

Wireless Authentication Solution and TTCN-3 based Test Framework for ISO-15118 Wireless V2G Communication

Zoltán Jakó, *Member, IEEE*, Ádám Knapp, *Member, IEEE* and Nadim El Sayed

Abstract— Vehicle to grid (V2G) communication for electric vehicles and their charging points is already well established by the ISO 15118 standard. The standard allows vehicles to communicate with the charging station using the power cable, i.e. a wired link, but it is improved to enable wireless (WLAN) links as well. This paper aims to provide an implementation that accomplishes a wireless authentication solution (WAS). With that the electric vehicles can establish V2G connection when approaching the charging pool, then identify and authenticate the driver and/or the vehicle. Furthermore, the paper presents a TTCN-3 based validation and verification (V&V) framework in order to test the conformance of the prototype implementation against the standard.

Index Terms—Vehicle-to-Grid, ISO 15118, wireless charging, Electric Vehicle, ITS, TTCN-3

I. INTRODUCTION

The proportion of Battery Electric Vehicles (BEV) and Plug-In Hybrid Electric Vehicles (PHEV), against conventional vehicles with internal combustion engine, is growing remarkably in developed countries. Led by the USA, the European Union and Japan the BEV and PHEV market is rapidly growing [1]. To serve this increased demand, massive charge point deployment is required. Nevertheless, due to business issues (e.g. billing) and grid limitations, smart charging is also a mandatory requirement to overcome the issues caused by mass electric vehicle (EV) recharging. For the sake of convenience hereafter the collection term EV for both battery electric vehicles and PHEVs is used.

The communication between EVs is an extensively researched topic and it is becoming an essential part of the C-ITS (Cooperative Intelligent transportation system) environment. The bi-directional communication between the vehicle and the charging point (and the grid infrastructure behind it) is referred to as vehicle-to-grid (V2G), thus V2G provides a communication interface for bi-directional charging (or discharging) of EVs. The EV charging station is the so-called EVSE (Electric Vehicle Supply Equipment). Inside the EV

This work is a part of the project NeMo - Hyper-Network for electro-Mobility that received funding from the European Union Horizon 2020 research & innovation program under grant agreement no 713794. Content reflects only the authors' view and European Commission is not responsible for any use that may be made of the information it contains.

Zoltán Jakó is with the Broadbit Hungary Kft., 1023, Ürömi utca 40, Budapest, Hungary (e-mail: zoltan.jako@broadbit.net).

Ádám Knapp is with the Broadbit Hungary Kft., 1023, Ürömi utca 40, Budapest, Hungary (e-mail: adam.knapp@broadbit.net).

Nadim El Sayed is with the DAI-Labor, Technische Universität Berlin, Berlin (TUB), Berlin, Germany (e-mail: nadim.elsayed@dai-labor.de).

there is a module responsible for the V2G communication. This module is referred to as Electric Vehicle Communication Controller (EVCC), while in the case of EVSE the literature uses the term Supply Equipment Communication Controller (SECC). The EV is capable of communicating with the charging point using its EVCC. The message exchange between the EV and the EVSE is standardized by ISO/IEC (International Organization for Standardization/ International Electrotechnical Commission) in the series of 15118 (e.g. [2] – [7]). As the communication parts of this generic equipment are the EVCC and SECC, ISO 15118 describes the communication between these components. ISO 15118 is the enabler of vehicle-to-grid applications.

The main challenge of any standardized technology is conformance and interoperability. Conformance testing checks a specific product (or maybe a part of a product) for compliance to requirements given in a base standard. A definition of interoperability testing is the "ability" of two or more systems (or components) to exchange and use information and execute successful procedures/sessions. The aim of interoperability testing is not restricted to demonstrating that products (from different manufacturers) can work together: it also shows that these products can work together using a specific protocol. Multi-vendor compatibility is crucial for the success of V2G technology.

The contribution of this manuscript is given as follows:

1. Introduce a prototype SECC implementation, which uses wireless (WLAN-based) communication to handle a V2G session with the EVCC. A wireless authentication solution (WAS) is presented that allows and handles the V2G communication and the identification of the EV via wireless links.
2. Provide a validation and verification (V&V) tool to test the V2G conformance of the implemented prototype against the base standard given in [3].

It is important to highlight the fact that V2G was originally planned to be used in a wired manner (i.e. using the charging cable with power line communication). However, wireless communication recently gained higher attention, even in the standardization process [8], [9]. Wireless communication between EV and EVSE is based on WLAN (802.11n or Wi-Fi). Hereafter the term wireless link is used, noting that it actually denotes WLAN in the context of this manuscript. To be more precise, the ISO 15118 foresees the option of wireless authentication especially in the draft of its sixth part [7] (more details are given in Section II.B). The wireless interface allows

Wireless Authentication Solution and TTCN-3 based Test Framework for ISO-15118 Wireless V2G Communication

the EV driver to start the V2G communication (and/or use optional value-added services) before parking. If the charge point is reserved, then the EV driver may be notified via wireless interface before parking. With wired communication this is only possible after parking and plugging the EV.

The conformance testing framework is based on script language used for testing purposes, the so-called TTCN-3 (Testing and Test Control Notation version 3) [10]. V2G has massive literature background related to security issues and performance tests. However, the conformance testing of the V2G protocol itself is less discussed. On the other hand, this is also a relevant issue, which enables the spreading of V2G technology worldwide.

A. Related Works

The first significant V2G related test paper was presented by Project eNterop [11]. They had created a conformance testing setup that is for black box testing of connected Systems Under Test (SUT) [12]. They define conformance tests, which can be fully automated. Furthermore they applied TTCN-3 scripts and later this test setup was used in ISO 15118-4 [5]. Shin *et al.*, in [13] provides a test system for EVSE in accordance with relevant standards, including ISO-15118-2,3 ([3], [4]) IEC-61851, IEC 61850-90-8 and HPGP (HomePlug Green PHY – Power line communication).

