FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Development of an Interoperable Communication between Electric Vehicles and Charging Stations Based on the ISO 15118 Standard

Frederico João Ayres Pereira da Cunha Ramos

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Justino Miguel Ferreira Rodrigues

Second Supervisor: David Emanuel da Silva Rua

July 23, 2025

Resumo

Veículos elétricos são um mercado emergente no contexto atual de soluções de mobilidade ambientalmente sustentáveis. Com a expansão da mobilidade elétrica, surgem vários desafios, como garantir que há potência suficiente no sistema elétrico para suportar a crescente infraestrutura de carregamento e a necessidade de padronizar os processos de carregamento dos veículos.

Com esses desafios em mente, esta dissertação visa contribuir para os estudos em curso sobre a infraestrutura de carregamento e as redes elétricas inteligentes, ao analisar as mais recentes normas criadas pela Organização Internacional de Normalização que abordam essas questões. Mais especificamente, será investigada a conexão entre os veículos elétricos e as suas estações de carregamento que, consequentemente, os ligam à rede elétrica, devido às capacidades de transferência de potência bidirecional introduzidas pela ISO 15118-20. Ao implementar um posto de carregamento baseado neste protocolo, o objetivo é desenvolver uma unidade de carregamento moderna, acessível e com funcionalidades bem integradas e robustas. Para além disso, o próprio protocolo será explorado em maior profundidade de forma a identificar potenciais áreas de melhoria, com especial foco no reforço da segurança das comunicações.

Keywords ISO 15118, ISO 15118-20, IEC 61851, EVSE, EVCC, HLC, PnC, V2G, BPT, TLS, análise, segurança no carregamento

Abstract

Electric vehicles are an emerging market and essential in today's environmentally sustainable transportation alternatives. As electric mobility expands several challenges arise, such as ensuring adequate power availability to support the growing infrastructure and the need to standardize electric vehicle charging processes.

With this in mind, this dissertation aims to contribute to the ongoing studies of charging infrastructure and smart grid nodes by studying the latest standards created by the International Organization for Standardization that address this issues. Specifically, we will investigate the connection between electric vehicles and their charging stations which, consequently, also link them to the electric grid, duo to the Bidirectional Power Transfer capabilities introduced by ISO 15118-20. By implementing a charging station solution based on this protocol, the objective is to develop an affordable, modern charging unit with well-integrated and robust features. Furthermore, the protocol itself will be explored in greater depth to identify potential areas for improvement, with a particular focus on enhancing communication security.

Keywords ISO 15118, ISO 15118-20, IEC 61851, EVSE, EVCC, HLC, PnC, V2G, BPT, TLS, analysis, charging security

Contribution to the Sustainable Development Goals (SDGs) of the United Nations

The alignment of this dissertation with the United Nations Sustainable Development Goals (SDGs) helps frame the societal and environmental relevance of the work developed. The table below identifies the goals and specific targets potentially impacted by the project, alongside its contributions and indicators for measuring progress.

Table 1: Alignment of the dissertation with the United Nations Sustainable Development Goals (SDGs).

SDG	Target	Contribution	Performance Indicators and Metrics
7	1	Smart charging capabilities (ISO 15118) enhance access to clean, modern energy for EVs.	Share of sessions using smart/renewable energy.
	a	Promotion of advanced energy systems via international research and tech adoption.	Technical collaborations or deployments involving ISO 15118-based smart grids.
9	4	Modernization of EVSE infrastructure with ISO 15118 enables automation and improves efficiency in energy use and charging protocols.	EVSE energy efficiency ratios. Number of sites with automated, ISO 15118-based EV charging.
11	2	Support for sustainable urban mobility via seamless and reliable EV charging.	EVSE penetration in urban locations.
	3	Promotion of integrated, sustainable urban energy planning with smart charging systems.	Metrics of grid stability and infrastructure integration.
13	3	Dissemination of applied clean energy technologies contributes to awareness and institutional capacity on climate mitigation.	Presentations, publications, and workshops delivered. Number of stakeholders reached or institutions involved.

Agradecimentos

Gostaria, em primeiro lugar, de agradecer à minha família, especialmente aos meus pais, que são em grande parte responsáveis pela minha educação, resiliência e pela pessoa que sou hoje. Agradeço também aos meus três irmãos, que sempre estiveram presentes quando precisei, tanto no meu percurso escolar e académico como no quotidiano e que são para mim grandes exemplos a seguir.

Em segundo lugar, quero expressar a minha gratidão à FCUP, onde realizei a licenciatura em Engenharia Física, por me dar bases sólidas que continuam a ser fundamentais no meu percurso académico e profissional, além de me ter permitido conhecer pessoas excecionais. À FEUP, onde concluio agora o mestrado em Engenharia Eletrotécnica, agradeço pela formação rigorosa e enriquecedora que recebi.

Quero ainda deixar um agradecimento especial ao INESC TEC, e em particular ao laboratório SGEVL e a todos os seus membros, que me proporcionaram, durante o último ano, um ambiente de trabalho e aprendizagem profissional verdadeiramente valioso. Agradeço em especial ao meu orientador, Justino Rodrigues, pela sua disponibilidade e orientação, bem como ao José Silva e ao Pedro Pascoal pelo apoio técnico prestado, tanto ao nível do software e formal do protocolo ISO 15118, como na vertente de hardware e implementação.

Por fim, quero agradecer aos meus amigos de longa data por fazerem parte do meu círculo próximo e por partilharem comigo tantos bons momentos ao longo dos anos. E, de forma especial, à Inês Nogueira, por estar sempre ao meu lado, por acreditar em mim e por me motivar a nunca desistir.

Frederico Ramos

"Tudo vale a pena se a alma não é pequena"

Fernando Pessoa

Contents

List of Figures

List of Tables

Listings

List of Acronyms

AC	Alternating Current
BNEF	Bloomberg's New Energy Finance
CA	Certificate Authority
CP	Control Pilot
CPO	Charge Point Operator
CSMS	Charging Station Management System
DC	Direct Current
DTO	Device Tree Overlay
EXI	Efficient XML Interchange
EV	Electric Vehicle
EVCC	Electric Vehicle Communication Controller
EVSE	Electric Vehicle Supply Equipment
HLC	High-Level Communication
HPGP	HomePlug Green PHY
IEC	International Electrotechnical Commission
INESC TEC	Institute for Systems and Computer Engineering, Technology and Science
ISO	International Organization for Standardization
LLC	Low-Level Communication
MME	Management Message Entry (management messages between PLC nodes)
MO	Mobility Operator
NMK	Network Membership Key
NID	Network Identifier
OEM	Original Equipment Manufacturer
OSI	Open Systems Interconnection
PLC	Programmable Logic Controller
PnC	Plug & Charge
PKI	Public Key Infraestructure
PP	Proximity Pilot
PWM	Pulse Width Modulation
PV	Photovoltaic
SDP	SECC Discovery Protocol
SECC	Supply Equipment Communication Controller
SGEVL	Smart Grids and Electric Vehicles Laboratory
SLAC	Signal Level Attenuation Characterization
SLAAC	Stateless Address Auto-configuration
SoC	State of Charge
SPI	Serial Peripheral Interface
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
V2G	Vehicle-to-Grid
V2GTP	Vehicle-to-Grid Transfer Protocol

Chapter 1

Introduction

In recent decades, the growing awareness of our climate's decline, largely attributed to carbon emissions from industrial activities and individual consumption, has made it essential to invest in alternatives that do not involve the burning of fossil fuels. A significant area for progress is the automotive industry, where traditional combustion vehicles are increasingly being replaced by electric alternatives.

Coupled with other factors, such as rising fuel prices and advancements in battery technologies this transition has been gradually expanding worldwide. According to Bloomberg's New Energy Finance (BNEF) projections, passenger electric vehicles (EVs) sales are expected to exceed 30 million in 2027 and grow to 73 million per year in 2040, contributing 33% and 73% to global car sales in those years, respectively [1], as shown in Figure 1.1.

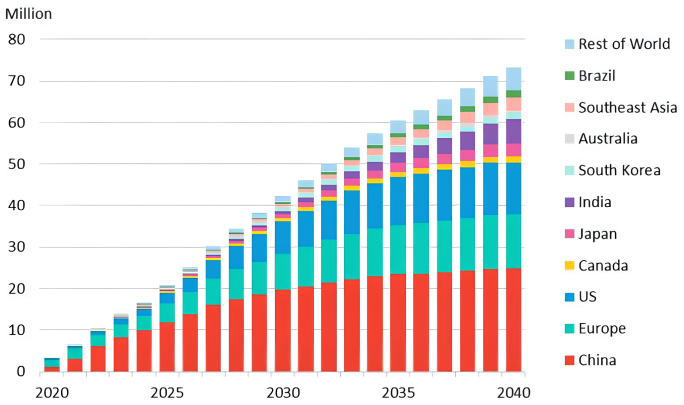


Figure 1.1: EV sales by market in BNEF's Economic Transition Scenario. From [1]

As the industry grew, so did the need for increased sophistication and standardization. These standards are essential for enhancing safety regulations, establishing mechanical and electrical norms, predicting potential issues, and ensuring compatibility across technological generations and countries' jurisdictions.

The development of theses protocols is guided by the general agreement on acceptable performance levels and procedures. This enhances the economic efficiency of development as producers and developers can share in best practices and lessons learned [2].

Such regulations apply not only to the vehicles themselves but also their charging infrastructure, including the integration with electric grids, and they facilitate the adoption of new technologies and innovations. Organizations like the International Electrotechnical Commission (IEC) and the International Organization for Standardization (ISO) provide several standards that support these objectives, promoting the safe and efficient development of electric vehicle technologies.

1.1 Motivation

The widespread adoption EVs introduces a range of challenges that must be addressed to ensure their sustainable integration. These challenges span from individual-level concerns, such as user experience, charging convenience, and vehicle life cycle considerations, to broader systemic issues, including the stability, reliability, and adaptability of the electric grid. These issues are intrinsically connected to the need for standardization in electric vehicle charging infrastructure and communication protocols.

1.1.1 Grid Stability and Infrastructure

The growing adoption of electric vehicles significantly increases the demand on the energy grid, presenting new challenges for ensuring a reliable and sufficient energy supply. Maintaining grid stability and addressing the additional energy has therefore become one of the highest priorities for the energy sector. If the distribution capacity remains unprepared or insufficiently upgraded, it could become the primary bottleneck of EV integration into the grid [3].

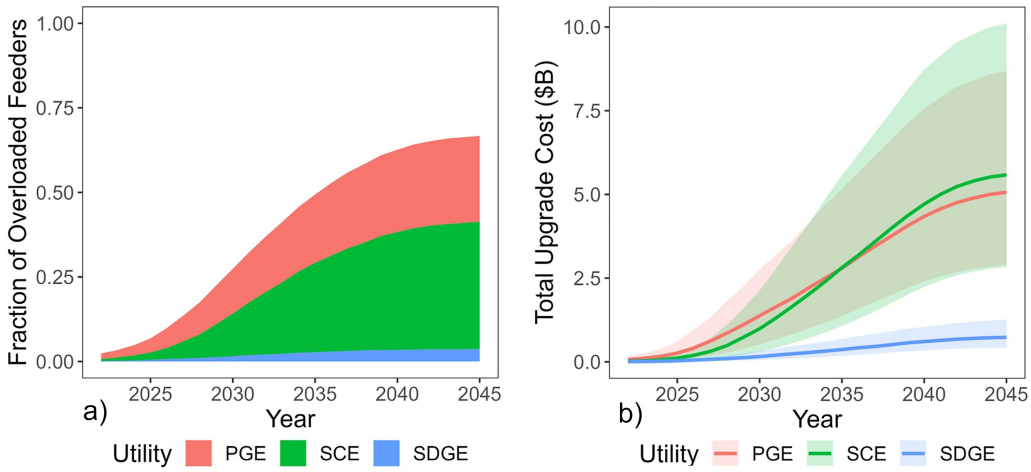


Figure 1.2: a) Share of feeders overloaded by EV charging demand for several utility companies in California. b) Total feeder upgrade costs for several utility companies in California, showing the median and 25th to 75th percentiles. Adapted from [4]

Furthermore, if predictions hold true, the cost of grid upgrades is expected to rise steadily over the coming decades to prevent the risk of increased grid overload and potential failures. This is evidenced by Figure 1.2 from a study on California's state grid, a pioneer in EV adoption, in the USA [4].

Such challenges highlight the need for other innovative solutions beyond infrastructure upgrades that support this transition effectively while also mitigating overall costs. One viable approach to address this problem is the implementation of Vehicle-to-Grid (V2G) technology, which enables bidirectional energy flow and greater interoperability between the electric vehicle and the power grid through the charging infrastructure. This can be achieved by developing standardized communication protocols and ensuring their widespread implementation to optimize efficiency and reliability.

Since battery cycle life is inversely related to the number and depth of charge-discharge cycles, [5] proposes an Electric Vehicle Charge-Discharge (EVCD) optimization model to minimize battery wear while supporting grid services. This requires close interoperability between the EV and the grid through the Electric Vehicle Supply Equipment (EVSE), as effective information exchange about battery constraints can enhance both vehicle longevity and grid stability.

Figure 1.3 shows the dynamic reduction in battery cycle life due to V2G operation, highlighting the trade-off between energy services and battery degradation. The EV owner or the EV's software can opt out of V2G service provisioning at any time, depending on the level of battery degradation deemed acceptable [5].

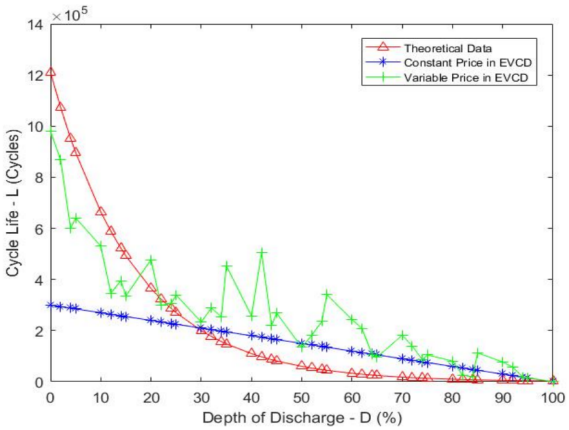


Figure 1.3: Dynamic cycle life for a Li-ion battery after V2G operation. From [5].

1.1.2 User Behavior and Intelligent Load Management

Efficient load management in electric vehicle (EV) charging infrastructure is becoming increasingly important as EV adoption grows. According to the authors of [6], load management can be significantly improved by analyzing user charging behavior and estimating parameters such as relative state of charge (SoC) and expected session duration. Their results show clear correlations between session start time, energy needs and duration (faster charging), as well as user-specific

SoC patterns. Some of these parameters, if available in the EV can be exchanged with the charging station for a more efficient charging session.

1.1.3 Standardization

The IEC-61851-1 standard compiles the most basic form of communication between the EV and the Electric Vehicle Supply Equipment (EVSE), serving as the backbone of the charging module. It uses a basic analog method for acknowledging charging status (on/off) and transmits the maximum allowed current that the EV can draw. Due to its simplicity, it is highly reliable, but it imposes limitations on the amount of information exchanged between the EV and the EVSE, thereby restricting the level of control over the charging session. The standard ISO 15118, addresses a more complex communication between the electric vehicle and the charging station through digital means, playing a crucial role in supporting the charging process. Its implementation will significantly enhance the integration of EVs into the energy system, such as enabling Vehicle-to-Grid (V2G) functionality, ultimately contributing to the stability and efficiency of the grid by using smart charging features.

It enables the transmission of larger volumes of information, supporting a wider range of potential actions. To safeguard this process, the protocol ensures that such data is transmitted securely, protecting potentially confidential information critical to the charging process. This lays a strong foundation for future advancements in security measures.

1.2 Objectives

ISO 15118-2, the second part of the protocol, defines the specifications for the network and application layers, including communication security requirements. The latest version, ISO 15118-20, released in 2022, expands on these specifications, introducing enhanced functionalities to better support both driver mobility demands and the grid's need for stable, energy-efficient, and secure operation.

The proposed research and consequent dissertation will be conducted at the Smart Grids and Electric Vehicles Laboratory (SGEVL) of INESC TEC, which provides an excellent environment, equipped with advanced facilities and supported by experienced researchers specializing in this field. It will study the referred standard to develop a software and hardware solution for next-generation chargers, enabling further analysis. The main objectives include:

2) Develop and integrate a hardware interface capable of supporting the ISO 15118 communication layer, ensuring full compatibility with EVCC implementations.

3) Evaluate the implementation using a certified EV simulator equipped with ISO 15118-20 capabilities, assessing interoperability, Plug & Charge (PnC), and V2G behaviors.

4) Validate the solution with a real electric vehicle supporting ISO 15118-2 (and, if possible, ISO 15118-20), to verify real-world compatibility.

5) Assess Transport Layer Security (TLS) integration in the protocol stack, identifying implementation challenges and evaluating performance and compliance with cybersecurity requirements.

In summary, the main objectives are to examine the ISO 15118-20 standard, its capabilities and its security aspects, which are intrinsically linked to its functionality. As illustrated in the report, this standard is only beginning to see implementation, with limited field testing. A thorough study and rigorous testing are therefore essential to contribute to potential refinements and to advance the future of electric vehicles.

1.3 Document Structure

This document is organized into the following chapters:

Chapter 1 — Introduction Provides an overview of the context, motivation, and objectives of this dissertation. It highlights the global shift toward electric mobility, the challenges associated with electric vehicle (EV) adoption, and the critical importance of standardization in enabling interoperability and grid integration.

Chapter 2 — Background and Literature Review Presents a comprehensive review of the communication standards for EV charging, with a focus on the evolution of ISO 15118 and its significance. The chapter also examines protocol architecture and security considerations, supported by insights from academic and industry research.

Chapter 3 — Development Methodology Details the methodology used for developing and integrating ISO 15118 into an existing AC EVSE. It covers hardware and software integration, the use of simulation tools, testing strategies, and the overall work plan for the project.

Chapter 4 — Results and Discussion Analyzes the outcomes of the implementation and testing activities. The chapter evaluates protocol reliability, flexibility, security performance, and real-world compatibility based on results from both simulated environments and testing with actual EVs.

Chapter 5 —Conclusions and Future Work Summarizes key findings and anticipated contributions of the work. Highlights potential improvements and outlines future directions for enhancing EVSE interoperability and grid services.

Together, these chapters provide a comprehensive framework for the dissertation, from the context and motivation to the theoretical background, methodologies, and anticipated conclusions. This organization aims to address critical challenges in EV integration through a focused study on

standardization. It seeks to enhance the efficiency and reliability of electric vehicle infrastructure, laying the groundwork for future advancements in the field.

Chapter 2

Background and Literature Review

This chapter will explore the theoretical fundamentals and literature review of EV charging and communication protocols, focusing on their evolution and significance in the industry. While the emphasis will be on the international standards used in Europe (ISO and IEC), a brief comparison with other standards will also be included.

Some of the choices made by the standards will be discussed, along with potential alternatives, including a comparison of their advantages and disadvantages. Additionally, the main hardware components and mechanisms of these protocols will be examined in detail, highlighting their functionality. Attention will also be given to security measures integrated into these protocols to ensure safe and reliable communication.

2.1 Communication standards and their evolution

This section outlines the evolution of key standards relevant to the development of this dissertation. A broad timeline is presented in Figure 2.1.

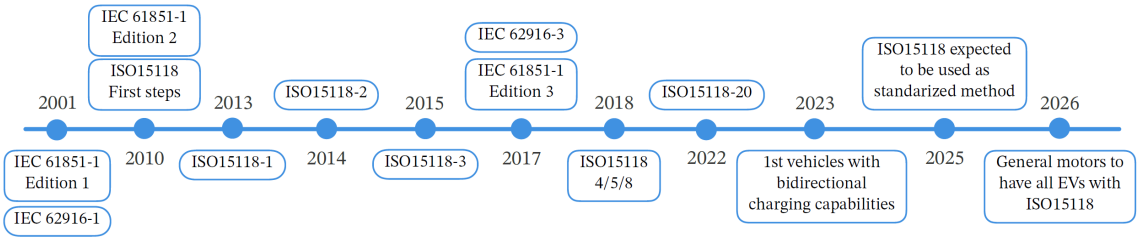


Figure 2.1: Timeline of key EV charging standards and expected future developments.

2.1.1 IEC 62196-3: *Plugs, vehicle connectors and vehicle inlets*

This standard outlines the requirements for charging outlets and vehicle connectors/inlets, which, although not directly central to the scope of this dissertation, is important for providing context. Initially released in 2009, IEC 62196-1 enhanced the Type 1 connector introduced in 2001, and laid the specifications for the development of the Type 2 connector.

The connector is designed for single-phase to three-phase Alternating Current (AC) charging. Later in 2017 IEC 62196-3 expanded on its previous release and added a Direct Current (DC) connection to the plug, allowing for models which include this connection to use either AC or DC charging, while occupying small space on the chasis. The connector was called Combined Charging System Type 2 (CCS2).

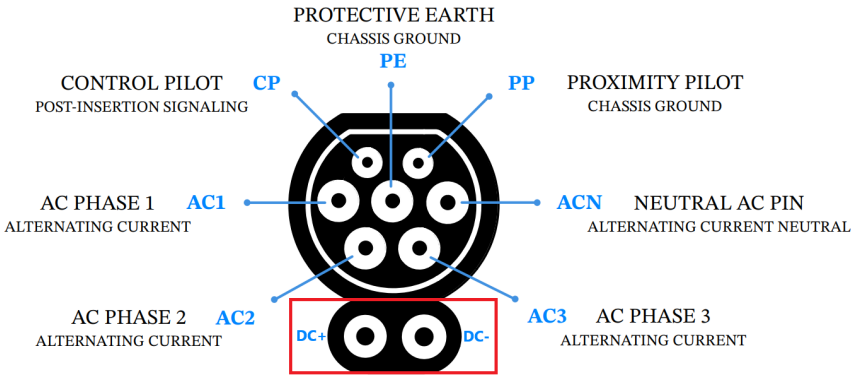


Figure 2.2: CCS Type 2 Connector. The red section highlights the CCS addition.

