the potential attacks on cellular networks will find their way to the vehicle and can impact the ECUs. Therefore, it is critical to enforce the security of the buses and ECUs when remotely diagnosing problems of various parts of the vehicles controlled by these ECUs.

Vehicles experience various defects. Some of these defects are considered safety-critical, while others are non-safety critical faults. Examples of these defects include problems with fuel consumption system resulting in fuel leakage and possibly a fire, broken or stuck accelerator controls, unexpected rupture of the engine cooling fan blades, improper operation of windshield wiper assemblies, wiring system problems that result in a fire or loss of lighting, a defect in child safety seats, inadequate operation of air conditioning and radio, ordinary wear of shock absorbers, batteries, brake pads and shoes, and exhaust systems, and excessive oil consumption. The vast majority of vehicle defects result in issuing Diagnostic Trouble Codes (DTCs), which are collected by the Electronic Control Units (ECUs) overseeing the operation of these components. Faulty ECUs or bus errors can also result in defects including many of the stated defects above. Currently, all repairs and maintenance are performed by vehicle dealerships. A future trend within the auto industry would be to execute these fixes remotely. This approach will save auto manufacturers a huge amount of money including penalties payed as a result of casualties arising from these defects and from recalls, help manufacturer discover potential recalls ahead of time, and improve their products using the big performance data that will be available. Dealerships' time will be saved through receiving the diagnosis and fixing procedures directly from the manufacturer site. Vehicle owners will feel safer, have increased trust in their vehicle's manufacturer, and save considerable amount of time including the time spent at the dealership. Obviously, for systems providing remote diagnosis to be productive and efficient, security is inevitable.

Pant, Pajic, and Mangharam [13] utilized an automotive ECU architecture for communications between the vehicle and a Remote Diagnostics Center to diagnose, test, update and verify ECUs' firmware. Their diagnostics scheme concentrated on both real-time and non-real time defects, and involved a decision making function to perceive and isolate faults in a system with modeling uncertainties. The suggested framework incorporated in-vehicle and remote diagnostics with the goal of making vehicle recalls management cost-effective. They only used three units in their approach. Their scheme completely ignored security enforcement.

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-562-3

A development of a prototype application for remote vehicle diagnostics, based on the Diagnostics over IP (DoIP) protocol was presented by Johanson, Dahle, and Söderberg [14]. Basic manipulation experiments with synchronous remote diagnostics read-out and control were portrayed. Various safety related concerns requiring closer investigation before a visible exploitation of remote diagnostics services becomes feasible were ascertained. Furthermore, a taxonomy of vehicle diagnostics applications was postulated. This was proposed to interpret the divergences between synchronous (online) and asynchronous (offline) setups in local and distributed settings. Their system merely dealt with remote vehicle diagnosis with no reference whatsoever to securing the remote vehicle diagnostics.

Ferhatović, Lipjankić, Handžić, and Nosović [15] introduced the implementation of a straightforward system for the diagnostics of vehicle faults. Their implementation deployed the standard diagnostic trouble codes and relied on a client-server setting. They presented some functionality and algorithms for that purpose. The communication link between the client and the server was achieved through mobile phone. There was no connection with the manufacturer site. Furthermore, securing the diagnostic process was not an option.

Oka, Furue, Bayer, and Vuillaume [16] introduced an analysis of the security properties for remote diagnostics with some overview of possible attacks. They investigated and categorized diagnostic services and examined mainly their suitability for being remotely performed. They later pinpointed relevant security properties for each of the suitable diagnostic service category. They indicated they will consider the security between the ECUs and telematics module and between the telematics module and the OEM server. However, no message was encrypted and no key management system was provided. Furthermore, authentication, integrity and confidentiality was loosely mentioned. They used only three components, ECUs, telematics module, and OEM server.

This paper presents a security architecture for remote vehicle diagnostics. The architecture includes a number of components. The vehicle site has three components: ECUs, Driver Interface Unit, and the Telematics Module. The Telematics Server, Diagnostics Engine, Knowledge Base Manager, and the Performance-Historical Data Manager reside at the manufacturer site. There are also two external components: Dealership Control Unit, and Supplier Control Unit. The heart of this architecture is the Security Engine. The remainder of the paper is organized as follows: Section II will discuss the use case scenario for the architecture. Section III will introduce the security policy. The remote vehicle security architecture is presented in Section IV. The paper is concluded in Section V.