Compared to these related works, our conformance testing framework differs in two aspects. First, our conformance testing framework is using Ericsson's Titan TTCN-3 complier [14], which is now open source. Therefore, there is no need to buy expensive software to compile TTCN-3 scripts. Secondly, in this manuscript the focus is on the wireless (WLAN based) communication between the tested system and the conformance test tool. This is a completely new paradigm, therefore the standardization process has just began [6], [7].

The manuscript is organized as follows. Section II gives a brief introduction to the series of ISO 15118. Section III introduces

the proposed WAS, meanwhile Section IV presents the conformance testing framework. Finally, Section V gives concluding remarks and concludes the paper.

II. STANDARDS OF ISO 15118

The series of ISO 15118 standard currently contains nine parts. Each part is responsible for a small piece of the field of V2G. In this section, a brief overview of this standard family is given. ISO 15118-1 has the title „General information and use-case definition“. This document collects the use cases and overall goals of the standard itself [2].

The second part [3] is the most important from all for us, since it defines the technical specifications of all application layer messages and their respective parameters exchanged between the EV and the EVSE.

The (wired) physical and data link layer requirements are given in ISO-15118-3 [4]. Power line communication as defined in the HomePlug Green PHY specification is applied to encode digital signals onto the Control Pilot (CP) pin, which is part of the charging cable. These layers establish the Higher-Level Communication (HLC) outlined in ISO 15118-2. This third part also concerns the interaction with another standard called IEC 61851. This specifies analogue signals that encode the available amperage at a charging station. ISO 15118 builds upon this analogue and mainly safety-related IEC standard and enhances the charging process with digital higher-level communication. Part 4 [5] is also important from the perspective of this manuscript. This part contains the conformance tests (TTCN-3 scripts) for the requirements specified in ISO 15118-2. Note that part 4 also contains lower layer test cases related to the wired link that are not considered in the present prototype system.

Part 5 is currently under preparation. When it is finalized, it will contain the conformance tests for the physical interface and its requirements defined in ISO 15118-3 [4].

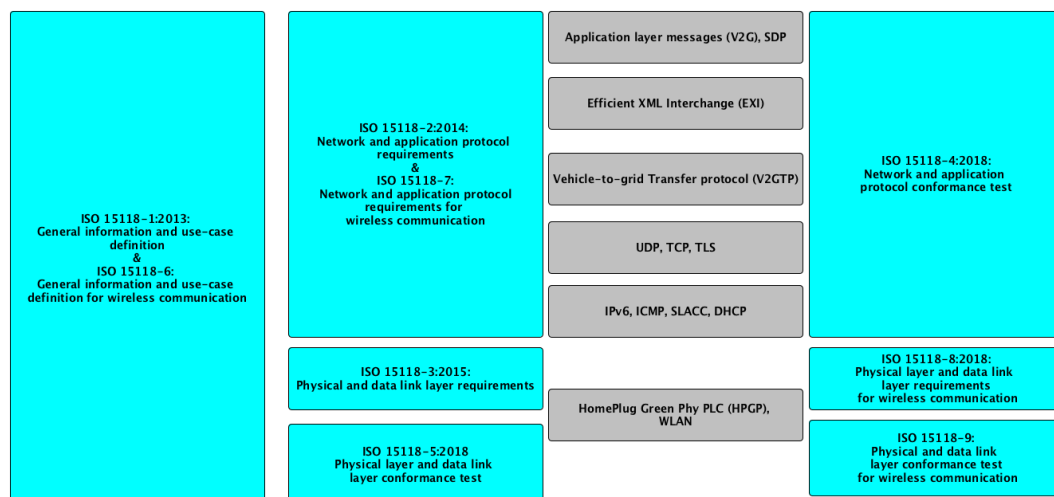


Fig. 1: Relationship between ISO 15118 parts

ISO 15118-6 [7] collects the general information and use-case definition for wireless communication, similarly to ISO 15118-1. It is foreseen that Part 1 and 6 will be merged into one document in the near future and both wired and wireless communication use cases will be available in the next version of ISO 15118-1.

The network and application protocol requirements for wireless communication will be presented in Part 7, however this document does not yet exist. It is also foreseen that Part 2 and 7 will be merged into one document.

Part 8 [6] is similar to Part 3, the big difference between them is that it contains the physical layer and data link layer requirements for wireless communication. The first version of 15118-8 was published in the first quarter of 2018.

Finally, Part 9 shall contain the conformance tests for wireless charging. On the other hand, Part 9 is under development, therefore, there is no document available yet.

An overview of ISO 15118 and the relationship between the parts are illustrated in Fig. 1.

A. V2G protocol stack (PLC)

The protocol stack of the V2G is presented in this subsection. The whole protocol stack is lavishly detailed in ISO-15118-2 [3], therefore we just give a brief presentation in this subsection. After the plug of the EVSE is connected to the EV an IPv6 address is assigned to the EV (the physical- and MAC layer link is established). The IPv6 address is assigned to the EV by DHCPv6 (Dynamic Host Configuration Protocol) and Stateless auto-configuration (SLAAC). Note that SLAAC is mandatory, but DHCPv6 is optional according to the standard. Subsequently, the EV shall send a SECC Discovery Request message as UDP multicast over IPv6. The SECC receives and replies to the request with a response message containing the link-local IPv6 address of the EVSE. This message exchange is the so-called SECC Discovery Protocol (SDP). Afterwards the HLC can start. HLC is the bidirectional digital communication that uses the protocol and messages specified in ISO 15118-2 and ISO 15118-3 (or 15118-7). HLC includes the Protocol Handshake using the Vehicle to Grid Transfer Protocol (V2GTP), over the Efficient XML Interchange (EXI) format, and the V2G messages (e.g. Session Setup Request message). HLC allows, among other things, to negotiate the charging parameters and to authenticate and authorize the EV and the user, utilizing more secure cryptographic certificates in the plug and charge case.