A key focus for this dissertation will be the Control Pilot (CP), a smaller wire in the plug used for communication paired with the Proximity Pilot (PP), which serves as the return path for the CP signal. These components will be explored in greater detail later.

2.1.2 IEC 61851-1: *LLC, Basic Charging*

Since the introduction of EVs, manufacturers have faced the challenge of accommodating diverse charging requirements across various car models, particularly concerning battery specifications. To convey charging status and battery parameters between the EV and EVSE, different communication methods have been developed, categorized as Low-Level Communication (LLC) and High-Level Communication (HLC) [7].

To address these challenges, the IEC 61851-1 standard was developed, featuring LLC communication. Its first edition, released in 2001, introduced a communication method specifically designed for EVs and charging stations. Later, in 2017, the release of the third edition of IEC 61851-1 expanded the standard to include support for DC charging. Using the CP wire, the EVSE generates a Pulse-Width Modulation (PWM) of fixed frequency $1kHz$ signal to initiate analog communication with the EV. The EVSE serves as the generator of the signal, transmitting information to the EV by varying the duty cycle of the PWM as shown in Figure 2.3.

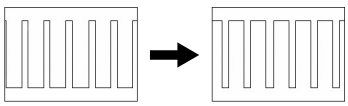


Figure 2.3: EVSE-to-EV communication method by changing the PWM duty cycle.

On the other hand, the EV communicates back to the EVSE by switching resistors into the circuit. These resistors modify the amplitude of the CP's signal, which the EVSE interprets to

understand the vehicle's status and readiness as it can be seen in Figure 2.4. This bidirectional communication ensures that both the EVSE and the EV remain synchronized during the charging process, maintaining basic efficiency and safety.

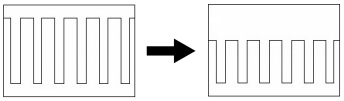


Figure 2.4: EV-to-EVSE communication method by changing the PWM amplitude.

A typical CP circuit is shown in Figure 2.5, which illustrates the electrical connection between the EV and the EVSE.

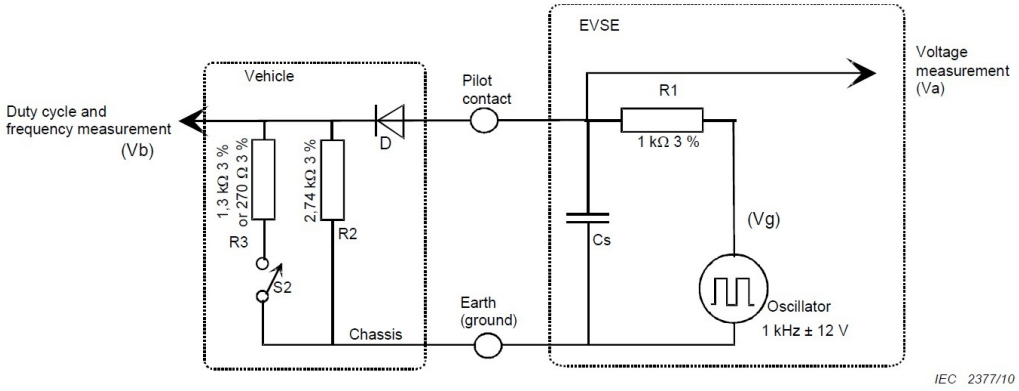


Figure 2.5: Typical control pilot circuit. From [8]

The communication protocol determines the charging parameters through the duty cycle and the states of the pilot signal. Table 2.1 summarizes how the duty cycle corresponds to the maximum current.

Table 2.1: Duty Cycle Interpretation and Maximum Current Available.

Duty Cycle Range	**Maximum Current Available**
Duty cycle < 3%	Charging not allowed
3% < Duty cycle < 7%	Digital communication required for DC or on-board chargers
7% < Duty cycle < 8%	Charging not allowed
8% < Duty cycle < 10%	6 A
10% < Duty cycle < 85%	Available current = (% Duty cycle) × 0.6 A
85% < Duty cycle < 96%	Available current = (% Duty cycle − 64) × 2.5 A
96% < Duty cycle < 97%	80 A
Duty cycle > 97%	Charging not allowed

The EV-EVSE states are explained by Table 2.2, which are induced by the car on the CP circuit by altering the signal amplitude.

Table 2.2: Pilot Functions and Vehicle States.

State	Vehicle Connected	Ready to Charge	Description
A	No	No	12 V, Vb = 0 V
B	Yes	No	9 V, R2 detected
C	Yes	Yes	6 V, ventilation not required
D	Yes	Yes	3 V, ventilation required
E	Yes	No	0 V, utility or EVSE problem
F	Yes	No	-12 V, EVSE unavailable

When charging becomes possible from the EV's side, the charging station determines whether it is safe to close the relay and supplying power to the vehicle. If deemed safe, the relay is closed, initiating power delivery and connecting the vehicle to the grid. Figure 2.6 illustrates a typical charging session from start to finish, showing its effects on the various components of the EV-CP-EVSE system.

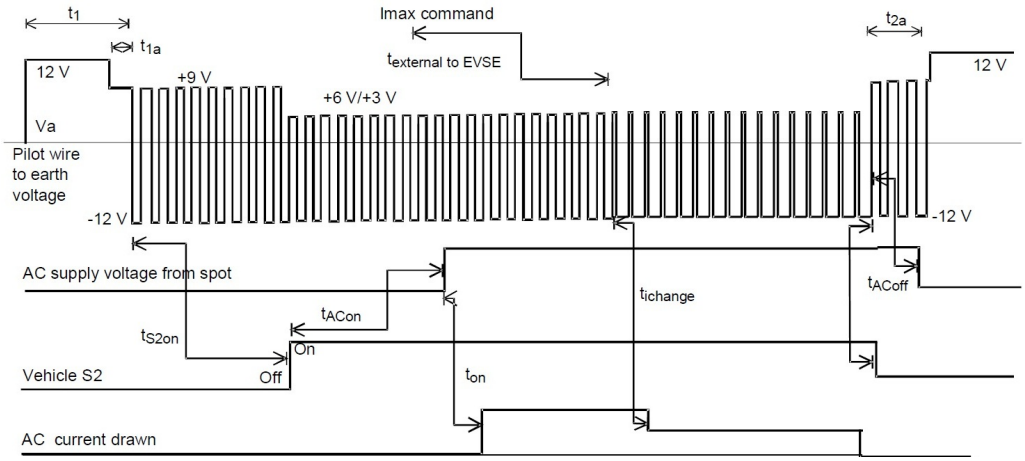


Figure 2.6: Typical charging cycle under normal operating conditions. From [8]

Although this protocol is very reliable duo to its simplicity by utilising Low Level Communication (LLC), it has significant limitations regarding the amount and type of information that can be communicated. It establishes the back bone (basic charging) of EV charging, in which more complex protocols as ISO 15118-3 will later be supported on.

2.1.3 ISO 15118: *HLC*

Building on the basic communication method established by IEC 61851-1, which makes it compatible with existing systems, ISO 15118 proposes to overcome its limitations while ensuring backward compatibility with older charging systems. This increase in message sophistication, as [9] tells us, allows for a secure, future-proof, and user-convenient way of charging an EV in such a manner that both the driver's mobility demands and the grid's demand for a stable, energy-efficient, and safe operation are met. Since ISO 15118 is the primary focus of this dissertation, a detailed

examination of each Open Systems Interconnection (OSI) layer will be presented, focusing only on the parts relevant to the protocol's requirements, as illustrated in Figure 2.7, to clarify its structure and functionalities.

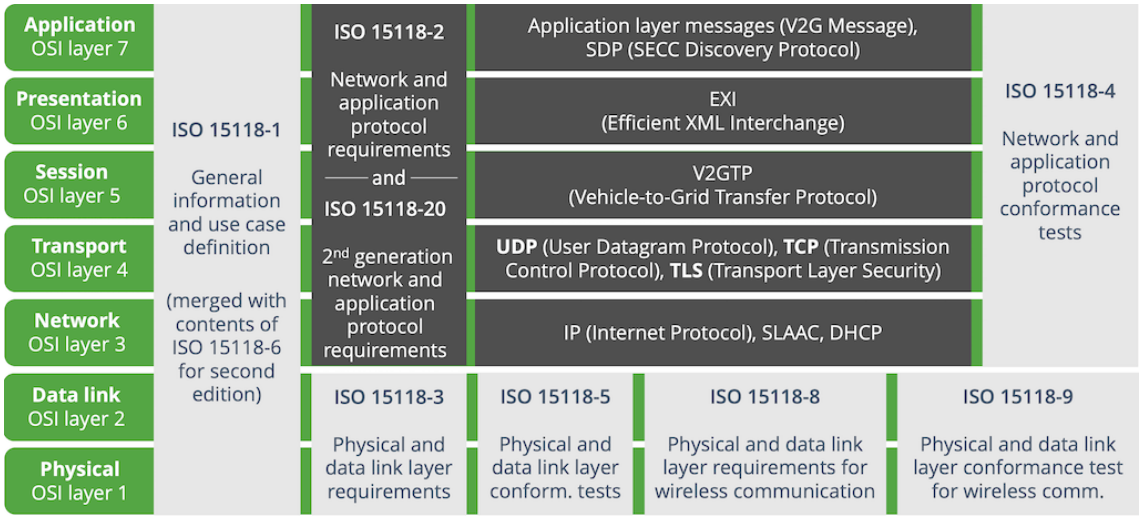


Figure 2.7: The eight parts of ISO 15118 and their relation to the seven OSI layers. From [9]

2.1.3.1 Physical: *OSI Layer 1*

Power Line Comunication (PLC) is a technique used to transmit data over the same cable used for power delivery. It achieves this by modulating data signals at higher frequencies than the power signal, allowing compatibility with AC, DC and similar systems.

In ISO 15118, the physical layer utilizes **HomePlug Green PHY (HPGP) PLC**, which operates in the 2 MHz to 30 MHz range, injecting data into the PWM transmission line (CP). It uses Orthogonal Frequency-Division Multiplexing (OFDM) to modulate the data signal, which is then superimposed on the electrical signal [10], as illustrated in Figure 2.8. OFDM works by splitting the transmitting data into multiple carrier channels, each at a different frequency, instead of the conventional transmission at a single fixed frequency.

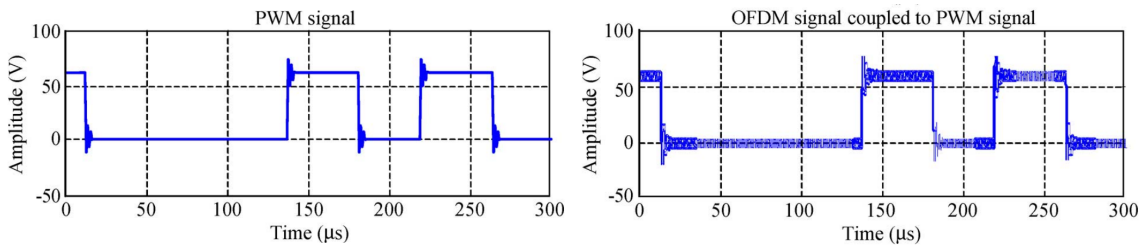


Figure 2.8: PWM signal without data transfer and with data transfer in the presence of the OFDM signal. Source: [10]

Note: To prevent confusion, it is important to distinguish between "PLC (Power Line Communication)," which refers to data transmission over power lines, and "PLC (Programmable Logic Controller)," which refers to industrial control devices. Unless explicitly stated otherwise, the term "PLC" in this dissertation will refer to Power Line Communication.

An advantage in using this method to transmit digital data between the EV and EVSE is it does not require any additional changes to the cable infrastructure. The only requirement is a PLC module capable of handling signal injection and reading at both nodes since the CP cable already exists inside the main charging cable. In a addition to the base protocol, ISO 15118-8 introduces an alternative option, allowing wireless communication between the two agents. This expansion provides greater system flexibility and introduces new security considerations in data transmission, which will be discussed later.

2.1.3.2 Data link: *OSI Layer 2*

Before any digital communication can be established certain considerations need to be made. The quality of the connection should be tested to ensure the vehicle and station are physically connected and there is minimal risk of data loss. This is achieved using the **Signal Level Attenuation Characterization (SLAC)** protocol, which operates in the second layer of the OSI model within the context of ISO 15118-3.

Several messages are exchanged between the EV and EVSE, with the number of lost messages and time intervals measured to calculate signal attenuation parameters for stable communication. By means of the signal strength measurement, the EV determines which EVSE nodes are available and which one is the correct counterpart to connect to. The EVSE measures the attenuation of the signal from the EV for different frequency groups and reports the results back to the EV. Based on these reports, the EV identifies the directly connected EVSE [11].

A logical network is first established between the PLC modules of both agents. Figure 2.9 illustrates the general management message entry (MME) exchange within the network, where the edges represent the definition of the Network Membership Keys (NMKs) for each PLC module using a predefined secure random algorithm. The NMK acts as a secret key, enabling secure communication between the EVSE processor and its PLC module, as well as between the EVCC and its PLC module, with the PLC modules communicating directly between themselves. To advertise the network, a Network Identifier (NID), derived from the EVSE's NMK, is broadcasted.

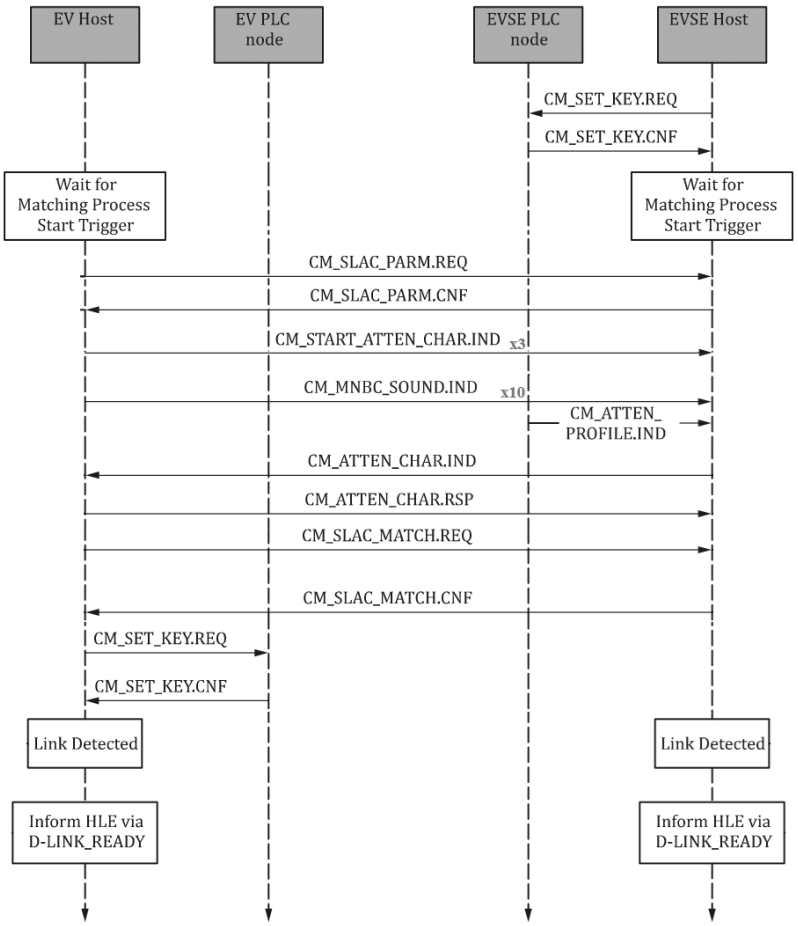


Figure 2.9: Simplified sequence of SLAC HPGP matching process. Adapted from [11]

These attenuation parameters and secret keys continue to be used by the PLC module in subsequent communication layers to ensure secure data exchange. The NMK plays a critical role in maintaining private communication between the EVSE and its PLC module. A single PLC module can manage communication with multiple vehicles simultaneously, as it can maintain several NMKs and transmit through different NIDs, each linked to a unique NMK, facilitating separate network connections. This means that multiple EVSE units can share a single PLC module, as each EVSE can be identified through its respective NMK and NID pair. Once a charging session concludes and the vehicle disconnects, both the NMK and attenuation parameters are reset, requiring a new SLAC procedure for the next vehicle connection. The SLAC process may be attempted up to three times if the initial attempt fails. Between retries, the CP state must transition to state E or F (see Table 2.2), and the communication parameters must be reset as if the EV had disconnected. These transitions, like all protocol defined steps, are governed by timing constraints. For AC charging, timers such as `T_EVSE_SLAC_INIT` and `T_STEP_EF` regulate this retry mechanism, ensuring synchronization between SLAC retries and the initiation of High-Level Communication (HLC). This retry sequence is illustrated in Figure 2.10.

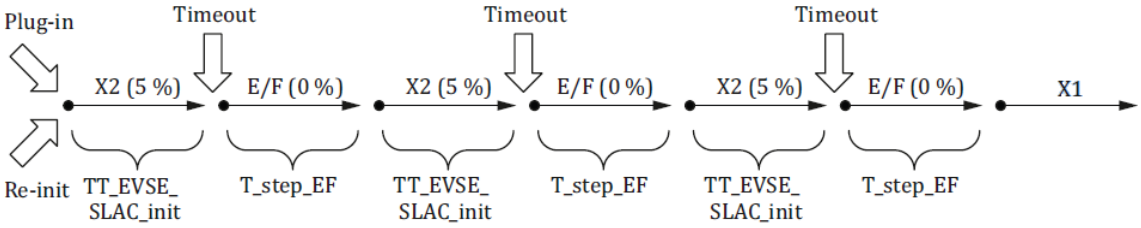


Figure 2.10: Timing diagram for SLAC process highlighting EVSE-side timeouts. From [11].

2.1.3.3 Network: *OSI Layer 3*

On layer three, the network layer, it is necessary for the Internet Protocol (IP) to assign unique IP addresses for the EVCC and SECC, necessary for routing data packets between both entities [9] using Transmission Control Protocol (TCP). IPv6 addresses are used for communication, with address management handled by **Stateless Address Auto-configuration (SLAAC)**. SLAAC enables IPv6 nodes to autonomously generate and configure their own IP addresses based on network prefixes received from Router Advertisement (RA) messages, eliminating the need for a dedicated DHCP server [12].

This decentralized address assignment mechanism simplifies network configuration, particularly in dynamic environments where nodes frequently join or leave the network. In the context of ISO 15118, this flexibility is essential as it allows electric vehicles (EVs) and charging stations (EVSEs) to establish communication seamlessly upon connection without manual IP configuration. However, security considerations such as the risk of SLAAC attacks, where an attacker can inject fake Router Advertisement (RA) messages, impersonate a legitimate router, and gain control over the network traffic, are threats to take into consideration that can lead to initial data interception and denial-of-service attacks [12].

2.1.3.4 Transport: *OSI Layer 4*

After the IP addresses have been set, a TCP/IP connection must be established to enable the exchange of data packets. Initial SECC Discovery Protocol (SDP) messages are transmitted using User Datagram Protocol (UDP) broadcast to establish the connection and configure the IP addresses, followed by the establishment of a TCP connection. There is also a timing constraint related to the interface between the SDP and SLAC procedures. If no SDP request is sent by the EV in `V2G_Comm_Performance_Time` the V2G Communication Session must finish, the SLAC-related values are reset and the EVSE may fall back to LLC communication (as per ISO). This behaviour prevents the charging process from becoming stuck indefinitely in this communication step, in case of communication failure or unexpected behaviour [13].

Even though critical pieces of information and certificates are exchanged, **Transport Layer Security (TLS)** was not imposed in ISO 15118-2. However, due to several security breaches, ISO 15118-20 now mandates its use to ensure secure communication offering increased performance, stronger cryptographic standards, and reduced latency.

Mutual authentication is used, where the EVCC and EVSE verify each other's identities. Certificate chains, used throughout TLS, consist of a sequence of certificates issued and signed by higher Certificate Authorities (CAs), ultimately tracing back to a trusted Root CA. The certificate structure example in Figure 2.11 illustrates the full set of certificates and signing relationships required for a complete V2G communication session with all functionalities enabled. Key authorities involved include the Original Equipment Manufacturer (OEM) and the Charge Point Operator (CPO), as detailed in Table 2.3. In private infrastructure deployments, where a single entity manages all certificates for both the EV and the EVSE, the certificate chain can be simplified, intermediate SubCA certificates are not mandatory in such cases.

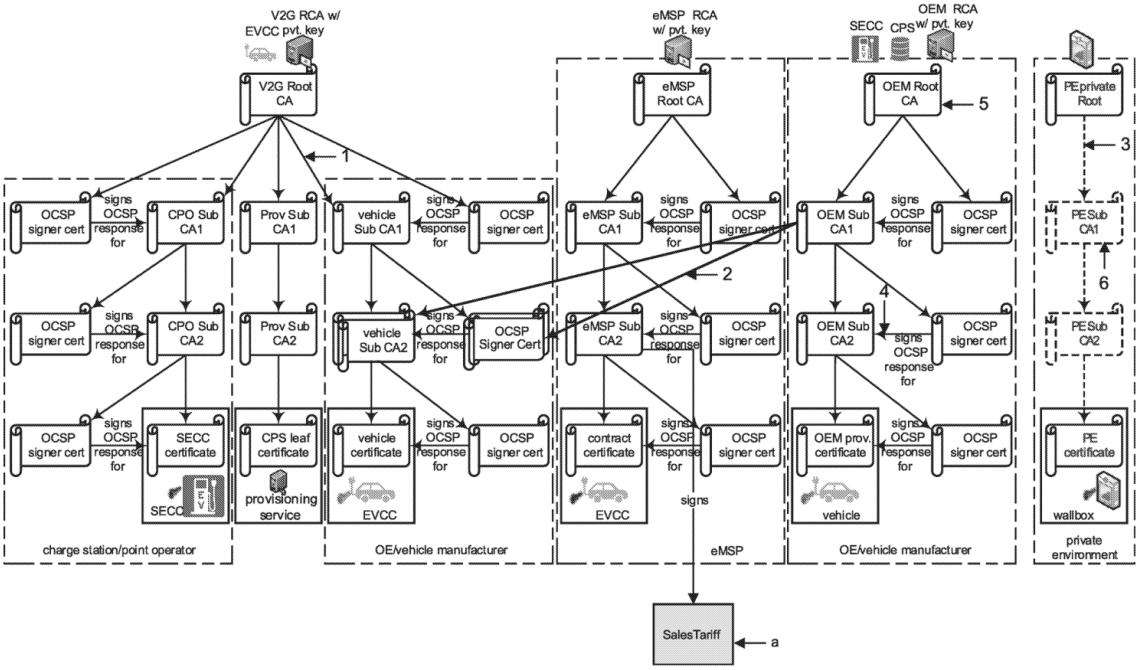


Figure 2.11: Example of certificate chain hierarchy for EVSE and EVCC authentication. From [14].