II. REMOTE DIAGNOSIS SCENARIO

Figure 1 is used to explain the remote diagnostics scenario. This scenario will be carried out without reference to the Security Engine (SE) to better understand the technical concepts of remote diagnosis. In the next section, security will be introduced. The symbols used are collected in Table 1 below. The remote diagnosis scenario is depicted in the following use case:

(1) When a problem occurs, Diagnostic Trouble Codes (DTCs) are generated.
(2) The DTC's are stored in the respective ECU's memory. In other words, the ECUs write down the conditions existing when the fault occurred and store them in their memory. The DTCs could also be distributed among several ECUs.
(3) The Onboard Diagnostic System (OBD-II) has access to these DTCs. Other information is also stored when the trouble occurs. This includes vehicle speed, engine RPM, engine coolant temperature, open/close states of the valves, and vehicle emission-related data required by law.
(4) The Telematics Module (TM) of the vehicle communicates the problem-related information from the OBD-II to the Telematics Server (TS) of the manufacturer.
(5) TS analyzes the uploaded information to see if further details are needed, and adds the vehicle VIN number and the diagnostics ID number (DID).
(6) TS transfers all this information to the Diagnostics Engine (DE) at the manufacture site.
(7) DE receives commands from the Diagnostic Center (DC) to start the diagnosis. The DE is in charge of the actual diagnosis. It behaves like an expert system for diagnosis.
(8) Diagnostics Engine extracts the possible symptoms from the diagnostics information.
(9) DE communicates with the Knowledge Base Manager (KBM) and provides the found symptoms.
(10) KBM consults its knowledge base (KB) to see if a solution can be found based on these symptoms.
(11) If further information is needed, DE will be consulted. It is possible that DE will contact the Telematics Server if it cannot provide what the Knowledge Base Manager asks for.
(12) The Knowledge Base Manger contacts the Diagnostics Engine and provides its findings. Here, either a solution is found or no solution exists.
(13) If KBM is unable to provide a solution using its knowledge base, DE will use its diagnostics algorithms to find a possible solution.
(14) If DE cannot find a solution, it will get in touch with the firmware Supplier Control Unit (SCU) residing at the supplier site to provide diagnostics information and symptoms.

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-562-3

(15) The SCU provides the solution. The solution can be updating the firmware of the ECUs that faced the problem, or completely flashing the firmware of the ECU to install a new firmware.
(16) When the solution to the problem is found by the Diagnostics Engine and that solution does not need the dealership to be involved, the commands to fix the problem are sent to the TS. This is further elaborated in steps 20-21.
(17) The Diagnostics Engine sends the fixes to the Knowledge Base Manager to update the knowledge base. It also sends the diagnostics details including symptoms and fixes to the Performance-Historical Data Manager (PHDM) to update the vehicle's performance and historical data stores. These will be very valuable assets for business intelligence.
(18) If there is a need to have the vehicle's engine turned off, the Telematics Server will inform the TM.
(19) The TM transmits a message to the Driver Interface Unit (DIU) to have the driver turn the engine off. When that happens, the TM informs the Telematics Server.
(20) TS sends messages containing the fixes in a form of diagnostics commands to TM.
(21) The TM communicates with the ECUs in question and the fixes will be applied.
(22) If the solution involves more work than just simple fixes, such as new update and ECU flashing, and there is no Firmware Over-The-Air (FOTA), the dealership must be involved.
(23) If the option of FOTA exists, the TS communicates with the TM to achieve that. In this case, the vehicle must not be running.
(24) If the dealership is needed, the Telematics Server will help the Diagnostics Engine in scheduling an appointment for the vehicle. It will communicate with TM requesting the dealer's name and address, and date and time of the appointment.
(25) The TM communicates a message to the DIU informing the driver of the problem and requesting the name and address of the dealer, and the date and time of dropping the vehicle.
(26) The received information from the DIU is sent to the TS via the TM.
(27) The Diagnostics Engine communicates with the Dealership Control Unit (DCU) at the dealership site. The DCU will receive the symptoms and fixes in addition to the date and time of the appointment. If there is a need to change the date/time, the TS will re-contact the TM. The scheduled date should also give the dealership enough time to prepare spare parts if needed.
(28) The vehicle will be fixed.
(29) If a new update or a completely new firmware is needed as a result of the problem in the vehicle in question, a recall will be issued by the manufacturer for all vehicles of that model and year.