The EV-EVSE can send or receive V2G application layer messages. On top, the possible message set is selected based on the usage. There are common V2G application layer messages and there are some sets related to the charging type (e.g. AC, DC or inductive charging, etc.). The V2G messages are described in the format of XML (Extensible Markup Language). A plain XML message contains significant overhead and unnecessary information (unnecessary regarding the EVCC or SECC part). Therefore, to reduce the size of the XML message it is encoded into EXI format. The resulting data is encapsulated into the V2GTP, which is encrypted using the TLS protocol, and transmitted using the general TCP/IP protocol suite to the EV or EVSE. The standard defines a couple of possible data links and physical layers as well [3], [6].

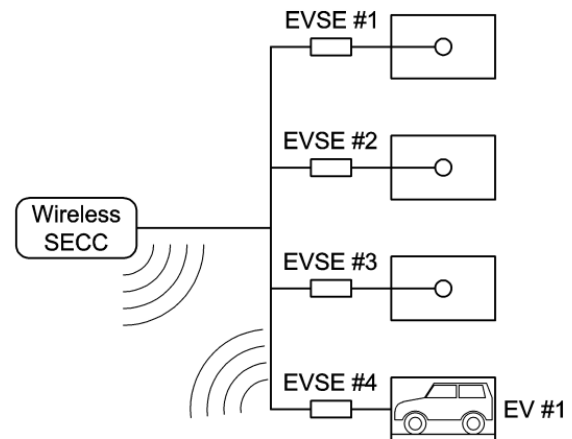


Fig. 2: Wireless communication between SECC and EV(s) as described in ISO 15118-6 [7]

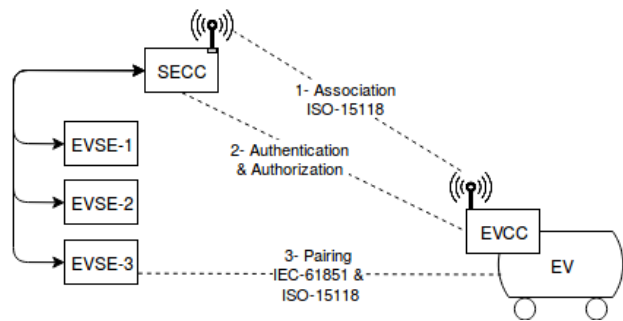


Fig. 3: Association and Pairing sequence

B. Wireless V2G

Unlike the PLC case, which may apply only to conductive charging, the wireless communication allows the support of more use cases such as the static inductive charging or the dynamic Wireless Power Transfer (WPT). The wireless parts of the ISO 15118 (parts 6, 7 and 8) base their considerations on three entities already defined in the ISO 15118-1. These are the following: EVSE, EVCC and SECC.

Unlike in the wired case (plug and charge), where the communication is rather point-to-point, the wireless communication is point to multipoint, which creates several challenges for the communication integrity, confidentiality and authenticity [15]. Thus, the ISO 15118 foresees an additional pairing mechanism to make sure that the EV, which is (wirelessly) communicating to the EVSE is in fact the exact one plugged at the Charge point or driving over the coil in case of WPT.

The main difference between the wireless V2G and the PLC V2G is that in the PLC case, the communication starts when the car is plugged, and the communication partner (EVCC and SECC) are unambiguously identifiable. Furthermore, the SECC knows exactly at which EVSE the EV is plugged. Where as in the wireless V2G, this is not true, and the wireless V2G protocol needs to define the necessary means to ensure unambiguity, confidentiality, mutual integrity and authenticity.

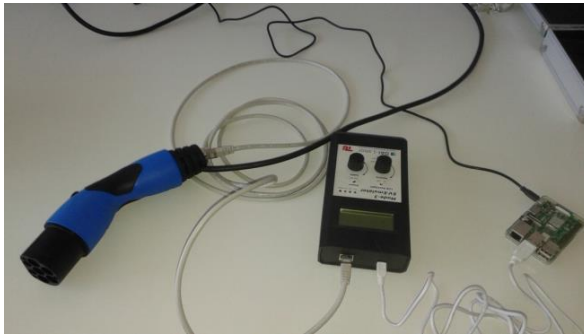


Fig. 4: Implemented EVCC (left) and EVSE (right)

The wireless communication between the EV and the EVSE is depicted in Fig. 2 and in Fig. 3. Each EVSE can be connected to one SECC only and the EVCC is able to communicate with the SECC over the wireless link. Furthermore, the association is defined as the process of establishment of wireless communication between SECC and EVCC. The Discovery is the phase in which EV obtains a list of available SECCs in its wireless communication range. This is handled by the SDP, similarly to wire environment. On the other hand, pairing is the process by which a vehicle is correlated with the unique EVSE at which it is located and from which the power will be transferred either through a cable or through wireless technology.

Pairing is done after the SDP and association phases. EVCC asks SECC the authorization to start a Pairing sequence. After a positive answer from SECC Pairing starts. EV starts the sequence of B-State, C-State, B-State toggle (referring to the different vehicle states from IEC 61851). The EVSE that detects the sequence of toggles informs SECC of the pairing toggles detection. SECC informs EVCC of the correct toggles reception. Depending on whether the location detected is convenient or not, SECC may decide to ask EVCC to change location. The implementation details are described in [16]. Once SECC Discovery, Association, and Pairing are done, the V2G application layer communication (i.e. HLC) can start. Note that this approach does not depend on the high precision localization (e.g. GPS) to determine the proximity of the EV to a certain EVSE. The following Section describes the prototypical implementation of our wireless authentication solution.

III. PROPOSED WIRELESS AUTHENTICATION SOLUTION (WAS)

This prototype implements the wireless communication for the conductive charging case, yet most of the components are applicable to inductive charging. This is especially true for the EVCC – SECC communication (SDP and Association), and the high level communication (V2G application layer message exchange). Merely the EVSE and the EV parts have to be adapted to implement the correspondent standards for wireless power transfer, which affects the pairing (and fine positioning) part of the implementation. Our implementation consists of three entities: EVCC, SECC and EVSE. Each of these entities is composed in its turn of different components that provide different functionalities. These components can interact through interfaces. The functionalities of the prototype

implementation of the ISO 15118 based wireless authentication cover all the layers of the ISO/OSI stack. It is important to highlight the fact that this implementation does not focus on the charge process (energy flow) itself, only on the communication (V2G) part. Therefore, there is no need to implement all the V2G message set (e.g. messages responsible for metering data exchange).