The EVSE sends a certificate chain to the EVCC during the initial handshake. This chain includes the SECC Certificate, which is the EVSE's identity certificate, the CPO Sub-CA Certificate, issued by a trusted CA under the CPO's infrastructure, and the Root CA Certificate, which is the highest-level authority certificate.

The EVSE also provides to the EVCC an OCSP response (Online Certificate Status Protocol) for each certificate in the chain. The OCSP response confirms whether the certificates sent are still valid or have been revoked by the issuing CA. The EVCC receives the certificate chain and validates it by checking the integrity using cryptographic hash functions to ensure the data has not been altered, verifying the certificate chain from the EVSE certificate back to the Root CA Certificate, and confirming the validity by checking the OCSP response provided by the EVSE.

Table 2.3: Public Key Infrastructure (PKI) Authorities and Roles in ISO 15118

Authority	Description	Example Entities
Root CA	The highest authority responsible for issuing certificates to subordinate CAs. It serves as the root of trust in the PKI structure and validates all certificates within the chain.	DigiCert, GlobalSign, Hubject
OEM CA	Certifies electric vehicles (EVCC) by issuing certificates to the vehicles. It ensures the vehicle can securely participate in Plug & Charge (PnC) sessions.	Tesla, BMW, Volkswagen
CPO CA	Certifies charging stations (SECC) by issuing certificates to the charging points. It ensures the charging station can be trusted during PnC transactions.	IONITY, BP Pulse

If the OCSP response is valid and the hash matches, the certificates are considered authentic. After this verification, the EVCC replies by sending its certificate chain, including its own certificate, the OEM Sub-CA Certificate, and the Root CA Certificate. The EVSE queries an OCSP server to verify the validity of the certificates. The EVSE then follows the same validation steps (as the EVCC did to the EVSE) to authenticate the EVCC.

Note: A hash is a unidirectional, fixed-length representation of data created using a mathematical algorithm. It ensures data integrity by detecting any modifications to the original data.

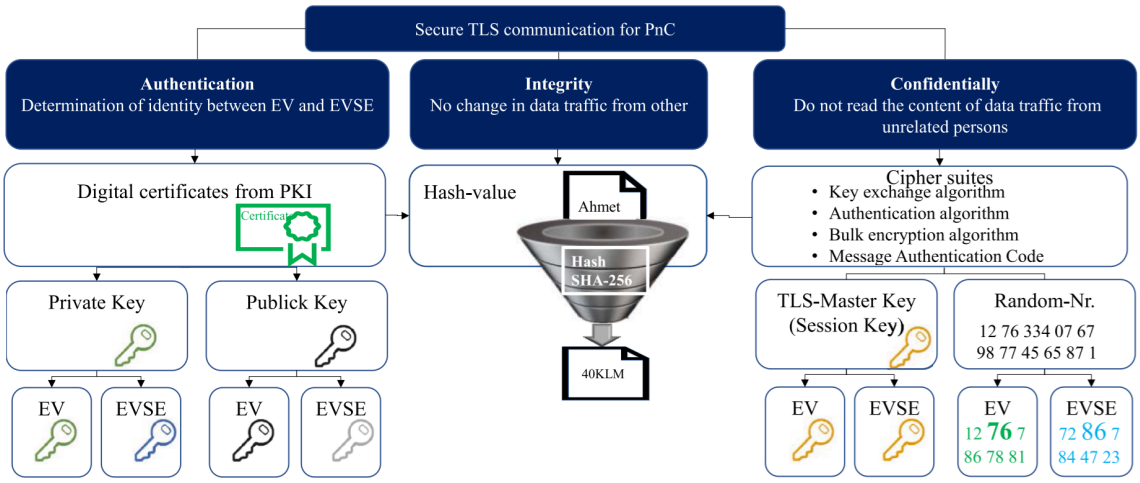


Figure 2.12: An Architecture overview of the EV charging ecosystems with PnC. Source: [15]

Figure 2.12 shows the main security features used in the TLS communication. This technology is the most convenient and most secure for EV charging. For these reasons, PnC is very attractive and topical for research and development, especially with TLS handshake. Without a successful TLS handshake, the charging process is interrupted, and no charging process may be started [15].

However, despite the security improvements of ISO 15118-20, careful implementation remains crucial. Improper integration has led to vulnerabilities, including reflection attacks, as confirmed by Drucker [15]. When correctly implemented, TLS 1.3 ensures secure, efficient, and low-risk charging sessions. This way, secure certificate exchange between the EV and EVSE is ensured, enabling the implementation of advanced charging functionalities in a protected manner.

2.1.3.5 Session: *OSI Layer 5*

The session layer, layer five, is responsible for defining the structure and semantics of transmitted messages. In the context of ISO 15118-2 and ISO 15118-20, this layer is defined by the Vehicle-to-Grid Transfer Protocol (V2GTP). V2GTP is based on TLS+TCP, utilizing a combination of IP addresses and port numbers, ranging from 49152 to 65535. It primarily defines a structured format consisting of a header and a payload, allowing for the efficient separation and processing of V2G messages [14].

The header, as with many standardized protocols, includes the version of the protocol, which is also verified with the inverse protocol version, to ensure correct message reception, calculated as one byte (255) minus the protocol version. It also categorizes the payload into payload types (to handle multiple individual messages within the payload) and specifies the total payload length. This structure ensures clear message parsing and efficient communication. The overall format is summarized in Table 2.4.

Table 2.4: V2GTP message structure and header structure.

Message part	Header								Body		
Byte No.	1	2	3	4	5	6	7	8	9	...	4294967295
Field	Protocol version	Inverse protocol version	Payload type		Payload length				Payload data		

2.1.3.6 Presentation: *OSI Layer 6*

At the presentation layer, the system is responsible for unpacking the transmitted payload into an easy-to-interpret format. The standard defined for this purpose in ISO 15118 is the Efficient XML Interchange (EXI). EXI allows XML messages to be encoded in a compact binary format using UTF-8 character encoding.

From layer 6, a certain V2GTP message is divided into its individual payloads, which are in EXI format, and then converted to XML in this layer. This approach enables faster transmission and processing of messages, as they are only fully unpacked at this stage of the OSI model before being passed to layer 7. Similarly, messages originating from the application layer are packed into EXI format before being transmitted to lower layers. Figure 2.13 illustrates the process of packing V2G messages into EXI format, encapsulating them into V2GTP (see Section 2.1.3.5), transmitting them, and finally unpacking them back into V2G messages.

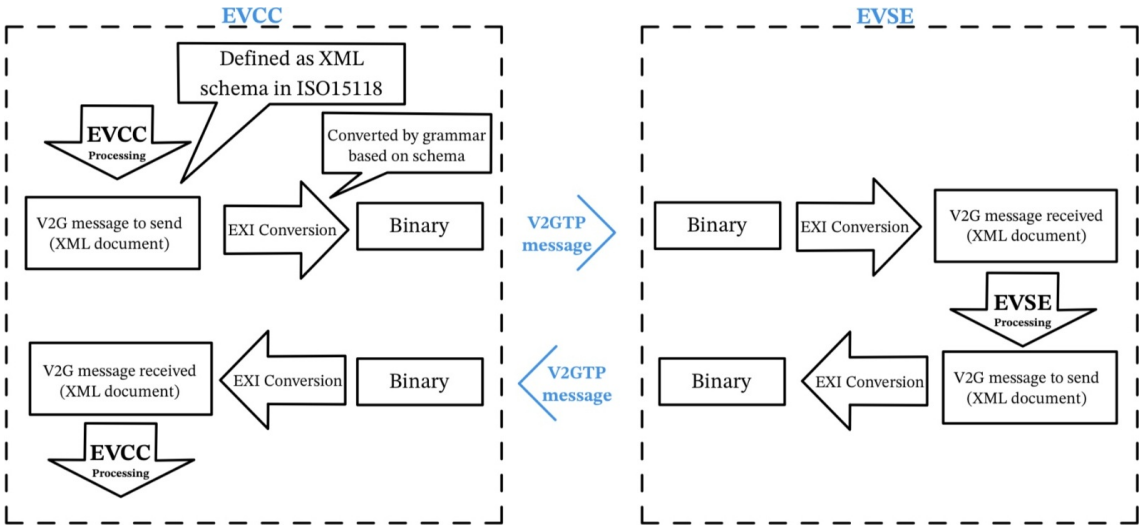


Figure 2.13: XML to EXI transformation format in V2GTP message exchange between EVSE and EVCC.

2.1.3.7 Application: *OSI Layer 7*

The application layer is responsible for overseeing the core processes, where V2G messages are constructed and interpreted. Throughout the entire charging process, the EVCC functions as the client (service requester), while the SECC operates as the server (service provider) [11]. Communication is always initiated by the EVCC, which sends the first request message, resulting in a response message from the SECC, as illustrated in Figure 2.14.

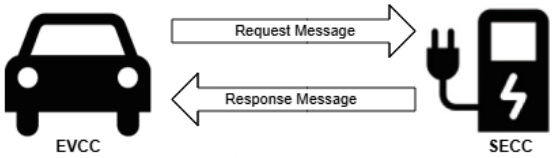


Figure 2.14: Typical message exchange between EVCC and SECC at the application layer of the ISO 15118 protocol.

While many messages can be exchanged during a charging session, the most significant ones for a Plug and Charge (PnC) scenario, as described in [14], are as follows:

- **Supported Application Protocol (SAP) Exchange**
  - **`SupportedAppProtocolReq`** – The EV sends supported protocol versions (release date, ISO -2, ISO -20, etc...) to the SECC.
  - **`SupportedAppProtocolRes`** – The SECC confirms with own supported protocol versions and sends the selected one back.
- **Session Setup**

  - **`SessionSetupReq`** – The EV requests to establish a session, including its unique ID (EVCC MAC).
  - **`SessionSetupRes`** – The SECC confirms the session and provides a session ID, allowing session resumption if needed.

- **Service Discovery**
  - **`ServiceDiscoveryReq`** – The EV requests available charging services.
  - **`ServiceDiscoveryRes`** – The SECC responds with available services, such as AC charging, payment options, and contract certificate handling.

- **Service Detail Request**
  - **`ServiceDetailReq`** – The EV requests further details about a selected service.
  - **`ServiceDetailRes`** – The SECC provides additional information about the service.

- **Payment Service Selection**
  - **`PaymentServiceSelectionReq`** – The EV transmits the wanted payment method from the ones provided.
  - **`PaymentServiceSelectionRes`** – The SECC confirms the selected payment method.

- **Payment Details (Certificate Exchange)**
  - **`PaymentDetailsReq`** – The EV provides its contract certificate and e-MAID for PnC transactions.
  - **`PaymentDetailsRes`** – The SECC acknowledges receipt and provides a challenge for authentication.

- **Authorization**
  - **`AuthorizationReq`** – The EV sends an authorization request containing the signed challenge.
  - **`AuthorizationRes`** – The SECC confirms successful authorization.

- **Charger Parameter Discovery**
  - **`ChargeParameterDiscoveryReq`** – The EV communicates its charging limits, energy requirements, and desired departure time. Additionally, it specifies the maximum resolution of the charging schedule (how many entries/tuples) that the SECC can later provide.
  - **`ChargeParameterDiscoveryRes`** – Based on the provided constraints, the SECC proposes a charging schedule that includes the maximum allowable power for each entry (step). The schedule may also incorporate a `SalesTariff` for each entry to incentivize efficient charging behavior.

- **Power Delivery**
  - **`PowerDeliveryReq`** – The EV evaluates the available schedules and selects the most advantageous one. It can either strictly follow the chosen schedule or modify it to create its own `ChargingProfile`, ensuring that it remains within the schedule's upper limits. Additionally, the EV indicates whether charging should start immediately or be delayed.
  - **`PowerDeliveryRes`** – If charging is approved, the SECC confirms power delivery. Upon receiving confirmation, the EV transitions to state C, signaling its readiness to begin charging.
- **Charging Status**
  - **`ChargingStatusReq`** – The EV sends periodic updates about the charging process.
  - **`ChargingStatusRes`** – The SECC confirms receipt and provides charging status updates or reports anomalies.

During the charging main loop, **`ChargingStatusReq/Res`** messages are always exchanged and can include meter readings. Their primary purpose, however, is to keep the session active, functioning similarly to a "ping" signal. At certain points, the EVSE can request the EV to stop charging or renegotiate the charging schedule with this message, for example, due to grid load changes or other operational constraints.

Additional messages can be exchanged during the charging process, such as metering receipt requests(**`MeteringReceiptRequest`**), which confirms the previously received meter readings from the EV. The EVSE responds with a **`MeteringReceiptRes`** to acknowledge the confirmation. The EV can send a **`PowerDeliveryReq`** message at any time to either terminate the charging session or renegotiate the schedule, for example, based on the received meter values. The sequence diagram related to the exchange of all these messages can be seen in Figure 2.15.

2.2 Advantages and Key Features of ISO 15118-20

ISO 15118-20 expands upon the earlier ISO 15118-2 standard by strengthening security measures and introducing new message structures that support additional features. This section explores the key advancements and capabilities of ISO 15118-20, focusing on improved security, PnC functionality, and Bidirectional Power Transfer (BPT). It also examines the protocol's major benefits, technical enhancements, and potential risks associated with its implementation.

2.2.1 Security and Complexity

As the data exchange between EVs and EVSEs grows in both volume and complexity, the management of this information also becomes more demanding. These charging stations facilitate the flow of information between the grid and EVs, improving energy flow management and enabling

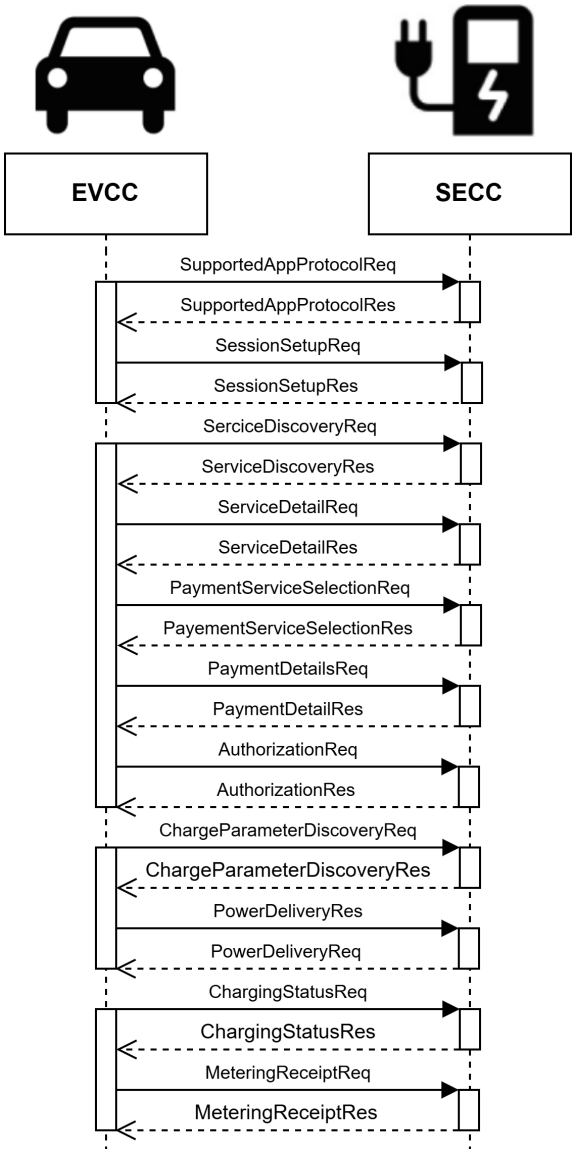
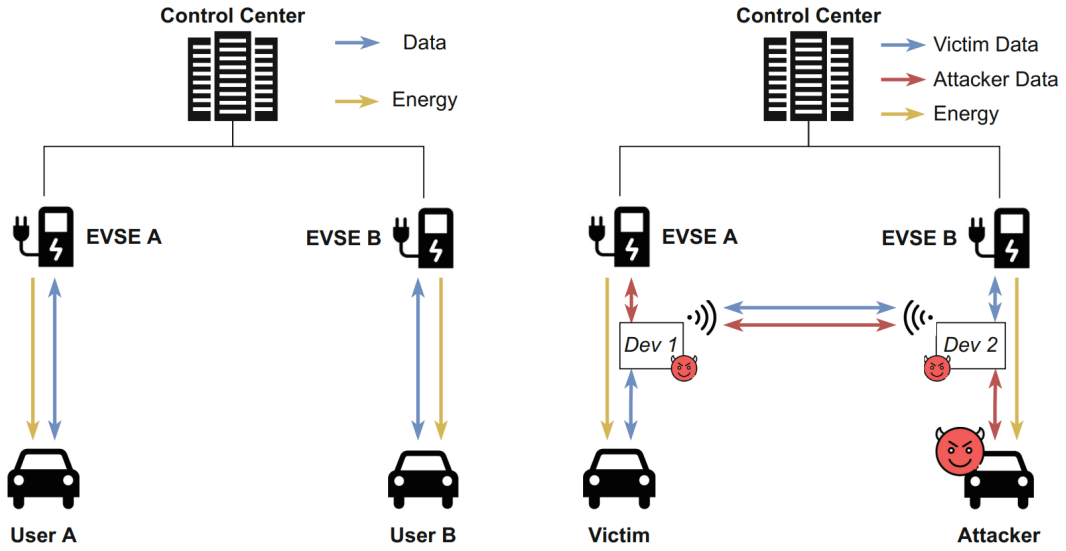


Figure 2.15: Simplified sequence Diagram for Message exchange in a charging session.

more efficient load distribution. However, as more nodes outside the control of energy operators are integrated, the system becomes increasingly exposed to potential risks and attacks[16].

With the growth of this market, new vulnerabilites keep being found and exploited. The release of ISO 15118-20 came to address many of this problems with new mandatory security features, but it has shown itself to not be immune to attacks either such as in [17], where a relay attack allows an attacker to not only discharge a victim's car through reverse 'vehicle-to-grid' charging, but additionally steal the profit from the energy too. In this type of attack, only two devices are required, such as basic microcomputers (e.g., Raspberry Pi). These can be maliciously installed at two separate charging stations as seen in Figure 2.16, each equipped with dual interfaces: one for demodulating the PLC signal on the CP pin and another for Wi-Fi connectivity [17].



(a) Without attack. (b) With *EVExchange* attack.

Figure 2.16: (a) Simplified scenario showing two EVs charging from two EVSEs, both connected to the same Control Center. (b) The same scenario as (a), but with the inclusion of malicious devices affecting the system. From [17]

There have been other studies discussing potential vulnerabilities within this protocol and the EV infrastructure as a whole. For example, [18] highlights how EVSEs could serve as an attack vector into the power grid. If a sufficient number of EVSEs were compromised and manipulated to alter their charging speed, a demand-side grid attack could be launched, potentially destabilizing the grid. Similarly, [19] describes downgrade attacks, where an EV could falsely claim to lack support for certain optional security features within ISO 15118-20, making the system more vulnerable to exploitation. A proposed solution involves the use of a Trusted Platform Module (TPM) to verify supported security features. However, this approach merely shifts the trust dependency to the TPM itself, leaving the core issue of trust delegation unresolved. In summary, the EV-EVSE connection represents a potential point of security vulnerability. Given that ISO 15118-20 is a relatively recent protocol and responsible for this communication, further testing, research and solution propositions are required to identify and address any security gaps.

2.2.2 Authentication

Previously, manual authentication required multiple steps, such as pressing a manual button, using an RFID card, a credit card or accessing a mobile app to initiate a charging session. These methods often had security vulnerabilities, making them both inconvenient and less secure for consumers. With PnC certificates, which are securely transmited in the context of ISO 15118-20 (having the previously security considerations), seamless convenient charging sessions are possible [20].

The **eMobility Account Identifier (EMAID)** is a unique identifier for the contract between the customer and the eMobility Service Provider (eMSP). It allows for the proper identification of the customer and their associated energy services during the EV charging process [14]. The EMAID ensures that the right account is linked, enabling billing and related services.

The **eMSP** is the entity with which the customer has a contract for all services related to EV energy transfer. The eMSP is responsible for managing the customer's account, authorizing charging sessions, and handling payments, though the actual billing process may be subject to local regulations [14]. The eMSP also issues the EMAID to customers for identification during the charging process.

Up-to-date certificates are required by the vehicle, which can be refreshed during the EVSE connection. These certificates include the EMAID, which holds the details of the eMSP. This information is crucial for billing and other related services. However, the ISO 15118-20 protocol itself does not manage the billing process directly. Instead, it facilitates the transmission of the necessary information to the relevant authority, according to local regulations. This technology has the potential to attract more clients to the EV industry due to its simplicity and setup speed.