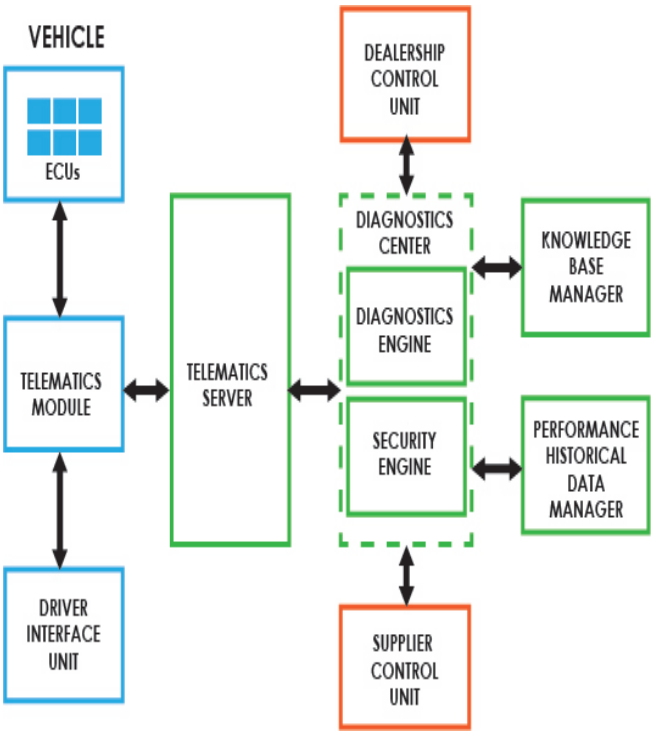

Figure 1. Vehicle remote diagnosis security architecture

TABLE I. SYMBOLS USED

Symbol	Meaning
DTC	Diagnostics Trouble Code
ECU	Electronic Control Unit
TM	Telematics Module
DIU	Driver Interface Unit
TS	OBD-Based Telematics Server
DC	Diagnostics Center
DE	Diagnostics Engine
SE	Security Engine
KBM	Knowledge Base Manager
PHDM	Performance-Historical Data Manager
DCU	Dealership Control Unit
SCU	Supplier Control Unit
OBD-II	Onboard Diagnostic System
VIN	Vehicle Identification Number
DID	Diagnostics Identification
PU	Public key
SK	Symmetric key
PR	Private Key
PDS	Performance data store
HDS	Historical data store
→	Then in Section III, Sends in section IV
← →	Both parties apply security requirements
X_S	Parties communicating using SK
MAC	Message Authentication Code
KB	Knowledge base

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-562-3

III. SECURITY POLICY

Security policies mandate what must be secured, and how to secure them to support the security architecture or the network security. Without a security policy, a network may be compromised. With the intention of safeguarding access to various components of an information system, a network security policy should be developed. It consists of a list of conditions and actions to prevent illegitimate access to private information. Network security management has been focusing on security policies to the extent that security policy repositories are at the core of many network security management systems. A security policy furnishes the basis for system security architecture [17]–[22]. In what follows, the security policy is represented by rules.