A. EVCC

The EVCC implementation refers to the conductive charging that uses the IEC 62196 Type-2 Connector Plug (illustrated in Fig. 4). The EVCC consists of the following components: EV-Emulator and EV-Controller.

The *EV-Emulator* is built up by off the shelf microcontroller (Atmel ATMEGA16p) and circuit elements for implementing the IEC 61851 functionalities necessary for the pairing, which is achieved by doing some toggling pattern on the wire, based on ISO 15118-3 [4]. Furthermore, the EV-Emulator implements a UART interface to the EV-Controller. Meanwhile the *EV-Controller* is a Raspberry Pi 3 device equipped with a WLAN module (802.11n) for discovering the different SECCs in the neighbourhood and reading out their Vendor Specific Elements (VSE) on the ISO-Layer 2 containing their EVSEID according to ISO 15118-8 [6]. The EV-controller associates with a SECC using IPv6 Stateless auto-configuration and implements a SECC Discovery Protocol (SDP) Client to get the SECC settings and endpoint parameters over UDP-Multicast. Using these parameters, the EV-controller performs a TLS handshake with the SECC by verifying the Root-V2G

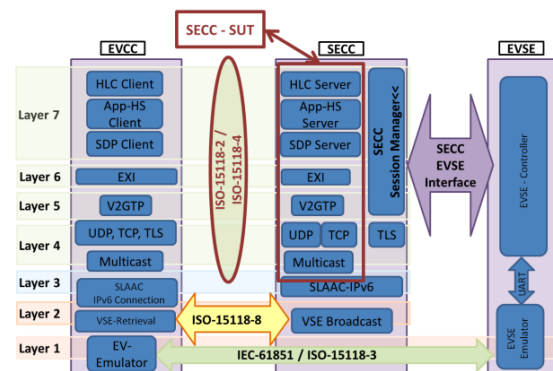


Fig. 5: WAS and SECC SUT

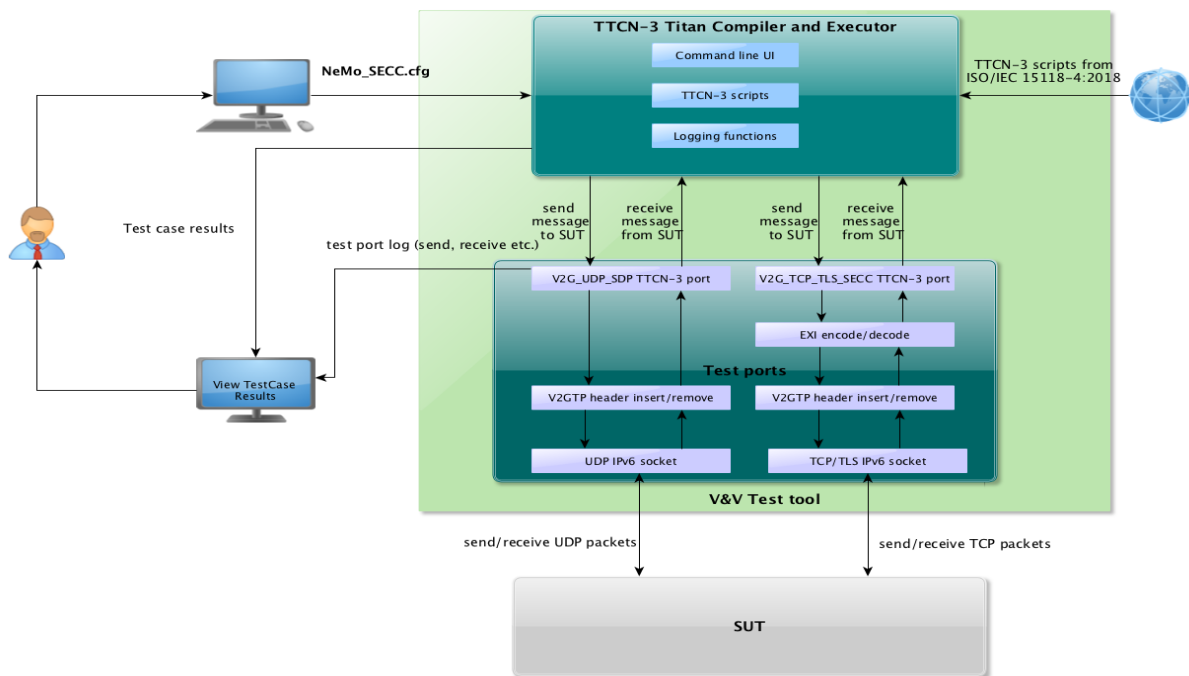


Fig. 6: The architecture of the Validation and Verification (V&V) test tool including the SUT

certificate, then its state machines start the HLC-Communication by implementing the client side of the ISO 15118-2 [3]. This is an EXI encoded communication, encapsulated in the Vehicle to Grid Transfer Protocol. When in pairing state, the EV-Controller sends the toggles through the EV-Emulator over the UART interface.

B. EVSE

The EVSE consists of the following components: EVSE-Emulator and Charge Point Manager (CPM).

The EVSE-Emulator is an OpenEVSE hardware [17][15], connected to an IEC 62196 Socket (see Fig. 4). Its firmware allows us to control and access the lower level functionalities of the EVSE according to the IEC 61851, which makes it the counter part of the EV-Emulator for reading the toggling on the wire. It implements a UART interface to the Charge Point Manager. The Charge Point Manager (CPM) is a Raspberry Pi 3 device connected with the EVSE-Emulator over USB, and implements the higher level messages of the pairing process. The CPM reads the toggles on the wire from the EVSE-Emulator over the UART, and communicates them to the SECC, that in his turn tracks which EVSE has detected the respective EV.