2.2.3 Bidirectional Power Transfer

One of the most interesting and useful features of this protocol is its ability to influence grid load distribution, not only by reducing consumption but also by actively supplying the grid. This is achieved through Bidirectional Power Transfer (BPT) capabilities. To enable this, grid specifications, known as grid codes, must be provided by the local authorities for electrical system integration. In DC charging, the power converter is located at the EVSE, simplifying the vehicle's discharge in compliance with these codes, as the station can directly be programmed with them. In contrast, AC charging has the power conversion within the EV itself, requiring more specific data to be transmitted to the vehicle, such as the ratio of active to reactive power [9].

As discussed in 2.1.3.7, service discovery messages determine whether the session supports both charging and discharging, and once certificates are exchanged, the charge parameter discovery messages define the limits of the bidirectional power flow. This summarizes the Vehicle-to-Grid process; however, with the ISO 15118's Bidirectional Power Transfer (BPT) functionality, Vehicle-to-Everything (V2X) becomes possible. This extends the use of the vehicle's energy, allowing it to interact with various systems beyond the grid. This expanded capability is illustrated in Figure 2.17.

This array of options brings significant advantages, enabling the transition to a smart grid and smart cities. In the case of powering fleets of EVs, wether they are company vehicles, shared vehicles, or public service vehicles, which are charged through a central charging system, the benefits are substantial. During off-peak hours or periods of low demand, these vehicles could be charged from the grid. Conversely, during high-demand periods or power outages, these vehicles could supply power back to the building or grid [21].

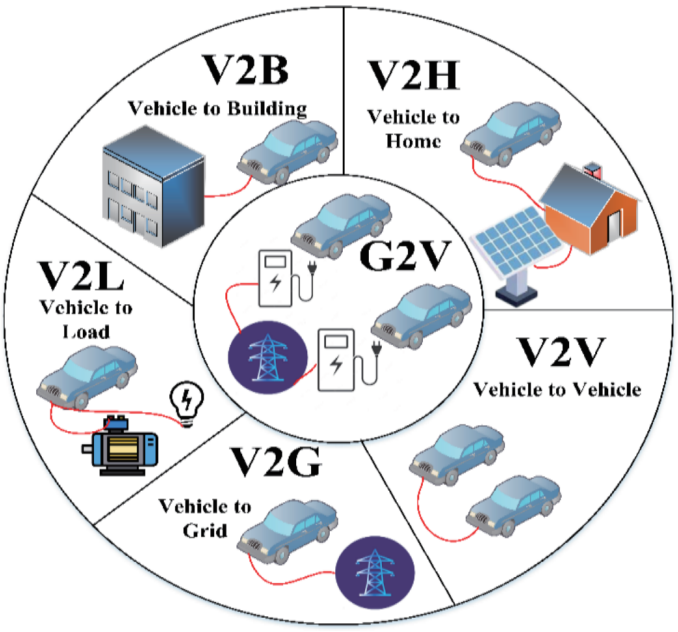


Figure 2.17: V2X topology diagram. From [21]

2.2.4 Smart Charging

One of the bases of smart charging is the control of power consumption and flow across several grid nodes, which are connected to Electric Vehicles (EVs). Using IEC 61851-1's basic charging, it is possible to control the maximum available current by adjusting the PWM (see Section 2.1.2) of the CP line. A schedule chosen by the CPO side delimits the maximum available **power** to be expended over time, which consequently modifies the maximum current and duty cycle generated. The vehicle then has $t_{ichange}$ to adjust the current being drawn to comply with the protocol.

When using digital communication with the ISO 15118 protocol, it is possible to enable a more detailed negotiation between the EV and the charging station regarding power consumption over time. The car charging process is associated with a charging schedule, which is determined by the **Charge Parameter Discovery** and **Power Delivery** messages, as referenced in Section 2.1.3.7.

Charging schedules proposed by the charging station can be influenced by higher-level central entities that define energy prices and monitor grid load. Additionally, EVSEs can also request a renegotiation of the schedule at any time. The final decision, made by the EV, takes into consideration the received information, such as charging times, tariff incentives, and the user's preferences — for example, whether the user prefers faster charging or a cheaper option (due to variable energy prices) [20]. An example of a simple schedule negotiation process can be seen in Figure 2.18.

A study conducted in Germany [22] tells us that a car stands around unused for an average of 22.6 h (94%) a day and therefore EVs, in conjunction with the currently developed technology of BPT (as discussed in 2.2.3) offer great potential for flexibility in the electricity grid contributing to its smart aspects. By using the storage capacity of EVs to support the grid, grid expansion costs can be reduced, such as by reducing grid bottlenecks or applying peak shaving. Peak shaving is the

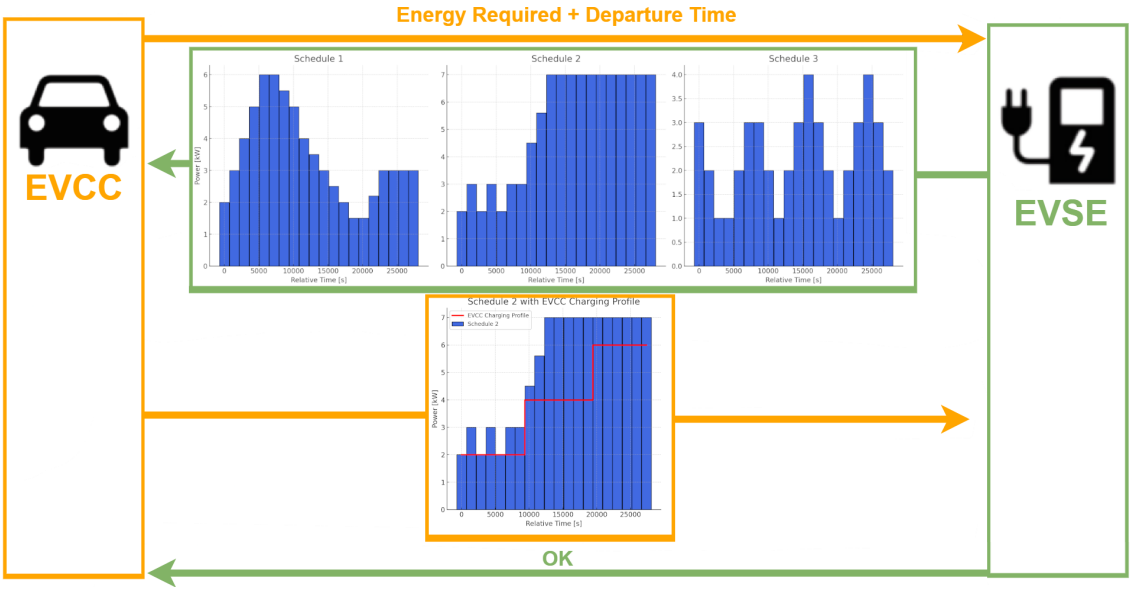


Figure 2.18: Example of the communication flow and negotiation process between EVCC and EVSE during schedule exchange for controlled charging.

process of reducing electricity consumption during periods of high demand by using stored energy or shifting usage to off-peak times, helping to prevent grid overload and lower energy costs.

2.3 Backend Communication: OCPP Considerations

During this chapter, all communication with higher-level entities, such as the CPO was facilitated using the Open Charge Point Protocol (OCPP). The complete structure and its communication standards are depicted in Figure 2.19.

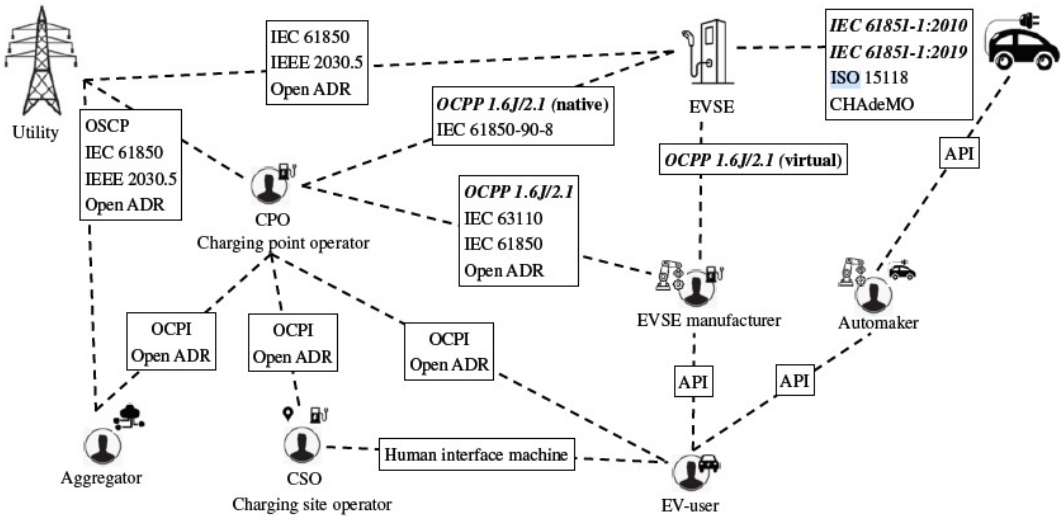


Figure 2.19: Communication standards for delivering grid services. From [23]

This communication, in the context of ISO 15118 is used for tasks such as certificate validation (see Section 2.1.3.4) and receiving updated specifications or limitations from the central server, belonging to the EVSE manufacturer, which can then, per example, influence the V2G messages exchanged with the EV (see Section 2.1.3.7). It enables the manufacturer not only to update certificates and private keys but also to deliver software updates to the EVSE for maintenance or any feature enhancement. The overall charging ecosystem involves multiple agents working in unison to ensure all processes run smoothly, including smart charging. The complete structure and its communication standards, like OCPP for certain nodes, are depicted in Figure 2.19.

2.4 ISO 15118 in EV market

The ISO 15118-2 protocol was released over a decade ago but has only recently begun gaining broader traction among car manufacturers. Early adopters such as Volkswagen, with its ID series, and certain BMW models implemented ISO 15118-2 as early as the 2020s. In contrast, ISO 15118-20, being a more recent and significantly more sophisticated evolution of the protocol, has only started to appear in commercially available vehicles within the last couple of years. Table 2.5 presents a list of vehicle models, their corresponding release years, support for either ISO 15118-2 or ISO 15118-20 and battery capabilities.

Table 2.5: Adoption of ISO 15118 Standards in Commercial Electric Vehicles. **Ann.** indicates functionality announced but not yet active. **Lmt.** indicates limited implementation or partial support. Data from [24].

Release Year	Vehicle Model	Standard	Charge Power (Max)		Battery		PnC	BPT
			AC (kW)	DC (kW)	Voltage (V)	Capacity (kWh)		
2020	Volkswagen ID.3 Pro	ISO 15118-2	11	124	400	58.0	Yes	No
2021	BMW i4 eDrive40	ISO 15118-2	11	207	400	80.7	No	No
2023	BMW i5 edrive40	ISO 15118-2	11	206	400	81.2	No	No
2023	Hyundai Kona Electric	ISO 15118-2	11	105	400	65.4	Yes	No
2023	Mercedes-Benz EQB 250+	ISO 15118-2	11	102	400	70.5	Yes	No
2023	Volkswagen ID.4 Pro	ISO 15118-2	11	175	400	77.0	Yes	No
2023	Volkswagen ID.7 Pro	ISO 15118-2	11	195	400	77.0	Yes	No
2023	Volvo EX30 Single Motor ER	ISO 15118-2	11	158	400	65.0	Lmt.	No
2024	BMW i4 eDrive40	ISO 15118-20	11	207	400	81.3	Yes	No
2024	Hyundai INSTER Long Range	ISO 15118-2	11	85	400	46.0	Yes	No
2024	Hyundai IONIQ 5 84	ISO 15118-20	11	263	800	80.0	Ann.	V2L
2024	Kia EV3 Standard Range	ISO 15118-2	11	100	400	55.0	Yes	V2L
2024	Kia EV3 Long Range	ISO 15118-2	11	100	400	78.0	Yes	V2L
2024	Mercedes-Benz EQS 450+	ISO 15118-2	11	200	400	118.0	Yes	No
2024	Renault 5 E-Tech	ISO 15118-20	11	100	400	52.0	Ann.	V2LHG
2024	Skoda Elroq 60	ISO 15118-2	11	165	400	59.0	Yes	No
2024	Skoda Elroq 85	ISO 15118-2	11	175	400	77.0	Yes	No
2025	Hyundai IONIQ 5 RWD	ISO 15118-2	11	263	800	80.0	Ann.	V2L
2025	Mercedes-Benz CLA 250+	ISO 15118-2	11	353	800	85.0	Yes	No
2025	Renault 4 E-Tech	ISO 15118-20	11	100	400	52.0	Ann.	V2LHG

This overview shows the slow but steady adoption of the ISO 15118 standard across the automotive industry, as well as the evolution of battery and charging technologies. AC charging is limited by the onboard converter, which is why most vehicles listed support a maximum of 11 kW

for AC charging. In contrast, DC charging utilizes the charging station's converter, allowing significantly higher charging powers. This results in shorter charging sessions and an improved user experience.

It is important to note that only a small number of vehicles currently support the more recent ISO 15118-20 protocol. Among these, not all have implemented bidirectional power transfer (BPT). Some models support only basic unidirectional charging despite announcing compatibility with more advanced protocols. Similarly, Plug and Charge (PnC)—a key feature of ISO 15118—is not consistently available. Some vehicles do not support it at all, while others advertise it but do not yet offer it in practice. This inconsistency suggests that manufacturers intend to unlock these capabilities via future software updates.

It is also possible for the same vehicle model to support different ISO 15118 protocol versions depending on the release year. A relevant example is the BMW i4, which was initially released with ISO 15118-2 support and more recently appeared in a variant supporting ISO 15118-20.

Interestingly, some vehicles that only advertise ISO 15118-2 support still implement BPT use cases such as Vehicle-to-Load (V2L). According to [11], these capabilities require enhancements to the standard application-layer message set defined in the protocol. This implies that certain implementations include proprietary or vendor-specific extensions to support external loads, even if they are not fully compliant with the ISO 15118-20 message definitions for energy transfer. In contrast, vehicles that natively support ISO 15118-20 and advertise BPT support not just V2L, but also more complex use cases such as Vehicle-to-Home (V2H) and Vehicle-to-Grid (V2G), expanding their role within a smart energy ecosystem.

2.5 Summary

This chapter provided an overview of the theoretical aspects of the standards that support the electric vehicle industry, highlighting their evolution down to each individual OSI layer. The main components and key actors of the charging infrastructure were discussed, along with the various charging methods, all of which are crucial to the subsequent work to be carried out.

It is clear that the protocol discussed was only released recently, in 2022, and its study is still ongoing. Several potential weaknesses and vulnerabilities of the protocol were identified. Some of the vulnerabilities presented have possible solutions that can be implemented on top of ISO 15118, though further research may be needed to explore alternative approaches.

From the study of the protocols, it can be concluded that the smart charging of vehicles depends on the proper functioning of several entities, with the charging station requiring extensive programming and testing to ensure it adheres to industry standards and communication techniques correctly. Even tho some studies have been made, further development of the charging infrastructure is crucial to ensure that the final solution is as efficient, robust, and scalable as possible.

Chapter 3

System Architecture, Implementation and Methodology

This chapter provides a comprehensive overview of the proposed development setup designed to validate next-generation electric vehicle communication technologies. The system consists of a charging station equipped with ISO 15118-20 support, enabling features such as PnC and BPT, alongside a complementary electric vehicle simulator developed for testing purposes. Together, these components form a controlled testbench aimed at assessing both the functional behavior and security robustness of the communication protocol.

3.1 Methodology Overview

This section summarizes the methodology adopted throughout the development and validation of the ISO 15118 compliant EVSE. The approach was structured to enable both simulation based and real-world testing, ensuring deployment readiness. The methodology can be broken down into the following key components:

- **Equipment and tools used:**
  - **Simulator:** *Trialog's ComboCS EV Simulator* [25], used to emulate EV-side behavior and perform structured charging sessions under ISO 15118 and DIN SPEC 70121 protocols.
  - **EVSE:** INESC TEC's charging station based on a Raspberry Pi 3B [26] as the computational unit, integrated with ChargeByte's PLC Stamp Mini 2 module [27], energy metering, relays, and PWM signaling circuits.
  - **Open-source software:** Adapted stacks from EcoG's `iso15118` [28] and `pyslac` [29] repositories, extended and integrated into the INESC TEC EVSE software framework.
  - **Packet analysis tools:** *Wireshark* [30] with the *dsV2Gshark* plugin [31], used for monitoring, dissecting, and validating SLAC and V2GTP message flows.

- **Experimental workflow:**
  - Development and configuration of a simulation environment using Trialog's ComboCS EV simulator.
  - Design and upgrade of a pre-existing IEC 61851 EVSE to support ISO 15118-2 and ISO 15118-20 communication.
  - Integration and adaptation of open-source architectures for HLC.
  - Execution of structured test sessions in the simulation environment using various configurations with systematic data collection for each scenario.
  - Deployment and validation of the developed solution in a real-world charging infrastructure, with execution of live charging sessions and collection of operational data for later analysis.
- **Data acquisition and analysis:**
  - Capturing timing and message flow data across ISO 15118 sessions with diferent configurations such as protocol version (-2 or -20), security level (no TLS, TLS1.2 or TLS1.3) and authorization method (EIM or PnC).
  - Comparison of simulation and real-world data to assess communication performance and compatibility.
  - Logging and classifying EVSE behaviors based on communication protocol outcomes.
- **Limitations and assumptions:**
  - The absence of real production certificates limited the validation of features such as Plug and Charge (PnC) and Transport Layer Security (TLS) to test environments using simulated credentials.
  - ISO 15118-20 functionalities were only tested under simulated conditions, using the provided certificate sets from Trialog, and not validated against physical EVs.
  - All testing was restricted to AC charging scenarios due to the EVSE hardware supporting only single-phase AC operation. No DC-specific messages or flows were implemented.
  - The charging schedules used were precomputed and fixed, without dynamic local calculation based on EV-provided parameters. Real-time schedule optimization lies outside the scope of this work.
  - Integration with higher-level management systems (CSMS) was limited. No new messages or control flows were introduced, as this would require adaptations in the CSMS. Furthermore, integration with even higher-level grid controllers was not possible. Such integration would be essential for enabling grid smart charging strategies, real-time power balancing, and energy market participation.

This structured methodology allowed for a progressive validation of ISO 15118 features and ensured that the final system could be evaluated both in isolated test conditions and within a real deployment context.

3.2 Trialog's ComboCS

To test the development of the charger, a simulator provided by INESC TEC from Trialog, EV ComboCS, will be used. This device, shown in Figure 3.1, allows us to emulate the electric vehicle (EV) side of a Combined Charging System (CCS) implementation in order to perform or simulate charging sessions compliant with IEC 61851-1, ISO 15118, or even DIN SPEC 70121, the last being a protocol from the german institute of standardization that will not be covered in depth in this dissertation.

Figure 3.1: ComboCS, the EV CCS Simulator. From [25]

3.2.1 ComboCS UI

The ComboCS simulator provides configurability over various ISO 15118 related parameters, enabling testing under multiple scenarios. It supports the selection of communication modes, including standard TCP or TCP with TLS encryption, as well as authentication methods such as External Identification Means (EIM) and Plug and Charge (PnC). Additionally, it allows the simulation of realistic charging conditions by specifying parameters such as the vehicle's departure time, requested energy amount and charging schedule preferences. Additionally, it includes several predefined test cases that assist in the development and validation of EVSE implementations. Most of these features can be easily configured through the simulator's web interface, as shown in Figure 3.2.

3.2.2 API Controller

While Trialog's graphical web interface offers a convenient way to configure and trigger simulations, it falls short when it comes to testing more advanced, dynamic behaviors. For instance, it does not support real-time interaction during a charging session, such as modifying battery parameters, simulating mid-session user actions, or triggering renegotiations, which are essential to test

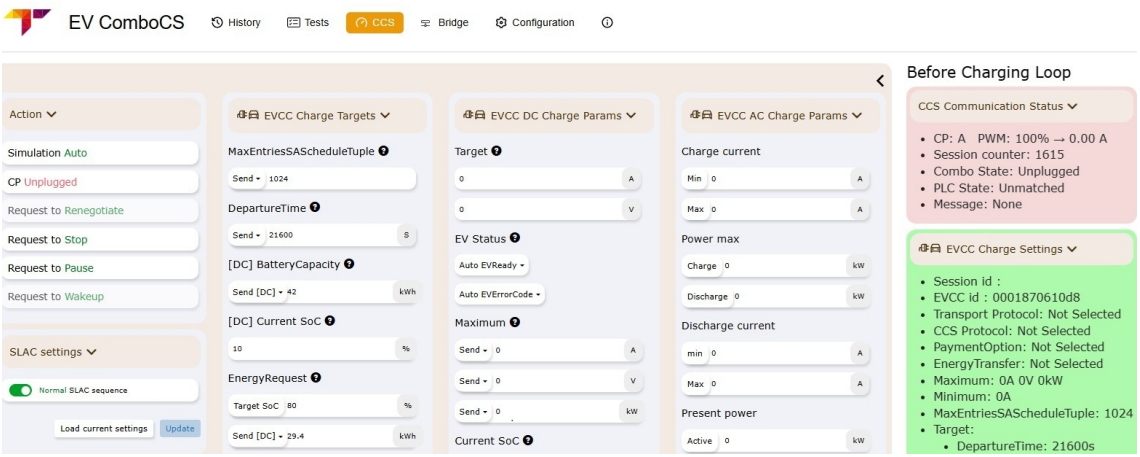


Figure 3.2: Snippet of ComboCS web interface.

the full flexibility of ISO 15118. To overcome these limitations, the ComboCS simulator provides a REST API that enables external tools to interact with its internal state programmatically. This API supports standard HTTP methods (GET, POST, PUT), which can be used to modify parameters such as battery state of charge (SoC), remaining energy, charging targets, and schedule-related flags during an ongoing session.