IF component = SE → component can access {TM, TS, SCU, DCU, DE, KBM, PHDM}
IF X is an algorithm and X belongs to the list of algorithms {RSA, EEC, AES, SHA-3, HMAC, DDA, CMAC, CTR, CFB, …} approved by SE → X can be used
IF X is a component & Y is a component & X and Y are allowed to communicate & X has algorithm Z & Y does not have algorithm Z → Z cannot be used
IF component = ECU | DIU → only TM can access them
IF component = DIU → ECUs cannot access component
IF component = SCU | DCU → component is not allowed to receive info about driving habits of the vehicle's owner including speed, route and the location of the vehicle when the fault occurred
IF component = TM → only TS can access component
IF component = TS → only TM & DE can access it
IF component = DE → only DCU, SCU, TS, KBM, & PHDM can access component
IF component = KBM | PHDM | DCU | SCU → only DE can access component
IF M is a message & M is sent to DCU | SCU → DCU | SCU cannot deny receiving M
IF M is a message & M is encrypted with PR | M is encrypted with SK → M is authenticated
IF M is a message & M is encrypted with Private Key | MAC(M) is encrypted with PR → M is signed
IF M is a message & X is the sender & Y is the receiver & X encrypts M with Y's Public Key | X encrypts M with SK → M is confidential
IF K is a key & SE did not distribute this key → K cannot be used
IF K is a key & X is a component & K is issued by SE → X must receive the validity period of K from SE
IF X is a component → X must have its own Intrusion Detection System
IF X is a component & A is a malicious activity & X detected A → X must notify SE immediately
IF X is a component & A is a malicious activity & X detected A →X must stop its communication
If X is one of the data stores in {PDS, HDS, KB} → X must be encrypted
IF X = PDS | HDS → X is only accessed by PHDM
IF X = KB → X is only accessed by KBM
IF X is a component & M is a message & M is received by X at time = t & M is received again by X at time = t +1 & … → X must terminate communication
IF X is a component & M is a message & M does not belong to the set of messages allowed for X → X denies M & X informs SE
IF X is a component & Y is a component & X and Y are allowed to communicate & P is a protocol & P approved by SE → X and Y can use P
IF component X belongs to {ECU, DIU, TM} → X's outgoing messages are encrypted with PU
IF component X belongs to {TM, DE, TS, PHAM, KBM, DCU, SCU} → X's sent messages are encrypted with SK
IF X is a component & Y is a component and X communicates with Y and D is an auditor → D may access HDS
IF X and Y are components & A is a malicious activity & X detected M | Y detected M & A is blocked → X and Y may continue their communication

IV. REMOTE DIAGNOSIS SECURITY ARCHITECTURE

The Remote Diagnosis Security Architecture (RDSA) will be explained below using the available communication between various parties. The messages sent will be represented symbolically including the type of encryption. In what follows, (PU, messages) is used to indicate that public key cryptology is used, (SK, messages) implies using symmetric key cryptology. This will be followed by the security requirements (enclosed by parentheses) applicable to the message.

A. The Security Engine

The heart of the remote diagnosis security architecture is the Security Engine (SE). Note that the connection between TM and SE in Figure 1 has been omitted for clarity purposes. The Security Engine is responsible for symmetric keys distribution and management, updating keys, issuing keys for Message Authentication Codes, and ensuring the security policy is not violated. It further controls the cryptographic algorithms and techniques used for encryption/decryption and message authentication. Initially, parties communicating based on symmetric cryptology have a preinstalled symmetric key that will be used just once by SE to forward the newly created symmetric key for each pair of parties. Once those parties receive these keys, the pre-installed ones are discarded. SE also uses the created symmetric keys to communicate the keys needed for message authentication. As part of the security policy, the Security Engine informs each pair of communicating parties what techniques they are allowed to use. The parties can agree on a subset of techniques out of the allowable set of techniques approved by the SE during handshaking. The

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-562-3

relationships below illustrate what the SE sends the parties (X_S) communicating using symmetric key.

SE → X_S: (Symmetric Key, MAC-Key, Algorithm Set)

B. TM, ECUs and DIU Communications

The Telematics Module communicates with both the ECUs and the DIU. The public and private keys for TM, DIU, and ECUs are preinstalled at manufacturing time. In addition, the ECUs and DIU have the public key of TM preinstalled, and TM has the public keys of DIU and ECUs preinstalled. To replace the pre-installed keys, ECUs and DIU must change their public and private keys and use the old private key to encrypt the new public key before sending it to TM. TM will create its new public and private keys and forward its public keys encrypted with the old ones to the ECUs and DIU. This procedure will be followed every time the Telematics Module issues a request to replace keys. TM receives messages containing the DTCs and other vehicle status information when the fault occurred, such as vehicle speed, coolant temperature, and engine RPM from the ECUs and OBD-II system. TM and ECUs authenticate each other. The messages containing DTCs and status information are encrypted using public key cryptology. Confidentiality of the exchanged messages is enforced, and the integrity of these messages is verified to ensure the messages have not been modified. On the other hand, the messages sent by TM to ECUs include remediation commands (fixes). These are also encrypted with public key. Confidentiality, integrity, and authentication are required. The same applies to the interaction between TM and DIU. The messages sent by the TM to DIU include problem, turning engine off, providing dealer address, dealer name, date of appointment, time of appointment and appointment details when it is scheduled. The DIU sends engine turned off, dealer address, dealer name, and requested date and time of appointment to TM. They first agree on the algorithms to be used for encryption and MAC, nonce(s), and the allowable waiting time for receiving a message to overcome replay attacks. The relations below exemplify these messages.