C. SECC

With regard to wireless V2G communication, the SECC sub component is the most important part of our implementation. The SECC consists of the following components: SECC-Controller and Charging Session Manager (CSM). Again, SECC-Controller is a Raspberry Pi 3 device equipped with a WLAN module, which operates in the Access Point (AP) Mode and spans an Automatic Wireless Charging (AWC) network, and includes its EVSEID in its VSE. It listens on the Multicast

group and implements the SDP Server. It also implements the Server side of the HLC Communication. In coordination with the Charging Session Manager (CSM) denotes the central logical component of the SECC where the other components are linked together. It tracks the different charging sessions of the different EVs and their respective EVSE, including their state. It interfaces with the SECC-Controller for the EV communication and with the EVSE-Controller for the charge point information retrieval and control.

IV. TESTING WIRELESS AUTHENTICATION SOLUTION WITH TTCN-3

In this section, a V&V testing framework is introduced for V2G communication. As mentioned in Section II regardless of the physical layer, the network and application requirements are common for any ISO 15118-based implementation. This is why the focus of the testing framework developed here is mainly based on the standard ISO 15118-2 (more precisely ISO/IEC 15118-2-ED2) and its respective conformance tests specified in the ISO/IEC 15118-4 [5]. This conformance testing framework is applied on the SECC part of the WAS presented in Section III.

A. Test bed

Conformance tests specify the testing of capabilities and behaviours of a System Under Test (SUT), as well as check what is observed against the conformance requirements specified in ISO/IEC 15118-2 [3] and against what the supplier states the SUT implementation's capabilities are. In this case, SUT is a software that implements SECC (highlighted with red box in Fig. 5). From the protocol point of view, the *client-server* model is used, where the EVCC takes the role of the *client* of the protocol, initiating the communications, and the SECC

takes the role of the *server*. The EVSE-SECC interface is out of the scope of the standardization and is handled in the WAS by the SECC-Session-Manager. The SECC SUT has a dummy SECC-Session-Manager, which does not rely on an actual EVSE. This allows us to focus on the relevant part for conformance tests, without relying on an EVSE and an EV to run the tests while validating and verifying major parts and aspects of the implementation. The conformance test cases are described leveraging this test architecture and are specified in TTCN-3 Core Language for ISO/OSI Network Layer (Layer 3) and above. Note that underlying protocols, such as UDP, TCP/TLS, etc., are not tested directly during the conformance tests; however, the test framework relies on them.

Nowadays TTCN-3 [10] is widely used as a testing language for standards in telecommunications, and it is even used in ITS. TTCN-3 is mostly applied for protocol testing but other test areas (software, system, etc.) and verification objectives (interoperability, robustness, etc.) are starting to use it. In this paper, TTCN-3 is used for the test cases implementation of protocols of ISO/IEC 15118-2-ED2.

B. Test System Implementation

The presented V&V test tool consists of several applications, configuration files, test scripts and run-time environment that run on an embedded computer. Test scenarios and cases are described in TTCN-3 language that is compiled into a binary program, the so-called ETS (executable test suite).

The TTCN-3 scripts are obtained from ISO/IEC 15118-4:2018 [5]. However, some parts are modified since these test scripts are written for wired case, not for wireless (e.g. absence of PLC) and ISO/IEC 15118-9 currently does not exist. These scripts are written in a specific script language (TTCN-3). The V&V test tool contains two main components. These are given as follows: TTCN-3 compiler/executor and Test ports (TPs).

These components and their subcomponents are depicted in Fig. 6. The used TTCN-3 compiler and executor is the open source Titan TTCN-3 compiler developed by Ericsson [14]. Titan is a TTCN-3 compilation and execution environment with an Eclipse-based IDE. The Test executor requires a configuration file (cfg). This file contains the input parameters (e.g. the group of test cases that should be executed, use TLS or not etc.). The Titan compiler builds an executable (binary) test suite (ETS) from the TTCN-3 scripts, the test port code and the Titan runtime library. Note that it is not mandatory for ETS to be executable. Titan allows very flexible runtime parameterization of the test cases (e.g. IP addresses, port numbers etc.). The values of runtime parameters need not to be defined at development time, however, default values can be specified, but they can be provided just before the test execution session. In this way, flexible execution scenarios can be created without re-building the ETS.

The TTCN-3 code is generic, therefore the interfaces between the tester and the tested entity (i.e. SUT) are specified at the level of the exchanged abstract data messages and signals. Setting up and maintaining the transport connections and sending/receiving "real" messages and signals are the tasks of interface adaptors. Adaptors are called test ports (TPs) and are plugins written in C/C++ (as illustrated in Fig. 6).

Note that ETS can run on a traditional PC or on a mini/embedded PC, only a Linux environment is required, with

the package of OpenSSL (in case of TLS). The computational capacity is usually not a bottleneck for such TTCN-3 based black box testing.

After executing the ETS (with the proper configuration file) the results of test cases are visible for the user by parsing the log file manually or via a graphical interface. The ETS is responsible for assembling the test packets, which are then injected into the network, and transmitted to the SUT that is the SECC in this case. Based on the response from the SUT or even on the existence of the response taking into account time restrictions as well, verdict is made and presented to the test engineer via a suitable, graphical user interface.

The test ports should take care of the following. In the case of SDP, it shall insert a V2GTP header to the SDP message; remove and process the V2GTP header from received message and send/receive UDP multicast message to/from SUT over IPv6.

In the case of V2G application layer message exchange the test port shall insert a V2GTP header to the V2G message; remove and process the V2GTP header from received message; encode/decode message with EXI, encrypt/decrypt V2G message (if TLS is enabled) and send/receive TCP message to/from SUT over IPv6.

C. Test Configuration

The main parameters of the test configuration – used by the test cases – are summarized in Table I.