To facilitate this interaction, a lightweight Python controller class was developed which wraps the API endpoints in an object oriented structure, making it easier to run automated test cases and inject parameter changes at runtime. It allows dynamic behavior to be simulated in the EVCC, mimicking real-world battery behavior as charging progresses. The core attributes and methods of the controller, focused mainly on AC charging, are illustrated in Figure 3.3.

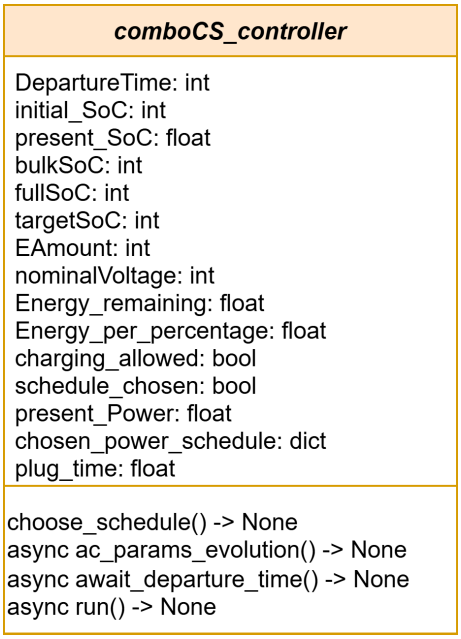


Figure 3.3: Class diagram summarizing the main attributes and methods used of the ComboCS API controller Python class.

The developed code interacts with the simulator's automatic operation by injecting information during its execution so that the EV characteristics actually evolve during charging, something that would not occur in normal operation. This is important for later verifying meter receipts, energy consumption, compliance with the schedule and other aspects. In ISO 15118-20, which enables dynamic control of charging, these parameters, especially the SoC, are also crucial to be exchanged between the EV and EVSE. The overall functionality of the system is illustrated in the flowchart shown in Figure 3.4.

For testing purposes, when the schedule finishes, the EV immediately requests to stop charging, which places both the EV and EVSE into idle mode. The only manual inputs required for the algorithm to function are the start input, which configures the EV with the desired characteristics and simulates the plug-in, and the end input, which simulates the plug-out. Many parameters can be configured on the EV as stated in 3.2.1, but the most important ones for our use are:

- Communication Protocol Used (ISO 15118-2, ISO 15118-20)
- Transport Protocol (TCP, TLS)
- Payment Options (PnC, EIM)
- Initial/Target/Full State of Charge
- Requested Energy Amount
- Departure Time

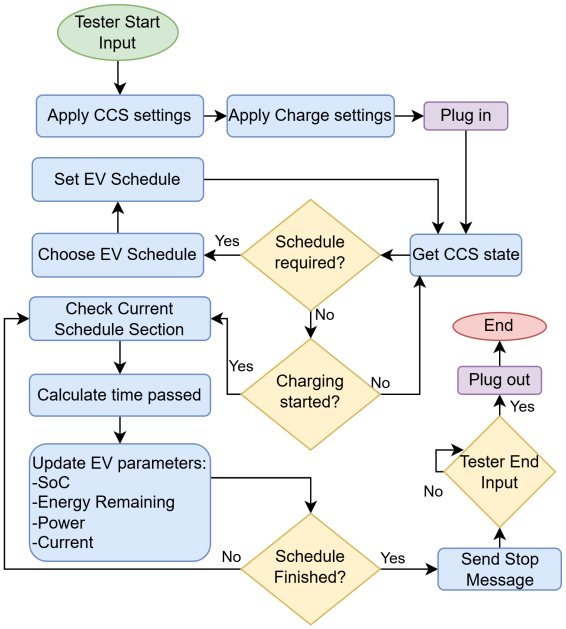


Figure 3.4: Flowchart of the ComboCS API controller Python class operation.

3.3 EVSE compliant with ISO 15118

This section outlines the design and implementation of an EVSE system that complies with the ISO 15118 protocol. It covers the essential hardware components, their integration, and the supporting software architecture necessary to enable the communication and control between the EVSE and electric vehicles.

3.3.1 Hardware Characteristics

In this section the focus is on the physical hardware that forms the backbone of the EVSE system. At the center of the setup is a previously developed IEC 61851 compliant charger, a single-phase AC unit capable of delivering up to 7360 W, designed and implemented at SGEVL INESC TEC. This charger has demonstrated robustness and reliability, with 12 units, with a prototype shown in Figure 3.5, currently deployed across INESC TEC's headquarters premises.

Figure 3.5: INESC TEC's previous IEC 61851-compliant charger.

As illustrated in the schematic in Figure 3.6, the computational module interfaces with several key peripherals, including the PWM generator, PLC module, relay, and energy meter—each responsible for a distinct function within the charging process. Among these, the PLC module is the only hardware component added to the preexisting IEC 61851-compliant charger in order to enable ISO 15118 communication. Consequently, the primary development focus lies in the interaction between the computational module and the PLC, as they constitute the core of the system's new functionality.

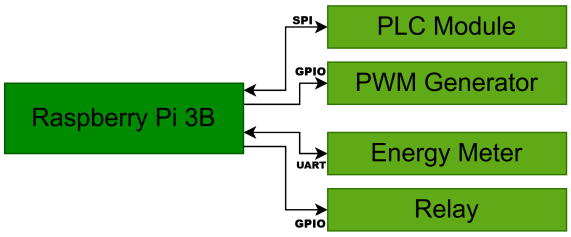


Figure 3.6: Block diagram of the EVSE's hardware components.

3.3.1.1 Computational Module

At the heart of the charger lies its computational module, responsible for processing data from both the charging control and energy metering modules. This includes the interpretation of the control pilot (CP) signal and energy measurements for various operational and monitoring tasks. For this purpose, a single-board computer is employe, *Raspberry Pi 3B*, widely recognized for its reliability in both research and industrial environments will be used. It can be seen in Figure 3.7.

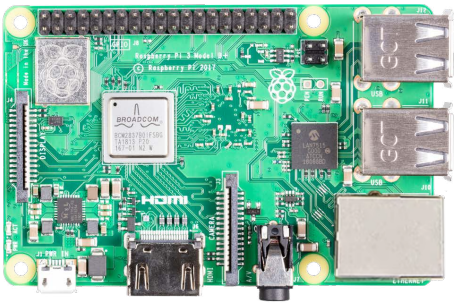


Figure 3.7: Raspberry Pi 3B unit used in the EVSE controller. From [26]

The Raspberry Pi serves as the central controller of the EVSE system, coordinating both communication and hardware operations. It can directly control the relay to connect or disconnect the power supply to the vehicle, generate and adjust the CP duty cycle and read energy measurements from the metering hardware. Additionally, as will be discussed in the following section, it interfaces with a PLC module enabling ISO 15118-based communication. Beyond these hardware tasks, the Raspberry Pi also manages any necessary local data storage and communication with external systems such as the Charging Station Management System (CSMS). Since the ISO 15118 protocol demands higher communication precision and increased data exchange compared to IEC 61851, the computational module is expected to operate under greater processing load. For direct development and maintenance, the Raspberry Pi is accessed remotely using Secure Shell (SSH), a network protocol that provides a secure way to access and control a device over any network, including unsecured ones. This enables safe remote monitoring, configuration, and system updates without the need for physical access. A summary of the Raspberry Pi 3B's hardware specifications is presented in Table 3.1.

Table 3.1: Computational Specifications of the EVSE [26]

Aspect	Specification
Model	Raspberry Pi 3B
CPU Clock	Quad-core 1.2 GHz
RAM	1 GB
Memory Card (ROM)	32 GB Ultra-Rapid SD Card
Operating System	Raspbian OS 32-bit
Processor Architecture	ARMv7

3.3.1.2 PLC Module

The main requirement for enabling communication under the ISO 15118 protocol is the ability to inject digital signals onto the CP line. To achieve this, a PLC module, *PLC Stamp Mini 2* from *ChargeByte*, will be employed. This module integrates the *QCA7000* chipset developed by *Qualcomm*, which supports communication based on the *HomePlug Green PHY* standard. This standard ensures compatibility with a wide range of commercial EVSE and EV components. A high-level overview of the module is shown in Figure 3.8.

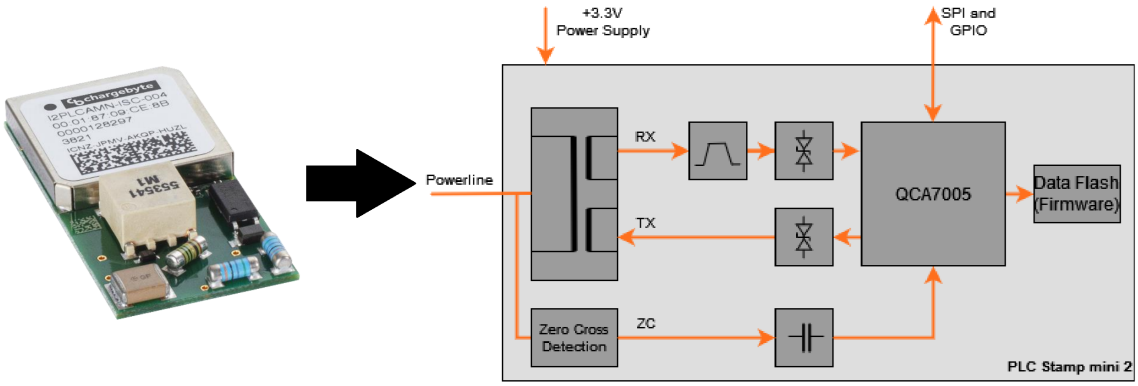


Figure 3.8: Image of the PLC Stamp Mini 2 Module and its Block Diagram. From [27]

When using the chip in an EVSE, a zero-cross detection circuit is required to identify the precise moments when the AC waveform crosses the zero-voltage level [27]. This allows PLC communication to align with the AC phase, effectively using the mains frequency as a timing reference. Such synchronization is essential for supporting coexistence functionality [27] and avoiding transmission collisions in powerline communication. The circuit utilized and recommended in the datasheet is shown in Figure 3.9.

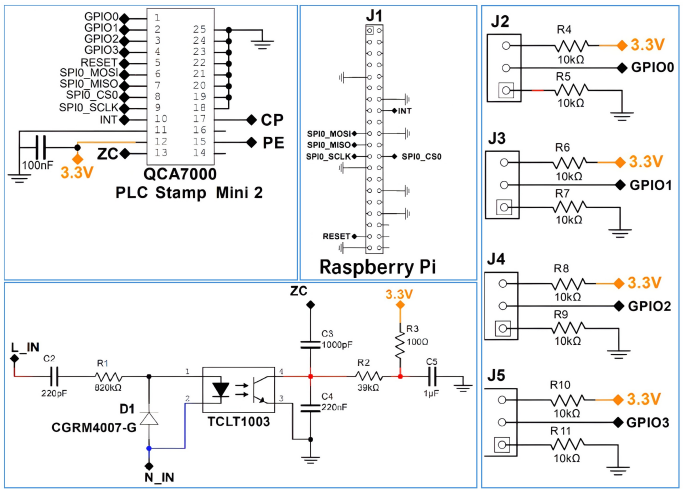


Figure 3.9: Schematic of the Raspberry Pi connections to the QCA7000, zero-cross detection circuit and required jumpers.

The QCA7000 communicates with the Raspberry Pi via Serial Peripheral Interface (SPI), with the interface connections defined in the schematic shown in Figure 3.9. A key advantage of using this chip is that the Raspberry Pi already provides a Device Tree Overlay (DTO), specifically ‘dtoverlay=qca7000‘, which allows the kernel to recognize and configure the QCA7000 as a standard Ethernet interface. This integration enables the use of the HomePlug Green PHY protocol by transmitting Ethernet frames over the CP line [27].

The chipset provides four additional GPIOs and one reset pin. The reset pin is controlled by a Raspberry Pi GPIO and can be triggered via software, making it useful for tasks such as flushing the QCA7000's internal memory during debugging operations. The four GPIOs are connected physically to configurable jumpers, as shown in Figure 3.9, and are not intended to be changed during runtime. These GPIOs must be correctly set at startup to ensure proper device initialization. Specifically, for SPI slave operation, the required configuration is pull-up on GPIO 0, pull-down on GPIO 1, pull-up on GPIO 2 and pull-up (any state) on GPIO 3. These requirements follow the configuration guidelines summarized in Table 3.2.

Table 3.2: GPIO Configuration Modes for QCA7000. Adapted from [27]

GPIO #	Function	Pull Up	Pull Down
0	Boot Source	Flash	Host
1	Host Interface	-	SPI Slave
2	SPI Slave Mode	Burst	Legacy
3	None	-	-

3.3.2 Firmware Development

After completing the necessary hardware integration the software implementation on the microprocessor can begin. This will be the most time-consuming part of the dissertation, as the integration between certain software modules is expected to be challenging, with some bugs and failures needing to be resolved.

3.3.2.1 PLC Configuration

In order for the PLC module to work with the Raspberry Pi 3, several considerations must be taken into account. Once paired with the Raspberry Pi via SPI, it creates an Ethernet interface, as detailed in 3.3.1.2. To facilitate consistent deployment and integration, a Bash script was created to configure the Raspberry Pi accordingly as seen in Listing 3.1, after which the device must be rebooted. Additionally, a `.link` file is created to ensure the QCA7000 interface is persistently named `qca0`, avoiding unpredictable naming issues (e.g., the interface being inconsistently assigned as `eth0` or `eth1`).

Listing 3.1: Bash script for enabling SPI and QCA7000 overlay on Raspberry Pi

```
sudo sed -i '/^dtparam=spi=on/d' /boot/firmware/config.txt
echo "dtparam=spi=on" | sudo tee -a /boot/config.txt

sudo raspi-config nonint do_spi 0

sudo sed -i '/^dtoverlay=qca7000/d' /boot/firmware/config.txt
echo "dtoverlay=qca7000" | sudo tee -a /boot/firmware/config.txt

sudo tee /etc/systemd/network/10-qca.link > /dev/null <<EOF
[Match]
Driver=qcaspi
[Link]
Name=qca0
EOF

sudo sed -i 's/\\s*net.ifnames=[^ ]*//g' /boot/cmdline.txt
sudo sed -i 's/\$/ net.ifnames=0/' /boot/cmdline.txt
```

An issue arose during development in which the IPv6 address associated with this interface would intermittently disappear, disrupting the ISO 15118 OSI layer 3 communication flow as in Chapter 2.1.3.3. The root cause of the issue was not identified yet, but is suspected to be related to the QCA driver on the Raspberry Pi. Several fixes were attempted, but the only definitive solution was to assign a static IPv6 address to the interface. The configuration shown in Listing 3.2 generates a unique random IPv6 address derived from the Raspberry Pi's serial number, guaranteeing its uniqueness. Note that this configuration requires the `ifupdown` package to be installed on the system. After applying the configuration, the device should be again rebooted.

Listing 3.2: Bash Script to allow static IPv6 configuration for the QCA7000 interface on Raspberry Pi

```
sudo apt install ifupdown -y

sudo tee -a /etc/network/interfaces > /dev/null <<EOL
# Configuration for QCA7000
auto qca0
iface qca0 inet6 static
    address FIXED_ADDRESS
    netmask 64
    accept_ra 0
    autoconf 0
EOL
```

3.3.2.2 Implementation

The implementation focuses on layering ISO 15118 functionalities over the existing control architecture. This development involves adapting open-source libraries, integrating PLC communication, and ensuring seamless interaction between the software modules and hardware interfaces. Emphasis is placed on modularity and protocol compliance, laying the groundwork for future enhancements and interoperability testing.

The software for the IEC 61851 charger, developed by INESC TEC, has been analyzed to facilitate integration with the ISO 15118 codebase and its original main architecture can be seen in the digram of Figure 3.10. The core framework is implemented in Python, which serves as the main integration point. Peripheral classes, which interact directly with hardware components as in Figure 3.6, such as the energy meter (used for reading energy drawn from or injected into the grid) and the basic charging module (responsible for managing the PWM duty cycle and controlling the relay), are written in C for faster performance and direct hardware access, which later interface with the core module through Python wrappers. However, this part of the implementation lies largely outside the scope of this dissertation, even though it is the base of the ISO 15118 protocol, and constitutes intellectual property of INESC TEC. As such, most details are not disclosed, and only minor modifications were necessary for integration purposes.

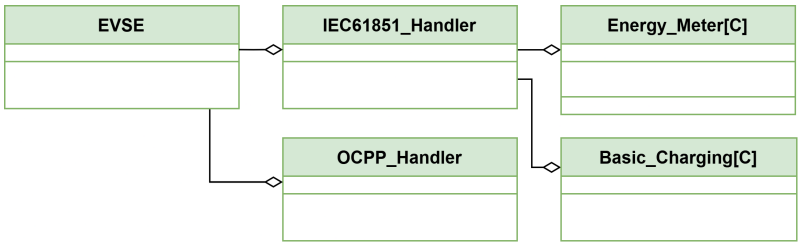


Figure 3.10: Original class diagram of the charger implementation prior to ISO 15118 integration. Class names annotated with **[C]** indicate classes implemented in the C language; all others are implemented in Python.

The ISO 15118 protocol is a highly detailed and layered communication standard, organized according to the OSI model, as discussed in Chapter 2.1.3. To simplify the development effort, the open-source EcoG stack was selected as the foundation for the ISO 15118 implementation [29][28]. Its Python-based architecture allows for seamless integration with the INESC TEC EVSE framework. The entire software system follows an object oriented approach, and its updated structure is illustrated in the class diagram shown in Figure 3.11.

Additionally, the diagram outlines the overall architecture of the developed code, with the green classes being developed originally by INESC TEC. Among these, the only significant modification was made to the Basic Charging module, enabling it to fix the duty cycle at 5% when HLC is active. Additionally, the Basic Charging module allows the F state to be enforced, which is crucial for resetting communications, particularly during SLAC retries, as discussed in Chapter 2.1.3.2. The blue classes originate from open-source repositories that were incorporated into

our framework, while the orange classes were entirely developed from scratch to ensure proper integration between the two codebases.

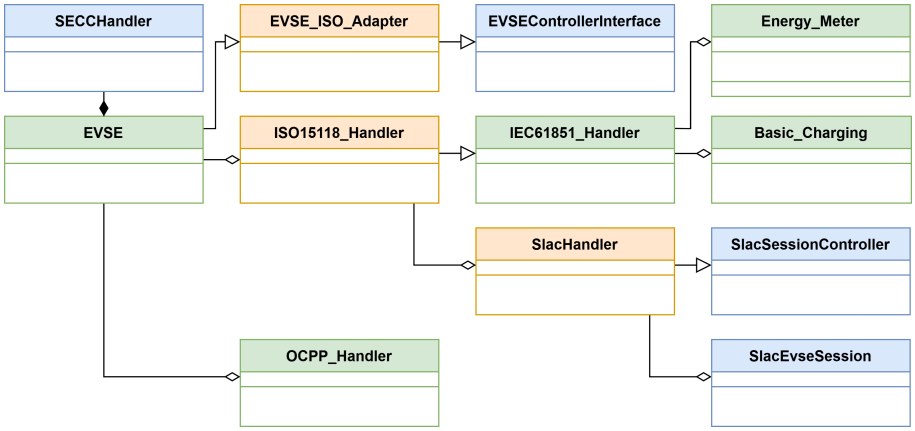


Figure 3.11: Class diagram of the ISO 15118 EVSE implementation. Green classes were developed by INESC TEC, blue classes originate from the EcoG codebase, and orange classes were developed specifically to interface between the two systems.

3.3.2.2.1 SLAC

One of the open-source repositories used is *pyslac* [29], which implements the SLAC process required by the ISO 15118 protocol. This module is responsible for retrieving attenuation parameters that are later used during the primary PLC communication with the EV. The code from this repository required significant modifications, as the original implementation was largely a mockup and remains incomplete in several areas. Additionally, EcoG has not actively maintained the *pyslac* repository focusing primarily on the *iso15118* module. The key modifications made to the pyslac internal code were crucial to the working of the code, and were as follows:

- Improved logging to provide better insights into the SLAC session flow, especially for session creation, matching, and errors.
- Added more informative log messages for key events such as SLAC session reset, matching progress, and set key procedure.
- Enhanced error handling to ensure better exception tracking and handling.
- Replaced hardcoded timeout for better configurability.
- Added missing logic to handle link health checks and transitions for the EVSE and EV matching state.
- Fixed issues in the SLAC protocl exit by resetting states and leaving the logical network gracefully.

- Refined logic for retrying SLAC matching on failure, with improved handling for retries and failure messages.

The class diagram in Figure 3.12 serves as a reference map for the development process, highlighting the components that were modified and their interface with the rest of the EVSE architecture. Other minor support classes, primarily related to message framing and construction, are not included in the diagram as they performed as intended and required no modification.

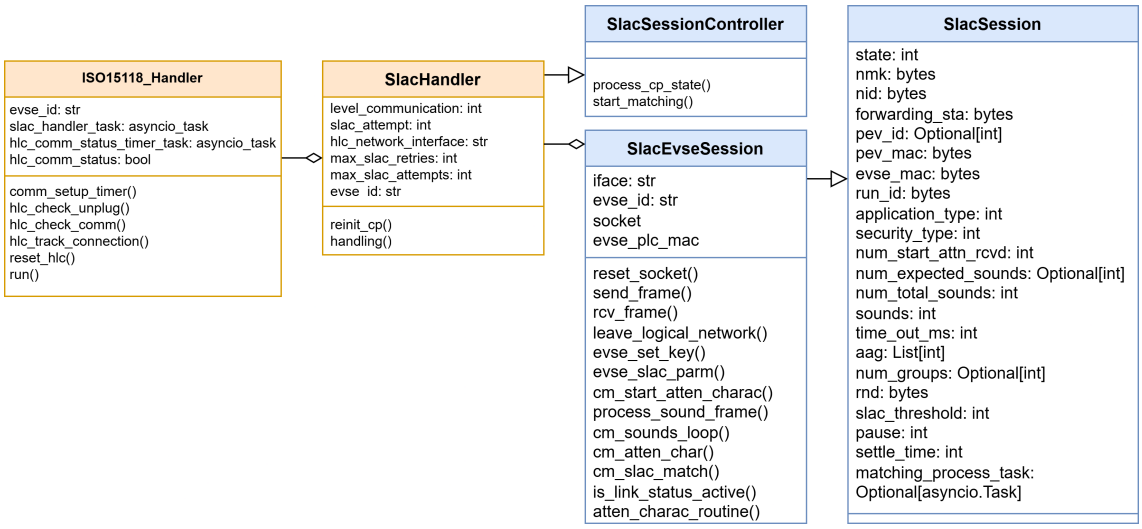


Figure 3.12: Class diagram highlighting the main *pyslac* classes involved in SLAC communication and their integration with the EVSE controller. Based on the source code from [29].