TM → ECU: (PU, remediation commands)
ECU → TM: (PU, DTCs, status info when fault occurred)
TM ←→ ECU: (confidentiality, integrity, authentication)
TM → DIU: (PU, turn engine off, request for dealership details, request for appointment details)
DIU → TM: (PU, engine off, dealer details, requested appointment date and time)
TM ←→ DIU: (confidentiality, integrity, authentication)

C. Telematics Server and TM Communication

The Telematics Server is the only component that can communicate directly with the vehicle. Virtually, it can provide various information to the vehicle through the TM. However, only the messages needed for this architecture will be introduced. Confidentiality, integrity, and authentication are also needed. The TS receives problem-related information from the TM. The TM accepts fixes messages, inquiries for further problem information, requests for turning the engine of the vehicle in question off, request for dealership details, and driver preferred appointment date and time. The symbolic representation for this communication is given below.

TS → TM: (SK, fixes, further info request, dealership info request, appointment date/time request)
TM → TS: (SK, problem-related info, dealership info, appointment date/time, engine off, VIN)
TM ←→ ECU: (confidentiality, integrity, authentication)

D. Telematics Server and Diagnostics Engine Communication

This is an internal communication within the manufacturer's site. The Telematics Server supplies the problem related information received from the TM to the DE after adding the VIN number of the vehicle and the Diagnostic ID (DID). The VIN number will help in retrieving further information about the vehicle in question if needed, and DID will designate the fault and will be used for indexing purposes. The TS will also provide further details from the TM if needed by the DE. The Diagnostics Engine will check if a solution exists, try to find a solution, and contact the supplier of firmware if it fails. In any case, a remediation procedure is sent back to TS. This includes fixing commands if there is no need to involve the dealership, request for scheduling appointment for the vehicle, need for further information about the problem, and request to turn the engine off.

TS → DE: (SK, problem-related info, further info, dealership info, appointment date/time preference, engine turned off, VIN, DID)
DE → TS: (SK, remediation procedure, dealership info request, appointment date/time, engine off request)
DE ←→ TS: (confidentiality, integrity, authentication)

E. Diagnostics Engine and KBM Communication

Prior to applying any diagnostic algorithms, the Diagnostics Engine consults the Knowledge Base Manager to see if any solution exists in the diagnosis knowledge base. It provides the KBM with all the symptoms of the problem, which are extracted from the DTCs. The Knowledge Base Manager will reason about its knowledge base using the provided symptoms. If a solution is already stored, KBM will send its details to DE. Otherwise, a "Solution does not exist" message is forwarded. If no solution exists, the DE will try solving it itself. If it finds a solution, it sends this knowledge to the KBM to be stored in the diagnosis knowledge base. In other words, the knowledge base is augmented. The DE will use the VIN

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-562-3

number to get any other needed information about the vehicle.

DE → KBM: (SK, symptoms, further info about Vehicle, DE's solution, DID)
KBM → DE: (SK, KBM's solution, DID)
TM ←→ ECU: (confidentiality, integrity, authentication)

F. Diagnostics Engine and PHDM Communication

As a result of various diagnoses, diverse important data is accumulated. Some of this data will be stored in the Performance Data Store (PDS) and the rest in the Historical Data Store (HDS). Examples of the data stored in the Performance Data Store are DTC's, symptoms, various vehicle status information when the problem occurred, solution, vehicle model and year. The HDS will include the above and other communication messages in the architecture. All this information is forwarded by the DE to the Performance Historical Data Manager to be stored in PDS/HDS. These two data stores will accumulate big data that will be used by the manufacturer for a range of analyses and statistics. These analyses and statistics are beyond the scope of this architecture. However, the PHDM does provide some simple statistics on the number of performance records for certain vehicle models and years, and number of historical records for all vehicle models and years.