D. Test Cases and Validation

Black box testing is used in this manuscript to test the SECC implementation in the WAS. This method of testing examines the behaviour of a SUT without considering the internal implementation and structure of the SUT, thus relying on the SUT's open interface for testing. The test tool acts as an EVCC and sends SDP/V2G requests to SECC. The SECC shall respond to them in time.

TABLE I
A COLLECTION OF CONFIGURATION PARAMETERS USED BY TEST CASES

Parameter	Value	Description
LogFile	NeMo-TUB-BIT-%n.log	The filename and path of the log file.
LogSourceInfo	yes	The tool should log the source information also.
PIXIT_SECC_CMN_TLS	false	Use TLS in the V2G communication.
PICS_CMN_CMN_V2gtpSdp	true	Test the SUT with V2GTP-SDP test case set.
PICS_CMN_CMN_Sdp	true	Test the SUT with SDP test case set.
PICS_CMN_CMN_SupportedAppProtocol	true	Test the SUT with V2G SupportedAppProtocol test case set.
PICS_CMN_CMN_SessionSetup	true	Test the SUT with V2G SessionSetup test case set.
pt_V2G_UDP_SDP_Port.debugging	yes	The log should contain UDP test port debug messages.
pt_V2G_UDP_SDP_Port.multicastAddress	"ff02::1"	IPv6/UDP multicast address used by SDP request.
pt_V2G_UDP_SDP_Port.multicastPort	15118	IPv6/UDP multicast port used by SDP request.
pt_V2G_UDP_SDP_Port.ifind	"eth0"	The index of the network interface.

TABLE II
DEMONSTRATION OF THE TEST CASES

Test Case identifier	Test objective	Expected behavior of SUT
TC_SECC_V2GTPSDP_001	The V&V test tool sends a “ <i>SECCDiscoveryReq</i> ” message with the V2GTP header information “protocolVersion” equals ‘0x01’H, ‘invProtocolVersion’ equals ‘FE’H and ‘payloadType’ equals ‘0x8001’H. (V2GTP Header is matched for V2G message content).	Test System then checks that the SUT sends a “ <i>SECCDiscoveryRes</i> ” message with the V2GTP header, information ‘protocolVersion’ equals ‘0x01’H, ‘invProtocolVersion’ equals ‘FE’H and ‘payloadType’ equals ‘0x8001’H.
TC_SECC_SDP_001	The V&V test tool sends a “ <i>SECCDiscoveryReq</i> ” message with ‘Security’ equals ‘0x10’H and ‘TransportProtocol’ equals to ‘0x00’H.	V&V test tool then checks that the SUT sends an “ <i>SECCDiscoveryRes</i> ” message with ‘Security’ equals ‘0x10’H, ‘TransportProtocol’ equals ‘0x00’H and a valid port and IP address.
TC_SECC_V2G_001	The V&V test tool sends a “ <i>SupportedAppProtocolReq</i> ” message with a list of valid AppProtocols including ISO namespace and all additional mandatory parameters.	The V&V test tool then checks that the SUT sends a “ <i>SupportedAppProtocolRes</i> ” message with response code ‘OK_SuccessfulNegotiation’ or ‘OK_SuccessfulNegotiationWithMinorDeviation’ and all additional mandatory parameters.

The SDP protocol uses UDP (on a fixed port 15118) for communication, meanwhile V2G uses TCP/TLS dynamic ports between the ranges of 49152 – 65535.

Since the SECC implementation of WAS does not cover all the V2G message set given in [3], and due to page limitations only three test cases are shown here.

In order to demonstrate the capabilities of the V&V test tool we choose three test cases, with each of them belonging to a dedicated protocol (i.e. V2GTP, SDP and V2G application layer message exchange). The objective of the test cases and the expected behaviour is collected in Table II.

1) V2GTP – V2G Transfer protocol

V2GTP is responsible for the encapsulation of an SDP Discovery Request or any V2G application layer message. The V2GTP message consists of two parts, the header and the payload. The payload contains the pure SDP or V2G message, meanwhile the V2GTP header contains information about the protocol version, the payload type and size.

2) SDP – SECC Discovery protocol

The SDP protocol is responsible for the SECC discovery and the negotiation of the transport protocol (i.e. to encrypt the transport layer messages). In this test the V&V tool sends the SDP request and the SUT shall respond with a valid and adequate SDP response. See Table II for further details.

3) V2G – SupportedApplication message exchange

The first V2G application layer message is entitled as “Supported Application Request”, which is also a negotiation message between the SECC and the EVCC in order to decide the V2G protocol version and other parameters. This message is an XML message encoded in EXI format. In this test the V&V tool sends the V2G “SupportedAppProtocolReq” request and the SUT shall respond with a valid and adequate response. See Table II for further details.

E. Test evaluation and results

The results of the tests are presented in this subsection. During the tests, requests were sent to the SUT and the corresponding responses were investigated. The time constraints of the V2G protocol is also taken into account by the test tool. The captured message structure of the given protocol (i.e. SDP, V2G) is illustrated by the Titan’s Eclipse IDE log viewer and with Wireshark packet sniffer. From Fig. 7, one can see the V2GTP

header. The most important part of the header is the field of protocol type with the value of ‘9000’H. This value denotes that the payload contains a SECC Discovery request message. Note that the payload size is two bytes.

This message is sent as an UDP multicast message to the IPv6 address of “ff02::1” on the port of 15118 by the V&V test tool (as depicted in Fig. 7). The SUT will receive the UDP multicast message and answer it with a dedicated (i.e. non-multicast) message.