An additional class, *SlacHandler*, was developed to interface the *pyslac*[29] module with the broader EVSE framework. This class is primarily responsible for monitoring the CP state, managing its duty cycle, tracking the number of SLAC attempts, and transitioning the CP to state F between retries. The *ISO15118_Handler* class manages the *SlacHandler* object and also monitors the state of the HLC session, if one is initiated. If a V2G Communication Session is not started with an SDP request (see Section 2.1.3.5), the system gracefully falls back to LLC communication.

The interaction between the `SlacHandler` and `ISO15118_Handler` classes is illustrated in Figure 3.13. In this diagram, the box labeled as "process SLAC messages" represents the main method adapted from the *pyslac* [29] codebase, which parses the received packets into their respective categories and expected sequence, as defined in Section 2.1.3.2, while also verifying their integrity.

It is important to highlight that the SLAC procedure may fail even before the global SLAC timeout is reached. In such cases, the session is marked as unmatched, which causes the CP state to immediately reset. These failures may derive from internal processing issues during message reception (see Figure 2.9) or from message-specific timeouts. For instance, although the attenuation characterization phase may complete successfully, subsequent stages such as the sound exchange may experience timeouts, causing the SLAC to abort prematurely.

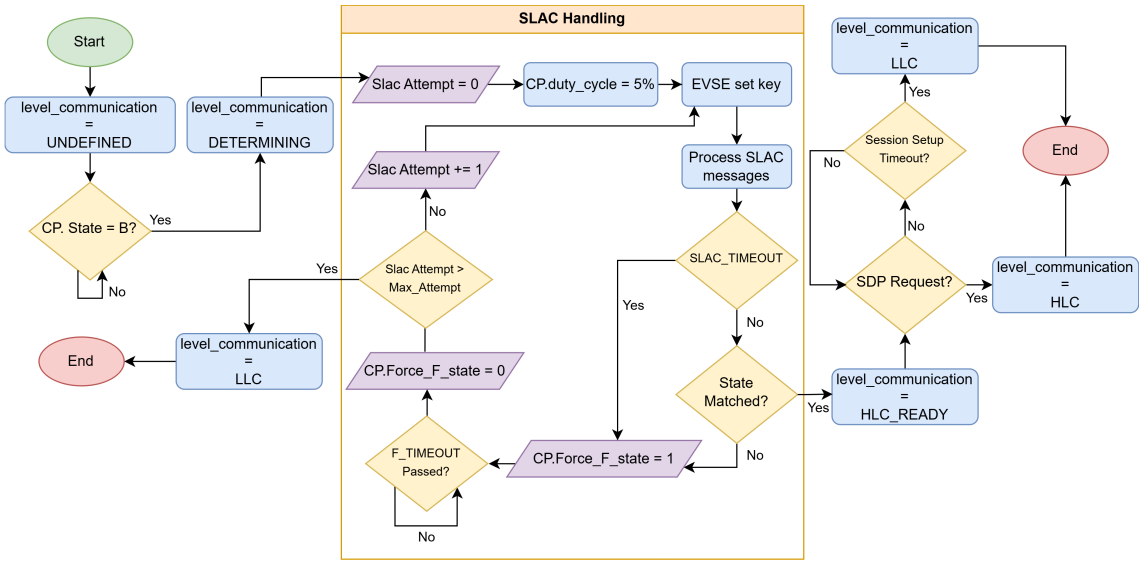


Figure 3.13: Execution flow of the SLAC matching process, including retry logic and fallbacks to LLC. Based on the source code from [28].

If all SLAC attempts are exhausted without establishing a valid match, the system falls back to LLC-based communication and proceeds with basic charging according to INESC TEC's implementation. Moreover, if the SLAC sequence is successful, the `ISO15118_Handler` waits for the reception of the SDP Request by the EVSE. Failure to receive this request within the expected timeframe will also result in the session reverting to LLC communication, as explained in Section 2.1.3.4.

3.3.2.2.2 ISO 15118 Stack

The other repository from EcoG that was integrated into the system is *iso15118* [28], which implements the ISO 15118 protocol stack covering OSI layers 3 to 7, as described in Sections 2.1.3.3 to 2.1.3.7. Similar to the *pyslac* repository, it follows an object oriented logic, however, its structure is significantly more complex and extensive, comprising over 70 classes in total.

Figure 3.14 illustrates the main classes of this infrastructure. For clarity, the majority of smaller classes, primarily responsible for message categorization and parsing, are omitted from the diagram. The `SECCHandler`, which acts as initializer for the `CommunicationSessionHandler`, integrates the EVSE component to enable communication both upstream (towards higher-level entities via OCPP) and downstream (towards power control modules and energy meters).

In order to enable this integration, the previous EVSE class developed by INESC TEC must be adapted so it can be passed as an argument to the `SECCHandler`, as depicted in Figure 3.11. In order to preserve the original EVSE implementation developed by INESC TEC and avoid direct modifications, an adapter class named `EVSE_ISO_Adapter` was created. In this way, the adapter inherits from `EVSEControllerInterface`, which defines the interface that an EVSE

controller must implement to interact with the ISO 15118 stack. In the context of object oriented programming, an interface class specifies a set of methods and behaviors that implementing classes must provide and return, without forcing their actual internal logic, thus ensuring flexibility and modular integration.

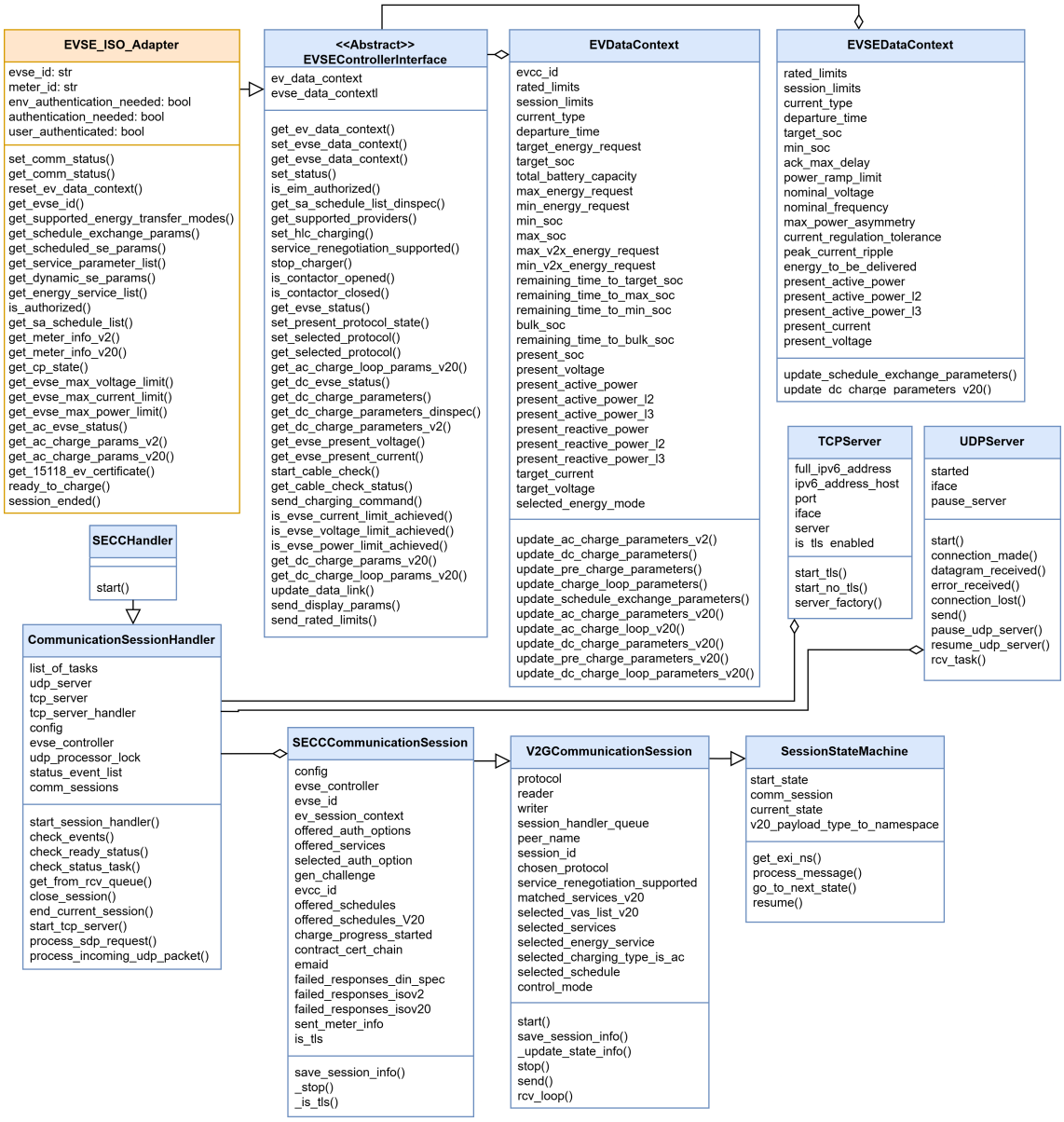


Figure 3.14: Class diagram illustrating the main components of the *iso15118* stack, highlighting its core architecture. Based on the source code from [28].

As seen in Figure 3.14, the main session handler contains a `SECCCommunicationSession` component, which may support multiple concurrent sessions. However, for this particular EVSE configuration, which has one processing unit per charging cable, there will only be one active communication session at a time. The handler also includes a `UDPServer` instance responsible

for handling the initial SDP messages. This UDP server remains active throughout the operation of the charger, but it is temporarily paused once a TCP server is instantiated to handle a connection. The `TCPServer` instance is created for the duration of a charging session and is closed once the session ends, after which the UDP server is resumed to listen for new SDP requests. The main operation of the ISO 15118 stack follows a state machine architecture, and the overall flow of the communication process is illustrated in Figure 3.15.

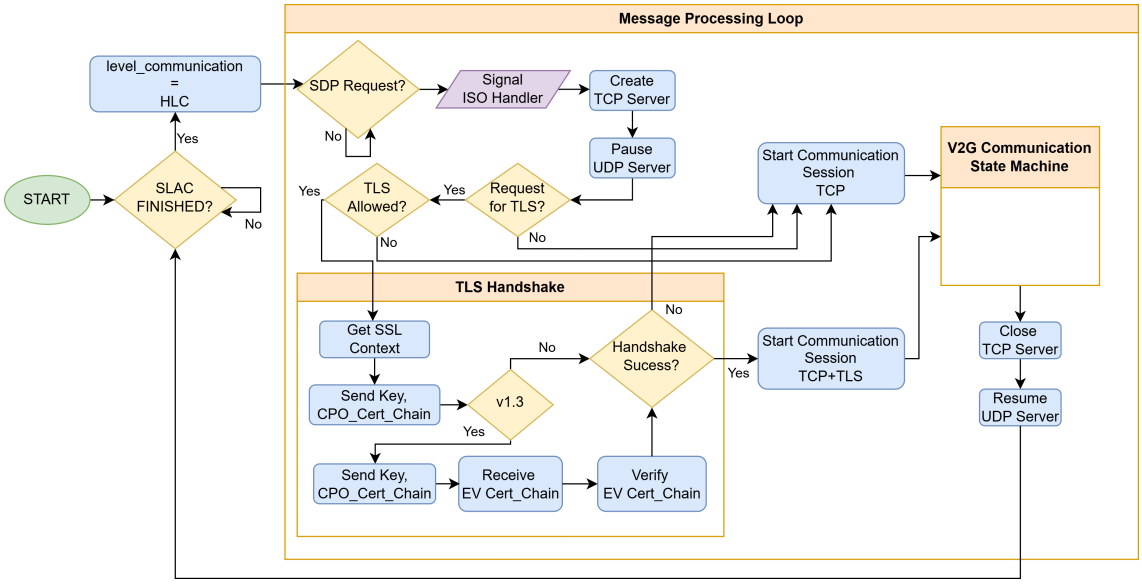


Figure 3.15: Communication flow in the ISO 15118 stack. Based on the source code from [28].

The simplified flow diagram illustrates the sequence of decisions performed by the SECC prior to entering the V2G Communication State Machine. The ISO 15118 communication stack operates asynchronously from the preceding basic charging and SLAC modules. Initially, the UDP server continuously listens for an incoming SDP request message. Upon its reception, the ISO Handler, described in the previous section, must be notified, as there is a maximum allowable timeout between the successful completion of SLAC and the initiation of the SDP exchange, as discussed in Section 2.1.3.4.

Following this, the EVSE compares the security level requested by the EV with its own settings. The system was modified to both enforce or disable TLS, enabling the charger either to operate without a certificate infrastructure or to require secure communication. If TLS is selected, a TLS handshake is initiated, during which session keys and the CPO's certificate chain are exchanged with the EV. The TLS handshake shown in Figure 3.15 is simplified for clarity. It only reflects the decision path based on whether mutual authentication is required—i.e., whether the EV's certificate chain must also be validated, as is the case in TLS 1.3 with mutual authentication. When establishing the SSL context, the architecture fetches the necessary certificates from the certificate handler, as explained in Section 3.4.

The flowchart does not include the internal details of the V2G Communication State Machine, which governs the core message exchange between the EVCC and the EVSE. This exchange, primarily situated at OSI Layer 7, follows a well-defined sequence of protocol messages. While a detailed explanation of this process is provided in Section 2.1.3.7, a summary of the code's functionality is illustrated in Figure 3.16.

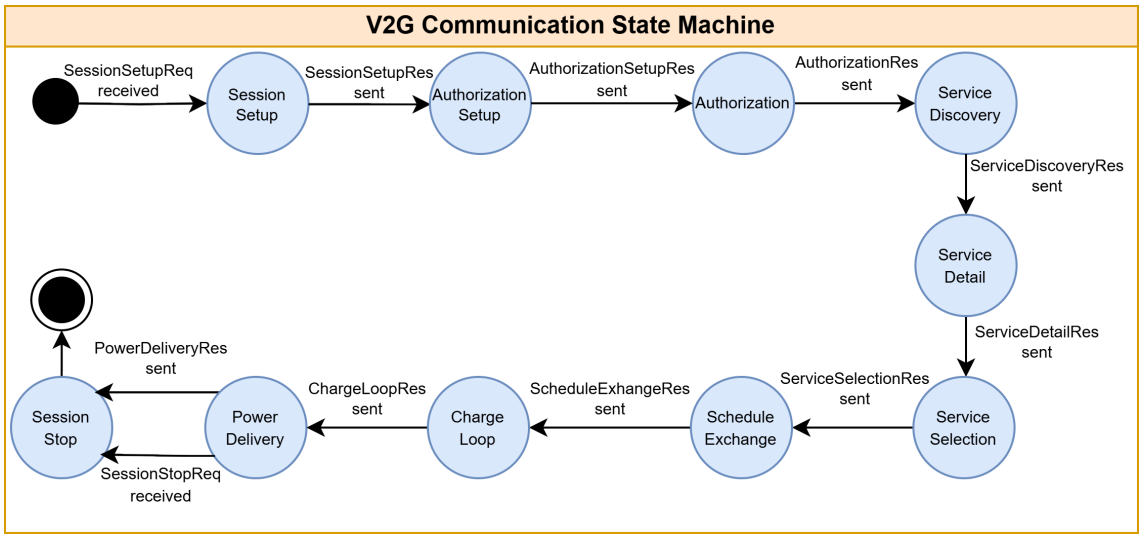


Figure 3.16: State sequence in the ISO 15118 V2G State Machine. Based on the source code from [28].

Within the state machine, several key states require particular attention. In the `Authorization` state, if External Identification Means (EIM) is selected, the EVSE must await the authorization token from INESC TEC's CSMS via OCPP.

In the `Schedule` state, if no valid local charging profile is available, or if the existing one does not satisfy the EV's charging requirements, a schedule negotiation should occur via OCPP. In this process, the EV communicates its charging needs, and an appropriate charging schedule is expected in return. However, due to current system limitations, negotiation with higher-level entities has not been implemented. Instead, a fixed, predefined schedule is provided to the EV. This is not considered a limitation, as charging schedule optimization and predictive algorithms lie outside the scope of this dissertation.

Finally, charging session termination can occur in two standard ways: either the EVSE actively sends a stop request, or it passively receives a `SessionStop` request initiated by the EV.

3.3.2.3 Monitoring Packet Flow Between EVSE and EV

In order to ensure full visibility into the packet and message exchange for both SLAC and ISO 15118 communication flow, it is critical to monitor the network traffic without any loss of packets. While internal logging in our codebase provides some visibility, they cannot capture every frame and are not always convenient for thorough analysis.

To achieve complete and reliable monitoring *Wireshark* [30] is employed, which is a network packet analyser capable of presenting captured packet data in as much detail as possible. This allows for precise inspection of communication between the EVSE and the EV. Since our computational platform, the Raspberry Pi(see Section 3.3.1.1), runs a non-graphical Operational System, Wireshark cannot run directly on it. Therefore, the packets captured on the Raspberry Pi's PLC Ethernet interface are forwarded over a secondary channel to a personal computer, where Wireshark runs with full graphical features. This redirection is implemented using a combination of two standard command-line tools:

- **tcpdump** — A packet capture utility that reads traffic from a specified network interface and outputs raw packet data using TCP or UDP.
- **netcat** — A networking utility used for reading and writing across network connections using TCP or UDP. In this context, it is used to forward the captured packet stream to the host PC.

With these tools installed on a Windows device, a listener is started on a certain port to receive the incoming stream and directly input it into Wireshark for analysis:

Listing 3.3: Listening and displaying live packets in Wireshark

```
nc -l -p %PORT% | wireshark -k -i -
```

Then, on the Raspberry, the following command is executed to capture and forward traffic from the PLC interface to the computer's IP address and designated port:

Listing 3.4: Forwarding packet stream from Raspberry Pi to PC

```
sudo tcpdump -i $NTWRK_INTERFACE -s 0 -U -w - | nc $IP $PORT
```

After completing this setup, it becomes possible to observe the live packet exchange between the EV and the EVSE. However, without protocol-specific decoding, these packets lack any meaningful description. Wireshark supports a wide range of network protocols and leverages plugins to provide detailed dissection of captured traffic. Nevertheless, it does not natively support the parsing of V2GTP (Vehicle-to-Grid Transport Protocol) messages (see Section 2.1.3.5).

To address this, the open-source plugin *dsV2Gshark*[31], developed by dSPACE, is employed. This plugin enables the decoding and visualization of V2GTP messages, making it possible to analyze the content of the V2G communication stack in Wireshark.

It is important to note that after the initial SDP handshake, subsequent messages can only be interpreted if TLS is not enabled. When TLS is active, message contents are, as expected, secured and encrypted (see Section 4), making them unreadable to the observer.

When live analysis is not a priority, but we still want to record the received packets for later inspection, a dedicated systemd service was created to locally store the captured traffic. This approach allows for continuous packet recording with automatic file rotation based on size.

Listing 3.5: Capturing and storing network traffic locally with automatic file rotation

```
PREFIX="$CAPTURE_DIR/capture_$(date␣+%Y-%m-%d_%H-%M-%S)"
tcpdump -i "$NTWRK_INTERFACE" -s 0 -C "$LIMIT_KB" -W 1 -w "${PREFIX}.pcap"
```

Note: On some computers the firewall blocked the incoming retransmitted packets from the EVSE. To resolve this issue a custom rule had to be added to the firewall configuration to allow traffic on the designated port.

3.4 Certificates for Testing and Simulation

ISO 15118-2 and -20 allow for more complex communication between the EV and EVSE, and the more advanced features of this protocol such as Plug and Charge (PnC), Vehicle-to-Grid (V2G), and Transport Layer Security (TLS) require specific certificates, as detailed in Section 2.1.3.4. Typically, access to the certificates by the EVSE is managed securely, for example through dedicated hardware security modules. However, for the purpose of the development of the framework, it is necessary to prepare certain testing certificates to allow the testing of all protocol features. These testing certificates, unlike production certificates, may not be stored securely. Along with the ComboCS simulator (see Section 3.2), Trialog has made available certificates for both ISO 15118-2 and ISO 15118-20 testing, which we will utilize for this purpose.

The EcoG *iso15118* framework utilizes these certificates in a specific configuration during runtime. Most of the required certificate chains are pre-joined and loaded directly by the codebase. The easiest way to integrate the provided certificates into our system is by converting their names and concatenating them into the final format expected by the framework. The full certificates list and conversion process is detailed in Appendix A, with a key transformation summarized in Table A.7, where the Leaf, SubCA2, and SubCA1 certificates are concatenated to form the pre-joined certificate chains.

The certificates provided also exhibit differences between their ISO 15118-2 and ISO 15118-20 versions, as detailed in Appendix A. To accommodate this, the code loading process was modified so that, after negotiating the protocol to use with the EV, the folder from which the certificates are loaded changes accordingly. This approach is used only for testing purposes since in a real production scenario there would ideally be a single set of certificates, with additional certificates included as needed for the most recent protocol version.