Performance Data = {DTC's, symptoms, various vehicle status information when the problem occurred, solution, vehicle model, model year}
DE → PHDM: (SK, performance data, all other communication messages)
PHDM → DE: (SK, performance statistics, historical statistics)
PHDM ←→ DE: (confidentiality, integrity, authentication)

G. Diagnostics Engine and DCU Communication

When the problem needs the dealership's interference, the DE informs the Dealership Control Unit at the dealership site. This is an external communication outside the manufacturer site. The dealership receives the diagnosis, remediation procedure, and the needed firmware or firmware fixes. Furthermore, DCU receives the information of the driver and details of the appointment. The dealership submits the details of fixing the vehicle and any possible functions in the vehicle impacted by the maintenance process. If the vehicle cannot be fixed, the DE will re-contact the firmware Supplier Control Unit. Here, the security requirement, nonrepudiation, is required to prevent the dealership from claiming it did not receive the messages sent by DE.

DE → DCU: (SK, diagnosis, remediation procedure, ECU firmware, driver details, appointment details)
DCU → DE: (SK, maintenance details, other functions impacted)

DCU ←→ DE: (confidentiality, integrity, authentication, nonrepudiation)

H. Diagnostics Engine and SCU Communication

When the DE is unable to find a solution for the problem, it contacts the firmware Supplier Control Unit. This is also an external communication that needs nonrepudiation to be applied. The SCU must receive the DTCs, the state of the vehicle when the problem occurred, DE's analysis of the problem and trials stemming from DE's attempts to fix the problem, and vehicle model and year. On the other hand, the SCU provides firmware update, firmware fixes, or new firmware, and affected ECU. Certainly, the vehicle model and year will be attached.

DE → SCU: (SK, DTCs, vehicle state, DE's analysis, model, year)
SCU → DE: (SK, firmware update, firmware fixes, new firmware, affected ECU, model, year)
SCU ←→ DE: (confidentiality, integrity, authentication, nonrepudiation)

V. CONCLUSION AND FUTURE WORK

This paper presented a comprehensive architecture for remote diagnostics of vehicle defects. This architecture is enhanced by adding a Security Engine to oversee and coordinate all possible security functions, procedures, policies, and key creation and distribution. Traditionally, the Telematics Control Unit (TMU) is in charge of telematics. Because TMU is an ECU, all the limitations of ECUs including message size apply here. If TMU is used in the above security architecture, public key cryptology would have been the right choice for its communication with TS because TMU can only handle very short messages. A new trend in vehicle industry is the use of a more powerful unit, the Telematics Module. This is included in the above architecture. A future improvement would be extending the system to deal with the buses defects, especially, the CAN bus errors. Here, another component will be added. For systems of remote diagnosing to be trusted, driver privacy must be enforced when sending the information to the manufacturer site. The next step after expanding the architecture would be implementing it.

REFERENCES

[1] On Semiconductor, "Basics of In-Vehicle Neworking (INV) Protocols," http://www.onsemi.com/pub_link/Collateral/TND6015-D.PDF, pp. 1-27, [retrieved: April, 2017].

[2] Freescale Semiconductors, "In-Vehicle Networking," https://cache.freescale.com/files/microcontrollers/doc/brochure/BRINVEHICLENET.pdf, 2006, pp. 1-11, [retrieved: April, 2017].

[3] S. Seo, J. Kim, S. Hwang, K. Kwon, and J. Jeon, "A Reliable Gatway for In-Vehicle Networks Based on LIN, CAN, and FlexRay," ACM

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-562-3

Transcation on Embedded Computing Systems, vol. 4, no. 1, Article 7, 2012, pp. 1-24.

[4] The Clemson University Vechular Electronics Laboratory, "Automotive Electronics," http://www.cvel.clemson.edu/auto/auto_buses01.html, [retrieved: April, 2017].

[5] K. Parnell, "Put the Right Bus in Your Car," Xcell Journal, Available: http://www.rpi.edu/dept/ecse/mps/xc_autobus48(CAN).pdf, [retrieved: April, 2017].