Note that the payload part shall contain the applied transfer protocol ID and the security layer ID related to SDP. In this case, a simple TCP connection was used without encryption

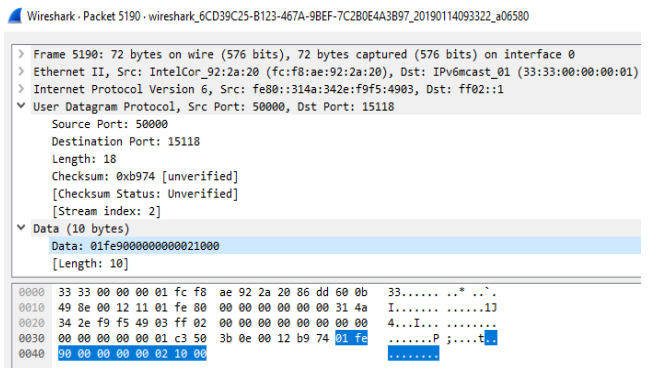


Fig. 7: Sent SECC Discovery Request message

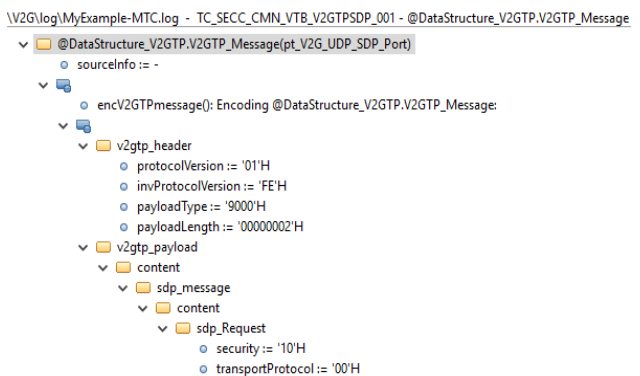


Fig. 8: Structure of the SECC Discovery Request message

```

V2G.log\MyExample-MTC.log - TC_SECC_CMN_VTB_V2GTPSDP_001 - @DataStructure_V2GTP.V2GTP_Mes
  ✓ @DataStructure_V2GTP.V2GTP_Message (pt_V2G_UDP_SDP_Port)
    ✓ sourceInfo := -
    ✓ @DataStructure_V2GTP.V2GTP_Message :
      ✓ v2gtp_header
        ✓ protocolVersion := '01'H
        ✓ invProtocolVersion := 'FE'H
        ✓ payloadType := '9001'H
        ✓ payloadLength := '00000014'H
      ✓ v2gtp_payload
        ✓ content
          ✓ sdp_message
            ✓ content
              ✓ sdp_Response
                ✓ secc_IPaddress := 'FE8000000000000062334BFFFE207941'H
                ✓ secc_Port := 55534
                ✓ security := '10'H
                ✓ transportProtocol := '00'H
            ✓ id 1

```

```
Wireshark - Packet 5284 - wireshark_6CD39C25-B123-467A-9BEF-7C2B0E4A3B97-20190114093322_a06580
```

> Frame 5284: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface 0

> Ethernet II, Src: Apple20:79:41 (08:01:35:2b:20:79:41), Dst: IntelGigE:92:2a:10 (f8:ae:92:a:20)

> Internet Protocol Version 6, Src: fe80::6233:4bff:fe20:7941, Dst: fe80::314a:342e:f9f5:49b3

✚ User Datagram Protocol, Src Port: 15118, Dst Port: 50000

Source Port: 15118

Destination Port: 50000

Length: 36

Checksum: 0x9726 [unverified]

[Checksum Status: Unverified]

[Stream Index: 3]

✚ Data (20 bytes)

Data: 01fe900100000014fe8000000000000062334bfff207941...

[Length: 28]

0000	fc f8 ae 92 2a 20 60 33	4b 20 79 41 86 dd 60 0c' 3 k yA..'
0001	2a 23 00 24 11 40 fe 80	00 00 00 00 00 00 62 33	4..5.g.....b3
0002	4b ff fe 20 79 41 41 3b	00 00 00 00 00 00 31 4a	k..yA.....12
0020	34 2e f9 f5 49 03 3b 0e	c3 50 00 24 97 26 01 fe	4...I...P.S.8
0040	90 01 00 00 00 14 fe 80	00 00 00 00 00 62 33b3
0050	4b ff fe 20 79 41 d8 ee	10 0c 00 00 00 00 00 00	k...yA.....

The desired SDP response (from the SUT) is illustrated in Fig. 8 and in Fig. 9, respectively. The interesting part of the message is related to the V2GTP protocol. In the header, the payload type is '9001'H, which denotes that the payload is a SDP response message (corresponding with the standard [3]). The payload size is 28 bytes, since the SDP response (V2GTP payload part) contains the link-local address of the SUT and the dynamic port, where the next (V2G application layer) message shall be sent. Furthermore, it contains the same values of the field security and transport protocol that was sent in the SDP request. According to Fig. 9, the SDP process completed successfully, thus the V2G application layer message exchange begins. The V&V test tool first sends an EXI encoded message to the SUT containing the parameters depicted on Fig. 11. Afterwards it starts the timer and waits for the SUT's response. This request message contains protocol namespace and the versions supported by the EVCC (in our case the test tool). The SECC (the SUT in this case) should respond that the proposed protocol version is supported by the SECC or not. If it is supported, then the SUT shall send an adequate response, containing a response code. Otherwise, the SUT ends a failed response code, to inform the EVCC that the proposed version is not supported.

Fig. 11: V2G Supported Application Protocol request message

Fig. 12: V2G Supported Application Protocol response message

This message pair is illustrated in Fig. 11 and Fig. 12, respectively. From the incoming response, the V&V test tool first checks the V2GTP header. In this case, the payload type should be ‘8001’H. Thereafter, from the payload the V&V test tool is able to decode the EXI stream and collect all the necessary information. In this example, the protocol version offered by the EVCC (test tool) is supported by the SUT, thus the negotiation is successful and the response code is *‘OK_SuccessfulNegotiationWithMinorDeviation’*. The results of three test cases are collected in Table III. All the three test cases are evaluated through wireless link and all of them ended with the verdict “Pass”.