3.5 Summary

This chapter presented the pre-existing EV simulator with ISO 15118 capabilities, detailing its configuration via both the graphical user interface (GUI) and a REST API. These tools enabled flexible testing workflows by allowing parameter adjustment during runtime, essential for protocol validation.

A comprehensive explanation of the proposed system's development was provided, including hardware integration, software architecture, and implementation steps. Key contributions included adapting the IEC 61851-compliant EVSE developed by INESC TEC to support ISO 15118 communication, integrating open-source software stacks, and resolving protocol-layer challenges.

In addition, the chapter introduced auxiliary procedures developed to facilitate testing and debugging. These included tools for packet capture and live network analysis using Wireshark and dsV2Gshark, as well as the generation and management of test certificates. The main goal was to ensure that ISO 15118 protocol support could be seamlessly integrated into a production-ready charger, without compromising its existing functionality. Therefore, the implementation prioritized robustness, modularity, and reliable error handling.

To consolidate the concepts presented, Figure 3.17 provides a high-level overview of the developed test platform. It illustrates the interaction between the CSMS, the EVSE, the ComboCS simulator, and a real EV, highlighting the communication flow and key system components.

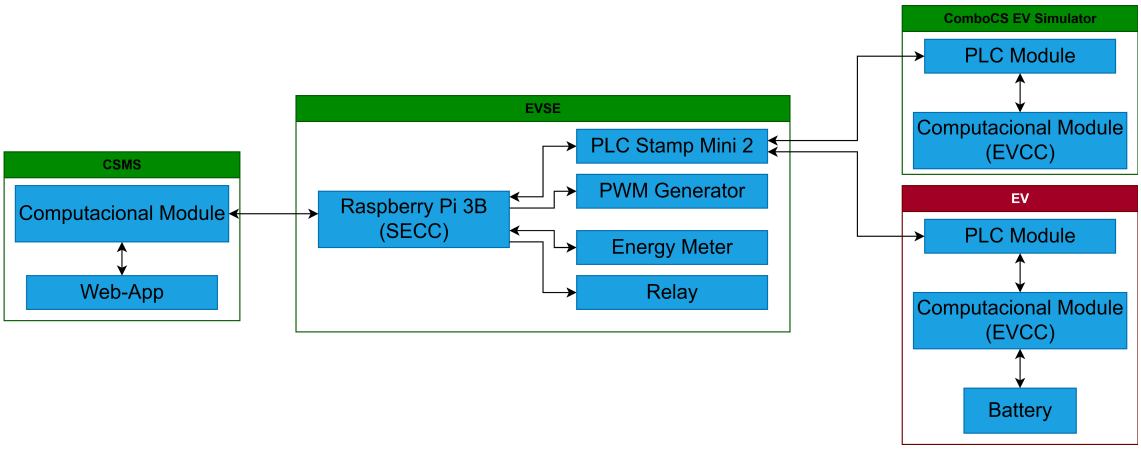


Figure 3.17: Simplified high-level block diagram of the overall EVSE test platform, showing communication between the CSMS, EVSE, simulator and EV.

Chapter 4

Results and Discussion

This chapter presents the results obtained from tests conducted in both simulated and real-world environments, with a focus on evaluating how different system configurations, such as communication security levels, payment options and charging schedule behavior, affect overall performance. The simulations were carried out using Trialog's ComboCS Electric Vehicle Simulator, which enabled a wide range of configurable test scenarios. For real-world validation, tests were conducted using a few chargers from INESC TEC's charging infrastructure.

The main objectives of this evaluation are to assess the performance and reliability of the developed charging system and to critically analyze the behavior and effectiveness of the ISO 15118 protocol under various operating conditions. All measurement data not explicitly shown in tables within this chapter is available in Appendix B.

4.1 Implementation with Simulator

In this section, the EVSE underwent extensive testing under simulated conditions using Trialog's ComboCS EV simulator (see Section 3.2). This simulation environment enabled safe development of the system architecture, validation of test cases, and prediction of real-world behavior, thereby helping to mitigate foreseeable issues before deployment.

4.1.1 Testing Setup

The initial testing setup focused solely on the communication flow between the EVSE and the simulator (EVCC), with no actual power transmission involved. As shown in Figure 4.1, the testbench consists of the developed EVSE stored within an INESC TEC charger enclosure, connected to the simulator solely through the CP line via a coaxial connector. The remaining two cables are a power supply cable, with a minimal current of approximately 2.5A which is sufficient for running the controller and basic peripherals and an ethernet cable which allows both connection to a potential CSMS and remote access for development via SSH. Testing was then conducted using both Trialog's graphical interface and the API controller as described in Section 3.2.2.

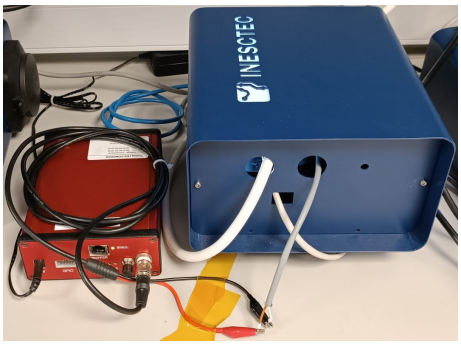


Figure 4.1: Testbench setup showing the developed EVSE connected to Trialog's EV Simulator via the CP cable.

4.1.2 ISO 15118-2 and 15118-20 Basic Communication

At the core of the ISO 15118 protocol lies the integration of the SLAC process, which was the first component tested and remains one of the most critical since it serves as a gateway to the rest of the ISO 15118 stack. During testing, several issues were encountered, most notably, SLAC sessions would often fail, `CM_SLAC_PARM.REQ` was never sent or even when completed successfully, would not trigger the transition to the next communication stage (SDP). After multiple attempts to resolve the issue on the EVSE side, the problem was ultimately traced to the simulator's PLC node. Reflashing the simulator's PLC firmware resolved the issue and restored proper SLAC functionality.

Following SLAC validation, the full ISO 15118 stack was thoroughly tested across a range of configurations combining protocol versions, authentication methods, and security layers. The matrix in Table 4.1 summarizes the combinations that were validated during testing.

Table 4.1: Matrix of tested ISO 15118 protocol configurations. An "X" indicates successful testing.

ISO Version	Payment Method	TLS 1.2	TLS 1.3	No TLS
ISO 15118-2	EIM	X	–	X
	PnC	X	–	–
ISO 15118-20	EIM	–	X	X
	PnC	–	–	–

Note: It is important to note that Plug & Charge (PnC) functionality could not be tested with ISO 15118-20, as this feature had not yet been developed or implemented by Trialog in their simulator at the time of testing.

4.1.3 TLS Integration

During the testing and analysis of V2G session message exchanges, accurately comparing message timings—particularly those in the charging preparation phase—was essential. When TLS

was not enabled, the messages were transmitted in plaintext, allowing for direct inspection of both the headers and contents in tools like *Wireshark*. However, once TLS was introduced, all message contents became encrypted, obscuring both their structure and semantics, as illustrated in Figure 4.2, where the previously identifiable `SupportedAppProtocolReq` message now appears merely as encrypted application data.

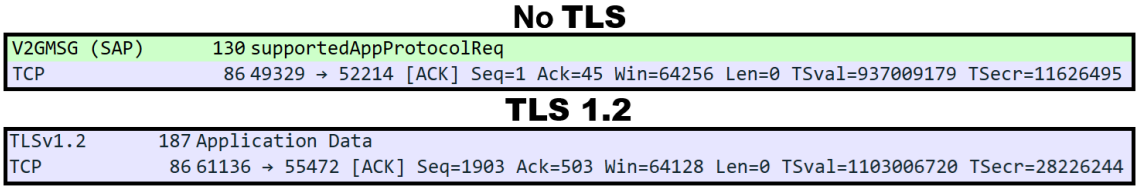


Figure 4.2: Comparison of TLS vs non-TLS packets visibility in Wireshark.

In this setup, although the TLS handshake could be captured, the only observable detail was the chosen cipher suite, information that does not pose a security risk on its own if secure algorithms are used. Other than that, no additional useful information could be extracted from the packets, as deeper analysis would require access to the cryptographic secrets generated by each end node.

4.1.3.1 Handshake Certificates Issue

The TLS handshake exchanges a certificate chain to authenticate both the server and client nodes which are, in this case, the EVSE and the ComboCS simulator. The root certificate, which serves as the trust anchor for these certificate chains, must therefore be available to each node so they can verify each other's authenticity. As detailed in Section 2.1.3.4, in TLS version 1.2 only the EVSE sends its certificate chain, meaning the simulator must have the corresponding root certificate to validate it. In version 1.3, however, both nodes exchange their certificate chains, which may be signed by different root authorities, potentially using cross-signing mechanisms, as illustrated in Figure 2.11.

In this simulation environment, the vehicle's certificate chain was signed by the V2G root authority, while the code architecture was expecting it to be signed by the OEM root. This mismatch led to issues during the TLS handshake, which required careful analysis to diagnose and resolve. The implementation was then updated to support vehicle certificates signed by the provided V2G root test certificate.

4.1.3.2 Inferring Message Types from Packet Size

As a workaround to the encryption, rather than analyzing the actual message content, the relative packet sizes were compared between sessions with and without TLS. This provided useful insight into the protocol's overhead and the approximate position of messages within the communication flow. A visual comparison of the packet lengths for both secure and unsecured sessions is shown in Figure 4.3.

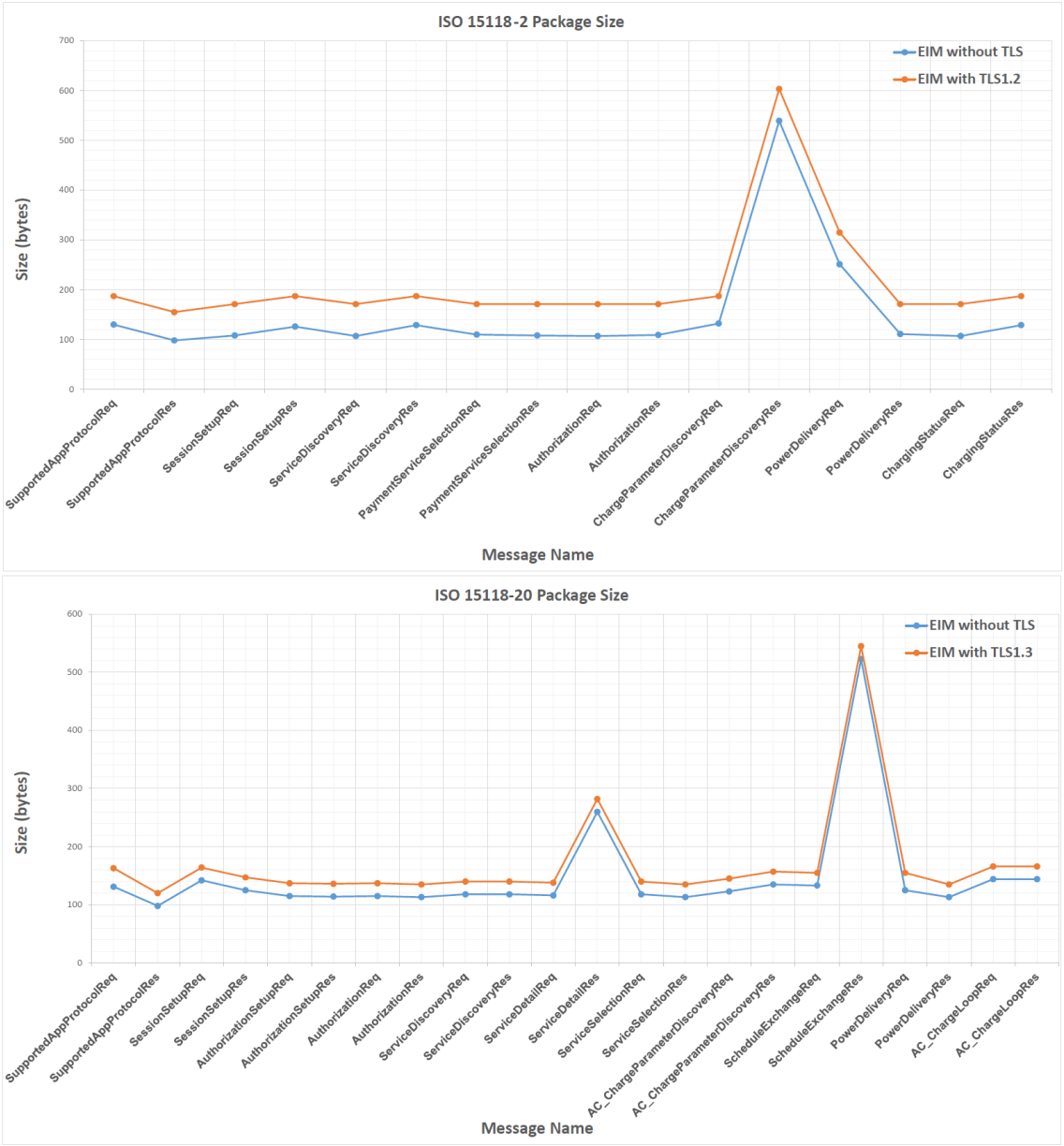


Figure 4.3: Comparison of packet lengths during V2G session setup in testbench sessions with and without TLS enabled.

Since both the simulator and EVSE configuration and decision logic remained identical across both TLS and no TLS V2G sessions the packet size trend follows a similar pattern in both cases, with an almost constant offset of 61 ± 3 bytes for TLS1.2 and 23 ± 3 for TLS1.3 observed throughout the message flow. This offset is due to the additional overhead introduced by the TLS encryption layer, which encapsulates each message with additional headers and metadata related to the protocol. This makes it possible to extrapolate the type of message being transmitted, assuming no unexpected renegotiations or retransmissions, based solely on the relative size and position of the encrypted packets.

4.1.4 Bidirectional Power Transfer (BPT) Testing

Bidirectional Power Transfer is supported exclusively through ISO 15118-20, which, in this evaluation, was tested only in simulation using the Trialog ComboCS simulator. During testing, the BPT channel parameter was transmitted and consistently set to `1`, indicating a single-phase power configuration. A charging schedule was successfully negotiated containing negative power values, which, according to the standard, implies that the EVSE is expected to receive energy from the EV (i.e., discharge from the EVCC to the grid or local load) during those time slots.

Although no actual energy transfer occurred, due to the absence of a physical power path or inverter in the simulation environment, the communication exchange and schedule negotiation were completed successfully. This confirms that the developed EVSE implementation supports the BPT communication logic and lays the groundwork for future real-world validation, once the appropriate hardware infrastructure and PKI certificate chains are in place.

4.2 Implementation with Electric Vehicle

Following extensive testing with the simulator, two chargers from INESC TEC's charging infrastructure were successfully upgraded to support High-Level Communication (HLC), as outlined in Section 3.3.1. In addition, the updated software architecture described in Section 3.3.2 was deployed. These modifications remained active for a duration of one month, during which operational data was collected. One of the chargers used during this evaluation period is shown in Figure 4.4.

Figure 4.4: One of the chargers from INESC TEC's infrastructure used during the evaluation period.

4.2.1 Charging Sessions

By acessing INESC TEC's internal CSMS, used for system monitoring and diagnostics, anonymized session data was obtained. An overview of charging session outcomes and communication methods is presented in Table 4.2.

Table 4.2: Session Breakdown for the two Chargers with ISO 15118 capabilities

Charger	Distinct Users	Total Sessions	LLC	HLC	Failed HLC
Charger 1	6	24	8	14	2
Charger 2	12	39	39	-	-

Out of all the charging sessions recorded over the testing period, a total of 18 distinct users used the charging infrastructure during the evaluation. However, successful High-Level Communication via the PLC nodes was established in only 16 sessions, all originating from just three distinct vehicles. This indicates that only 3 out of 18 distinct cars observed during the trial were compatible with the ISO 15118 protocol. For the remaining of vehicles that did not support HLC, it is unclear whether this was due to their launch date or a lack of protocol implementation by the manufacturer, even in newer models. These results suggest an estimated HLC compatibility rate of approximately 17%.

4.2.1.1 Failed HLC Sessions

During the evaluation period, only three distinct vehicles connected to the chargers and supported the protocol under study: a BMW i5, a Renault Megane E-Tech, and a Renault 5 E-Tech, all present in Table 2.5 as supporting this protocol. The BMW i5 followed the protocol flawlessly, enabling a complete ISO 15118-2 session with successful schedule negotiation. The two Renault vehicles, however, exhibited identical issues during the initial stages of the V2G communication session, as illustrated in Table 4.2.

The first two messages, `SupportedAppProtocol` and `SessionSetup` completed successfully, with both DIN SPEC 70121 and ISO 15118-2 protocols advertised. After selecting ISO 15118-2, the `ServiceDiscoveryRes` message correctly lists available services to the EVCC. However, at this point, the communication was abruptly terminated by the vehicle side, with the TCP server closing the connection. This unexpected behavior did not trigger a fallback to an LLC session, thereby preventing any form of charging. As a result, modifications were made to the code architecture to allow fallback behavior in such scenarios.

Further testing with both Renault vehicles was made by altering variables such as the advertised service set (forcing ISO 15118 DC charging message flow) and modifying the selected protocol (forcing negotiation to DIN SPEC message flow). The only configuration accepted by both vehicles was negotiation through the DIN SPEC protocol. However, DIN SPEC 70121 is a protocol intended only for DC charging, which this EVSE does not support. Furthermore, implementation of DIN SPEC was out of scope for this dissertation, even though supported by the base code architecture derived from the EcoG stack.

Notably, a similar issue has been reported in the public repository of the ISO 15118 stack maintained by EcoG [28], where a GitHub issue [32] describes the same communication breakdown occurring with a Tesla Model S and a Renault Zoe. No resolution has been proposed. Marc Mültin, co-author of the ISO 15118 protocol [11][14], commented on the issue, attributing the behavior to possible timing violations or other EV side faults that are not reported explicitly. According to the standard, such failures should be signaled through a dedicated error message, but this appears not to be the case.

4.2.1.2 Missing HLC Capabilities

While Table 2.5 lists vehicles known to support ISO 15118 communication which includes the two Renault models that failed communication mid-session due to protocol negotiation issues, additional unexpected cases were observed. Notably, at least one vehicle model that is publicly documented as supporting ISO 15118, the Volvo EX30, connected to the charger but failed to initiate the SLAC process entirely. Instead, the charging session immediately defaulted to the basic IEC 61851-1 communication protocol without any attempt to start a high-level communication (HLC) session.

This behavior contrasts with the two Renault vehicles discussed earlier, which did begin HLC handshakes but failed during the ISO 15118 message exchange. The Volvo EX30's immediate fallback to basic charging suggests that HLC functionality was either disabled, not activated, or selectively limited by the manufacturer in this particular configuration. This example highlights the inconsistency that can exist between a vehicle's advertised capabilities and its actual behavior in the field. It also reinforces that HLC availability may depend on specific factors of the OEM such as firmware versions, regional configurations, or optional feature activation.

4.2.1.3 Certificate Constraints

As previously mentioned, ISO 15118-20 mandates the use of TLS, which in turn requires a functional Public Key Infrastructure (PKI). Consequently, it was neither feasible nor expected to test the latest version of the protocol in a real scenario. Implementing ISO 15118-20 would have required either configuring a production ready CPO certificate chain, requiring additional planning and coordination with a V2G Root Certificate Authority, or establishing a private environment as discussed in Section 2.1.3.4. The latter would also demand manually installing a trusted root certificate (ideally issued by INESC TEC) on the EV.

During testing, not all vehicles requested the use of TLS during the SDP phase some requested the use of unsecured TCP communication during the SDP phase. However, the EVSE retains the ability to independently mandate the use of TLS by responding with the preference of using secure communication. When this renegotiation occurs, the TLS handshake is initiated. As anticipated, the handshake fails for all EVs due to the absence of the required certificate chain. As a result, the communication session automatically falls back to using unsecured TCP, without encryption.

Due to these certificate related constraints ISO 15118-2 was the only version evaluated in the live testing environment. For the same reasons only External Identification Means (EIM) was used as the authentication method, while Plug & Charge (PnC) was not supported. Since INESC TEC already had an existing authentication system in place, EIM was easier to integrate into the infrastructure.

4.2.2 Timeouts and User Experience Considerations

The timing parameters involved in the SLAC process (see Section 2.1.3.2), which is responsible for establishing the communication link, can introduce inconveniences, particularly when backward compatibility is required. If a vehicle does not support HLC, the SLAC procedure, according to the protocol, may be retried up to three times, resulting in a total of four attempts. Given that `TT_EVSE_SLAC_init` and `T_step_EF` are 20 seconds and 4 seconds respectively, this leads to a total potential delay of $(20+4)\times 4 = 96$ seconds before an LLC charging session based on IEC 61851-1 (see Section 2.1.2) can commence. This can significantly degrade the user experience. Even though `TT_EVSE_SLAC_init` is nominally 20 seconds, the EV is required to send a `CM_SLAC_PARM.REQ` within 10 seconds. Therefore, there exists an apparent 10-second buffer that is not strictly necessary [13]. To improve user experience, it is important to ensure that this process fails as quickly as possible when HLC is not supported. This can be achieved by either reducing the number of allowed retries or by adjusting our timers to remove this buffer, effectively setting `TT_EVSE_SLAC_init` to 10 seconds instead of 20.

The average time required for user authentication via INESC TEC's web-based authentication was measured at approximately 44 seconds. In comparison, the delay introduced by two SLAC attempts, totalling 28 seconds, remains below this average. As a result, the additional time introduced by SLAC retries has minimal impact on the overall user experience when compared to the previous setup, where charging sessions were initiated using only the IEC 61851 basic charging protocol without ISO 15118 support.

4.3 Protocol Flow Analysis

Since both simulated and real-world tests were conducted, this section presents a thorough and comparative analysis of the overall protocol flow, with a focus on communication efficiency, message timing, and deviations observed between simulated and live environments. Particular attention is given to the responsiveness of the system during HLC exchanges, the occurrence of timeouts, and fallback behavior.