[6] D. K. Nilsson, P. H. Phung, and U. E. Larson, "Vehicle ECU Classification Based on Safety-Security Characteristics," in Proc. the 13th International Conference on Road Tarnsport Information and Control (RTIC'08), Manchester, England, UK, 2008, pp. 1-7.

[7] CCS, "Electronic Control Units (ECUS)," 2014, http://www.ccs-labs.org/teaching/c2x/2014s/05-ecus.pdf, pp. 1-27, [retrieved: April, 2017].

[8] STW, "Control System Electronics," 2011, http://www.stw-technic.com/wp-content/uploads/2011/05/controllers.pdf, pp. 1-19, [retrieved: April, 2017].

[9] ETAS GmbH, "Electronic Control Unit (ECU) – Basics of Automative ECU," 2014, http://www.scribd.com/doc/268828296/20140121-ETAS-Webinar-ECU-Basics#scribd, pp. 1-30, [retrieved: April, 2017].

[10] Freescale, "Future advances in Body Electronics" https://cache.freescale.com/files/automotive/doc/white_paper/BODYDELECTRWP.pdf, 2013, pp. 1-18, [retrieved: April, 2017].

[11] T. Hoppe, S. Kiltz, and J. Dittmann, "Automotive IT-Security as a Challenge: Basic Attacks from the Black Box Perspective on the Example of Privacy Threats," Computer Safety, Reliability, and Security, 2009, pp. 145-158.

[12] M. Wolf, A. Weimerskirch, and C. Paar, "Security in Automotive Bus Systems," in Proc. the 2nd Embedded Security in Cars Workshop (ESCAR 2004), Bochum, Germany, 2004, pp. 11-12.

[13] Y. Pant, M. Pajic, R. Mangharam, "AUTOPLUG: An Architecture for Remote Electronic Controller Unit Diagnostics in Automotive Systems," Technical Report, Department of Electrical and Systems Engineering, University of Pennsylvania, 2012, pp. 1-13.

[14] L. Ferhatović, A. Lipjankić, A. Handžić, and N. Nosović, "System for Remote Diagnostic of Vehicle Defects," in Proc. The 17th Telecommunications Forum (TELFOR 2009), Serbia, Belgrade, 2009, pp. 1323-1326.

[15] M. Johanson, P. Dahle, and A. Söderberg, "Remote Vehicle Diagnostics over the Internet using the DoIP Protocol," in Proc. The Sixth International Conference on Systems and Networks Communications (ICSNC 2011), Barcelona, Spain, 2011, pp. 226-231.

[16] D. Oka, T. Furue, S. Bayer, and C. Vuillaume, "Analysis of Performing Secure Remote Vehicle Diagnostics," in Proc. Computer Security Symposium (CSS 2014), 2014, pp. 643–650.

[17] A. Mishra, A. K. Jhapate, and P. Kumar, "Improved Genetics Feedback Algorithm Based Network Security Policy Framework," in Proc. The 2nd International Conference on Future Networks, Sanya, China, 2010, pp. 8-10.

[18] N. Ben Youssef and A. Bouhoula, "Systematic Deployment of Network Security Policy in Centralized and Distributed Firewalls", in Proc. The 2011 IEEE International Conference on Privacy, Security, Risk, and Trust, Boston, Massachusetts, USA, 2011, pp. 1214-1219.

[19] C. Tang and S. Yu, "A Verification Algorithm of Network Security Policy Repository," in Proc. The 2009 International Conference on Information Technology and Computer Science, Kiev, Ukraine, 2009, pp. 297-300.

[20] X. Wang, W. Shi, Y. Xiang, and J. Li, "Efficient Network Security Enforcement with Policy Space Analysis", IEEE/ACM Transcaction on Networking, vol. PP, issue 99, 2015, pp. 1-13.

[21] D. Chemyavskiy and N. Miloslavskaya, "A Concept of Unification of Network Security Policies," in Proc. The Fifth International Conference on Security of Information and Networks, Jaipur, India, 2012, pp. 27-32.

[22] T. Bourdier and H. Cirstea, "Symbolic Analysis of Network Security Policies Using Rewrite Rules," in Proc. Symposium on Principles and Practicies of Declarative Programing (PDPP'11), Odense, Denmark, 2011, pp. 77-88.

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-562-3