Test Case identifier	Verdict	Comments
TC_SECC_V2GTPSDP_001	Pass	V2GTP Header message was correct. (SDP Response Message).
TC_SECC_SDP_001	Pass	SDP Response message was correct.
TC_SECC_V2G_001	Pass	SupportedAppProtocolRes message was correct

V. CONCLUSION

In this paper a wireless authentication solution prototype has been presented, which allows electric vehicle owners to identify themselves nearby the charging station, but before connecting the plug to the EVs. Furthermore, we built a conformance test system for the SECC in accordance with the ISO/IEC 15118 standards. The conformance tests are evaluated with a TTCN-3 framework. The main advantage of the proposed V&V test tool is that it is configurable and extendable, therefore subsets of V2G message exchanges can be also executed, or new TTCN-3 test cases can be added next to the conventional ones.

In the manuscript, three test cases were introduced from the developed set for illustration purpose. From the designated tests, it is approved that SDP and V2G communication is possible via wireless links.

The possible future works include the followings. The Test tool covers the V2G messages given in the current version of ISO 15118-2. In the next version of this standard (expected at the end of 2019 or mid 2020) will have more V2G messages, which are related to bidirectional- and wireless charging. One possible extension of the test tool is to support those new message pairs. The current version of the V&V tool uses an Eclipse plugin. In the future, it is desired to have a dedicated graphical user interface (GUI). Another possible extension is to support other charging related protocols, next to V2G, like OCPP (Open Charge Point Protocol).

VI. REFERENCES

- [1] M. Mültin, "ISO 15118 as the Enabler of Vehicle-to-Grid Applications," 2018 International Conference of Electrical and Electronic Technologies for Automotive, Milan, 2018, pp. 1-6.
- [2] ISO 15118-1:2013: Road vehicles -- Vehicle-to-Grid Communication Interface -- Part 1: General information and use-case definition
- [3] ISO 15118-2:2014: Road vehicles -- Vehicle-to-Grid Communication Interface -- Part 2: Network and application protocol requirements
- [4] ISO 15118-3:2015: Road vehicles -- Vehicle to grid communication interface -- Part 3: Physical and data link layer requirements
- [5] ISO 15118-4:2018: Road vehicles -- Vehicle to grid communication interface -- Part 4: Network and application protocol conformance test
- [6] ISO 15118-8:2018: Road vehicles -- Vehicle to grid communication interface -- Part 8: Physical layer and data link layer requirements for wireless communication
- [7] ISO 15118-6:2015: Road vehicles -- Vehicle to grid communication interface -- Part 6: General information and use-case definition for wireless communication
- [8] O. Simon and D. Shkadarevich, "Application of V2G communication for wireless interoperable power transfer," 2017 Twelfth International Conference on Ecological Vehicles and Renewable Energies (EVER), Monte Carlo, 2017, pp. 1-5.
- [9] A. Krivchenkov and R. Saltanovs, "Analysis of wireless communications for V2G applications using WPT technology in energy transfer to mobile objects," 2015 56th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCon), Riga, 2015, pp. 1-4.
- [10] ITU-R Z.161 : Testing and Test Control Notation version 3: TTCN-3 core language, <https://www.itu.int/rec/T-REC-Z.161/>
- [11] K. Hänsch et al., "An ISO/IEC 15118 conformance testing system architecture," 2014 IEEE PES General Meeting | Conference & Exposition, National Harbor, MD, 2014, pp. 1-5.
- [12] S. Gröning, C. Lewandowski, J. Schmutzler and C. Wietfeld, "Interoperability Testing Based on TTCN-3 for V2G Communication Interfaces," 2012 International Conference on Connected Vehicles and Expo (ICCVE), Beijing, 2012, pp. 298-303.
- [13] Minh Shin, Hwimin Kim, Hyoseop Kim 1 and Hyuksoo Jang, "Building an Interoperability Test System for Electric Vehicle Chargers Based on ISO/IEC 15118 and IEC 61850 Standards", Applied Sciences Vol.6, No.6:165
- [14] Ericsson's Titan TTCN-3 compiler, <https://projects.eclipse.org/projects/tools.titan>
- [15] El Sayed, N, Lan L L. I. N., Goa, F., & Shi, X. "eCo-FEV: efficient Cooperative infrastructure for Fully Electric Vehicle." IEEE Transportation Electrification, April 2014.
- [16] N. El Sayed, "A Prototypical Implementation of an ISO 15118 Based Wireless Vehicle to Grid Communication for Authentication over Decoupled Technologies" 2019 International Conference of Electrical and Electronic Technologies for Automotive, Torino, 2019.
- [17] OpenEVSE tool kit, <https://www.openevse.com/kits.html>



Ádám Knapp received his M. Sc. degree in software engineering in 2011 from the Budapest University of Technology and Economics, Hungary. He is a member of IEEE. At the present, he is software developer at BroadBit Hungary Kft. and assistant research fellow at BUTE, Dept. of Networked Systems and Services. His main working area includes communication theory, 4/5G mobile networks and cooperative intelligent transportation systems.



Zoltán Jakó received his M.Sc. degree in electrical engineering, from Budapest University of Technology and Economics (BUTE), Budapest, Hungary, in 2011. He received the Ph.D. degree in the Department of Networked Systems and Services, BUTE at 2017. He is a member of IEEE. He is software developer at BroadBit Hungary Kft. and assistant research fellow at BUTE, Dept. of Networked Systems and Services since 2011. His research interests includes network design with stochastic geometry, next-generation heterogeneous network analysis, vehicle-to-vehicle (V2V) communication and vehicle-to-grid (V2G). He has been the involved several EU FP7 and Horizon2020 research projects.



Nadim El Sayed graduated with distinction from the Technische Universität Berlin (TUB) in Computer Engineering with specialization on Telecommunication Networks in 2010, and received a distinction award from the VDI in 2011. Since then he has been a researcher at the DAI-Labor, TUB in the Competence Center for Networks and Mobility. He was involved in many national and EU projects working in the areas of Mobile Communication, E-Mobility, Smart Grids, IoT and Industry 4.0. He has published scientific journal papers, conference articles and book chapters, and authored open source software for smart-metering and power quality monitoring. In 2018 he was an invited expert speaker at the German federal Ministry of economy regarding the directive 2014/94/EU2014 on the deployment of alternative fuels infrastructure.