4.3.1 SLAC Impact on V2G Communication Session Start Time

The SLAC process plays a pivotal role in initiating a high-level charging session, serving as the stepping stone between the EVSE and EVCC. This initial link setup not only enables physical

layer connectivity over PLC, but also acts as a gatekeeper to the ISO 15118 protocol stack. Consequently, the performance of the SLAC has a direct influence on the overall responsiveness and perceived efficiency of the charging system. Understanding how different EVs and simulation environments behave during this stage is essential to optimizing interoperability, minimizing delay, and ensuring a seamless user experience.

In this section, we assess and compare the SLAC timing characteristics across various test scenarios, highlighting how they impact the total time required to initiate a charging session. Table 4.3 presents the average and standard deviation of SLAC message timings for ComboCS and BMW i5, while Table 4.4 shows the same for Renault 5 E-Tech and Renault Megane E-Tech.

Table 4.3: Average and standard deviation of SLAC message timing for ComboCS and BMW i5.

	ComboCS		**BMW i5**	
Message	Avg (s)	σ (s)	Avg (s)	σ (s)
SLAC_PARM	1.27	0.04	1.9	0.2
SLAC_ATTEN	2.8×10^{-1}	0.1×10^{-1}	4.71×10^{-1}	0.03×10^{-1}
SLAC_ATTEN_CHAR	1.4×10^{-1}	0.5×10^{-1}	2×10^{-2}	1×10^{-2}
SLAC_MATCH	8.1×10^{-1}	0.7×10^{-1}	8.7×10^{-1}	0.5×10^{-1}
Wait SDP	7	2	5.5	0.6

Table 4.4: Average and standard deviation of SLAC message timing for Renault 5 E-Tech and Renault Mégane E-Tech.

	Renault 5 E-Tech		**Renault Mégane E-Tech**	
Message	Avg (s)	σ (s)	Avg (s)	σ (s)
SLAC_PARM	3×10^{-1}	1×10^{-1}	1.1	0.3
SLAC_ATTEN	3.8×10^{-1}	0.1×10^{-1}	5.1×10^{-1}	0.1×10^{-1}
SLAC_ATTEN_CHAR	4.5×10^{-2}	0.5×10^{-2}	4.1×10^{-2}	0.7×10^{-2}
SLAC_MATCH	8.0×10^{-1}	0.6×10^{-1}	8.1×10^{-1}	0.7×10^{-1}
Wait SDP	12	1	9.6	0.7

The `CM_SET_KEY.REQ/CNF` messages, which are the first exchanged as part of the SLAC handshake, are not shown in the tables since their timing depends solely on the EVSE's internal configuration process and does not vary across different vehicles or PLC chips. Notably, the Renault vehicles exhibit the lowest latency across most SLAC message exchanges, with the Megane showing slightly more consistent timing than the Renault 5. In contrast, the BMW i5 demonstrated the slowest SLAC process overall. The ComboCS simulator falls in between the two groups, with timing performance that is stable and representative, reinforcing its value as a representative implementation of the SLAC procedure.

To provide a clearer comparison between different configurations, the message timing sequences were plotted as bar graphs in Figure 4.5. The graph highlights the performance differences between devices.

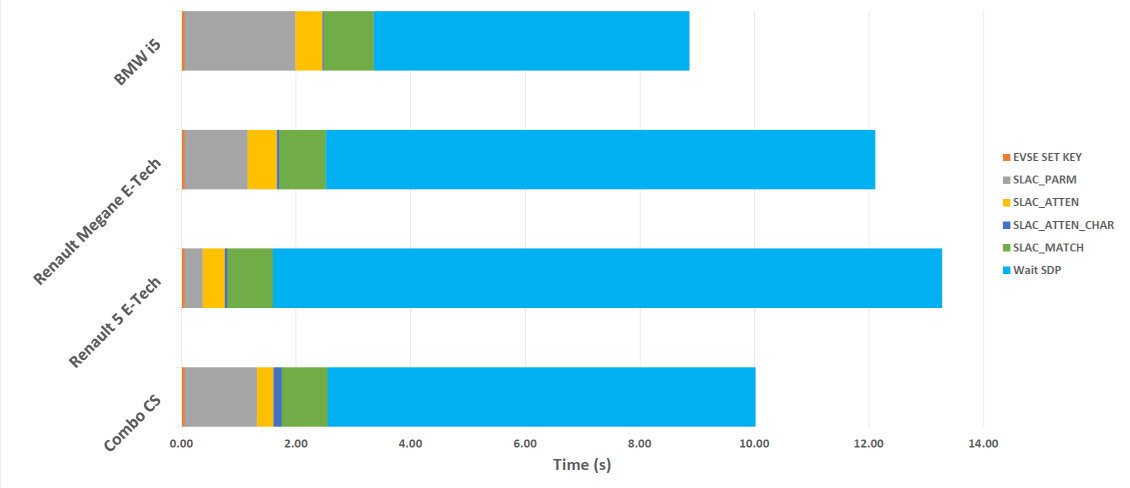


Figure 4.5: SLAC message timing comparison between developed EVSE and different test systems, using average durations per message type.

It is important to note that while the SLAC message timings initially suggest that the BMW i5 performs the slowest, a deeper analysis reveals otherwise. The final timing interval, measured between the end of SLAC and the beginning of the SDP message, captures the remaining handshake overhead, including the finalization of the PLC pairing on the EV side, particularly the `CM_SET_KEY.REQ/CNF` messages, as illustrated in Figure 2.9. These messages are not visible from the EVSE's perspective in Wireshark traces. This interval also includes the initialization of the UDP/SDP client on the EVCC or simulator. When this is taken into account, the BMW i5 demonstrates the fastest overall transition into the ISO 15118-2 communication stack, surpassing both the simulator and the Renault vehicles in end-to-end readiness efficiency.

All ten `CM_SOUND.REQ` and `CM_SOUND.CNF` messages exchanged between the EVSE and EVCC were successfully transmitted and received in every test scenario. This consistent exchange of SOUND messages is a strong indicator of link stability and communication reliability during the SLAC matching phase. Their successful completion across all tested configurations, both simulated and real-world, confirms that the underlying PLC channel and cable setup was of sufficient quality to support reliable data transmission, and that no packet loss or signal attenuation occurred severe enough to disrupt this critical portion of the handshake.

4.3.2 V2G Communication Session Message Sequence

Following the validation of the SLAC protocol, this section focuses on the message sequence defined by ISO 15118-2/ISO 15118-20, which constitutes the core of the V2G communication. A detailed analysis is provided for various message exchanges, considering different configurations such as authentication methods (EIM and PnC) and the use or absence of Transport Layer Security (TLS).The evaluation is based on multiple charging sessions conducted using both the ComboCS simulator and real electric vehicles, allowing for a assessment of the protocol's behavior.

4.3.2.1 EIM Communication Sequence: TLS vs. No TLS

In order to evaluate the impact of TLS on the timing efficiency of the ISO 15118 message flow, several charging sessions were conducted using the ComboCS simulator. These sessions enabled a comparison between encrypted communication sessions using TLS (versions 1.2 and 1.3) and unencrypted sessions over plain TCP. The primary objective was to assess the temporal overhead introduced by the TLS handshake and subsequent encryption on the V2G communication session.

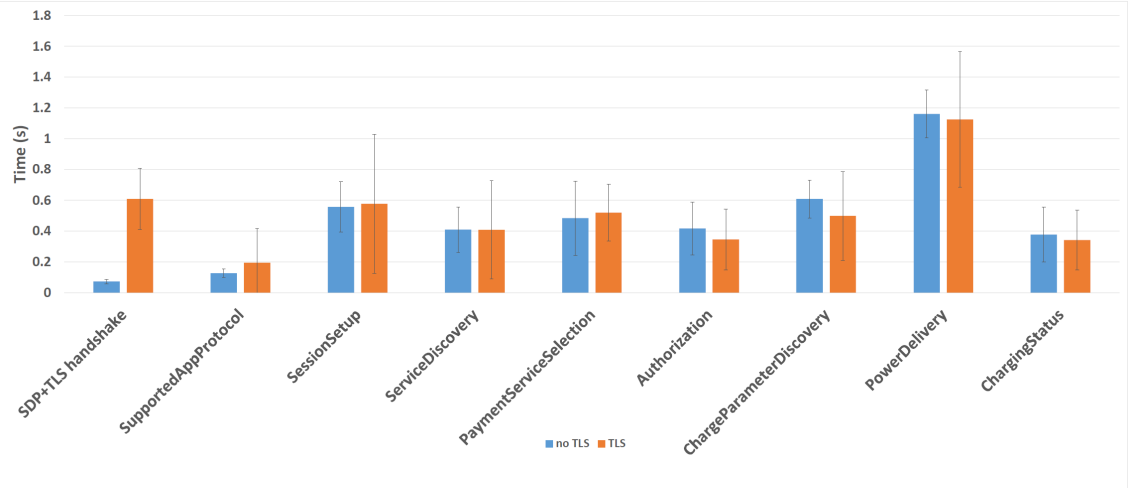


Figure 4.6: Message exchange timing in ISO 15118-2 with TLS 1.2 and without TLS.

Figure 4.6 presents the timing breakdown for each message exchanged during ISO 15118-2 sessions under EIM authentication. As expected, the `SDP+TLS Handshake` segment takes significantly longer, as it includes cryptographic operations and key negotiation processes. However, apart from this initial step, no substantial differences in timing were observed across the rest of the message flow between the TLS and no TLS cases.

It is also worth noting that while the average timing values remain relatively consistent, the variability is more pronounced in the TLS sessions. Specifically, messages exchanged over TLS 1.2 exhibited average standard deviations ranging from approximately 30% to 60%, whereas non-TLS sessions showed less variability. This suggests that the overhead introduced by TLS is less about fixed delays and more related to timing fluctuations due to encryption and networking dynamics.

A similar analysis was carried out for ISO 15118-20 sessions using TLS 1.3, as shown in Figure 4.7. The trends observed remain consistent: TLS introduces noticeable overhead during session initiation, but has minimal impact on individual message durations thereafter. In these sessions, standard deviations were somewhat lower compared to TLS 1.2, with encrypted messages showing variability between 24% and 31%. This confirms that while TLS ensures secure communication, its temporal cost is primarily concentrated at the session setup phase.

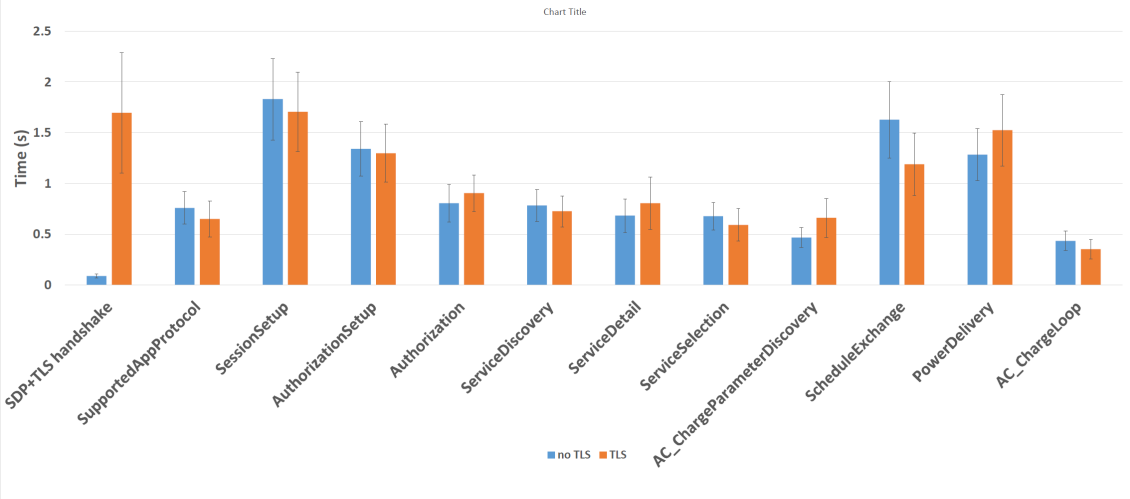


Figure 4.7: Message exchange timing in ISO 15118-20 with TLS 1.3 and without TLS.

4.3.2.2 Benefits of Plug & Charge in Session Initialization

Several communication tests were conducted between the EVSE, the ComboCS simulator, and real electric vehicles, offering valuable insight into message timing and overall protocol efficiency. Even in partially completed sessions, such as those described in Section 4.2.1.1, it was possible to analyze the initial sequence of message exchanges. Figure 4.8 presents a breakdown of the timing for each individual ISO 15118 message exchanged during a V2G session under EIM authentication without TLS. This detailed view allows for the comparison of relative processing and transmission delays across different systems.

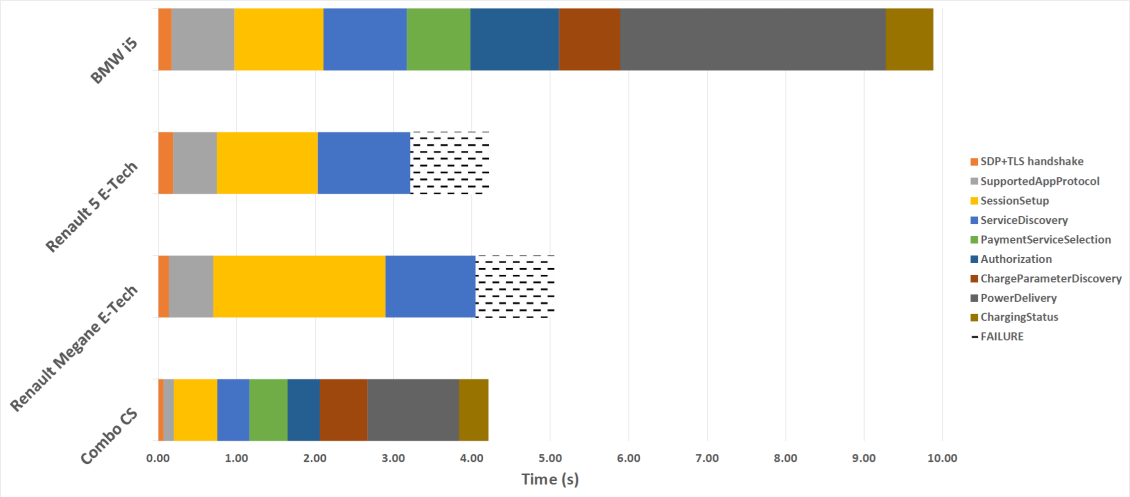


Figure 4.8: EIM without TLS. Timing of individual ISO 15118 message exchanges during a V2G session.

A notable timing difference can be observed between the EV simulator and the BMW i5, primarily due to additional processing delays on the vehicle side between message exchanges. The

simulator operates with rigid, pre-configured timing parameters and minimal runtime computation, which results in faster responses. For example, in the case of `PowerDelivery` messages, the simulator does not evaluate the received charging schedule or perform any calculations to determine an optimal charging profile. In contrast, the BMW i5 is likely making internal decision processes and safety checks which introduces delays. These differences highlight the variability in response behavior between the simulator and a fully integrated production EV.

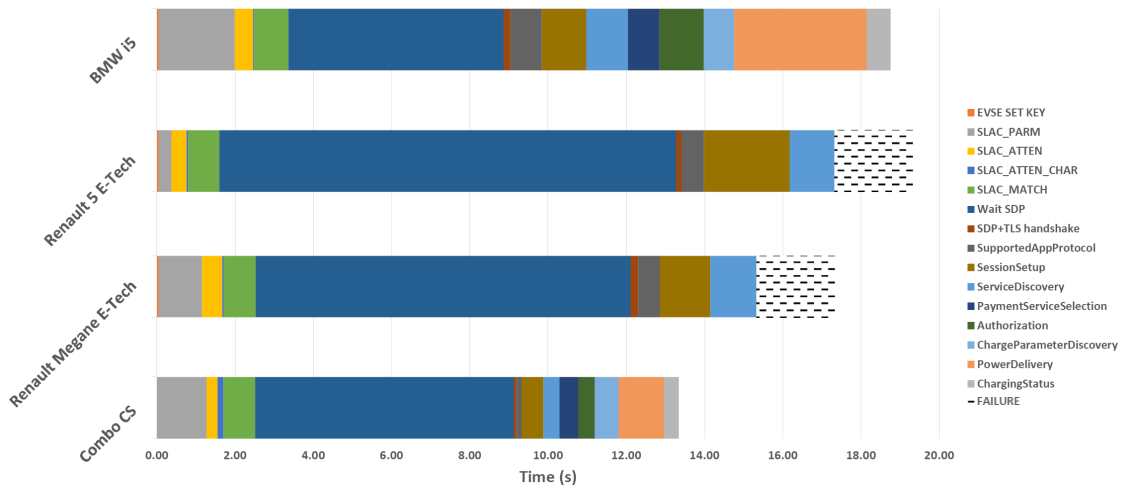


Figure 4.9: EIM without TLS. Total communication duration from vehicle plug-in to the start of power delivery.

The overall communication timeline from the moment the vehicle is plugged in until the initiation of power transfer is illustrated in Figure 4.9. For the BMW i5, which was the only real electric vehicle to successfully complete a full ISO 15118 session with this EVSE, the entire charging setup sequence took approximately 19 seconds. This duration is comparable to the fallback timeout of 28 seconds discussed in Section 4.2.2, which occurs when a vehicle lacks ISO 15118 support and defaults to a basic IEC 61851 session.

These measurements, however, do not account for the time required for user authentication in the case of External Identification Means. To enable a fair comparison between EIM and PnC, this delay must be considered. Based on data from INESC TEC's charging infrastructure, users take an average of 44 seconds to authenticate after plugging in. Although this authentication occurs in parallel with the SLAC process and the initial ISO 15118 messages up to the `AuthorizationReq`, the authorization itself only completes once the user has logged in, effectively introducing a hold-up at this stage of the communication flow.

In contrast, PnC eliminates the need for manual authentication, integrating this step seamlessly into the ISO 15118 message flow. `PaymentDetailsReq/Res` is the only additional message in a PnC session, in which the required certificates are exchanged. This message replaces the manual authentication process and allows the overall session to proceed without external user input, resulting in a more efficient and seamless communication sequence. These substantial differences in charging setup duration are clearly illustrated in Figure 4.10.

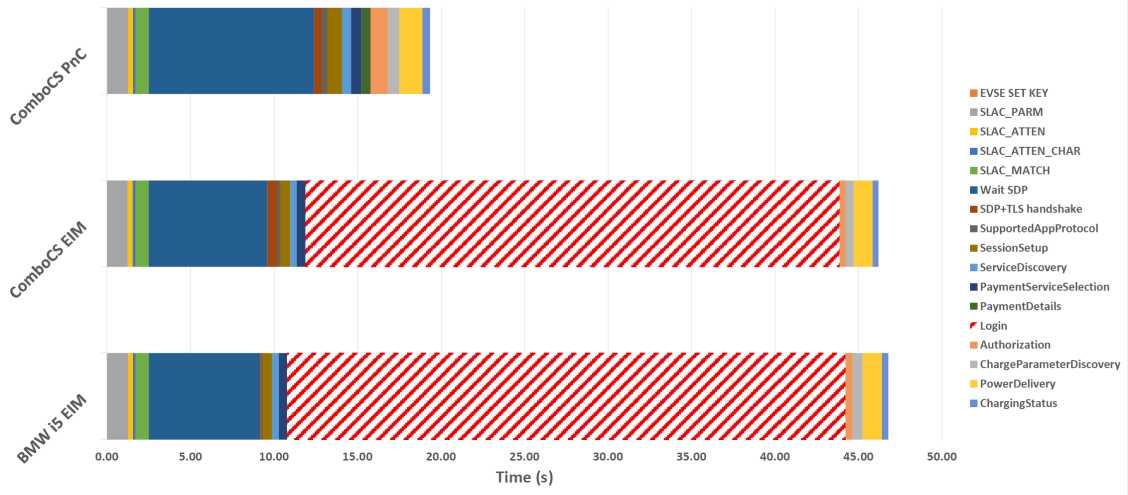


Figure 4.10: Comparison of total communication duration from vehicle plug-in to the start of power delivery: PnC vs. EIM.

4.3.2.3 Smart Charging Considerations

One of the core features of ISO 15118-2 related to smart charging is the ability to dynamically negotiate and adjust the charging schedule with the EVCC, including support for bidirectional energy transfer. As previously discussed, bidirectional capabilities could only be evaluated using the simulator. Schedule renegotiation can be initiated either by the EVSE, by indicating in the `ChargingStatusRes` message that a new `PowerDelivery` exchange should initiate, or by the EVCC itself directly with a `PowerDeliveryReq`.

Both mechanisms were successfully tested. The EVSE initiated renegotiation was validated with both the simulator and the BMW i5 having timing results consistent with earlier communication sequences. EVCC initiated renegotiation was easily tested in the simulator environment, as it allows full control over the behavior of the emulated vehicle, including the timing of such requests.

In contrast, this behavior could not be explicitly be configured or triggered on the BMW i5. However, during complete charging sessions, it was observed that the vehicle autonomously requested a schedule renegotiation approximately every 70 minutes after the previous negotiation. This periodicity appears to be specific of this vehicle's implementation, as ISO 15118-2 does not prescribe any fixed interval or behavior for renegotiation triggers on the EV side. These observations underline the flexibility of the protocol, while also highlighting potential variations in implementation between manufacturers.

4.3.2.4 ISO 15118-20 vs. ISO 15118-2

The most recent version of the protocol, ISO 15118-20, introduces a broader set of functionalities compared to ISO 15118-2, including native support for bidirectional charging (BPT), dynamic tariff structures, improved schedule negotiation mechanisms, and more robust error handling. As discussed in Chapter 2, it also mandates the use of TLS 1.3 for all secure communication, which enhances data integrity and privacy. To directly assess the efficiency of the new protocol version, a comparative analysis was performed between the total communication durations observed in two configurations as seen in Figure 4.11.

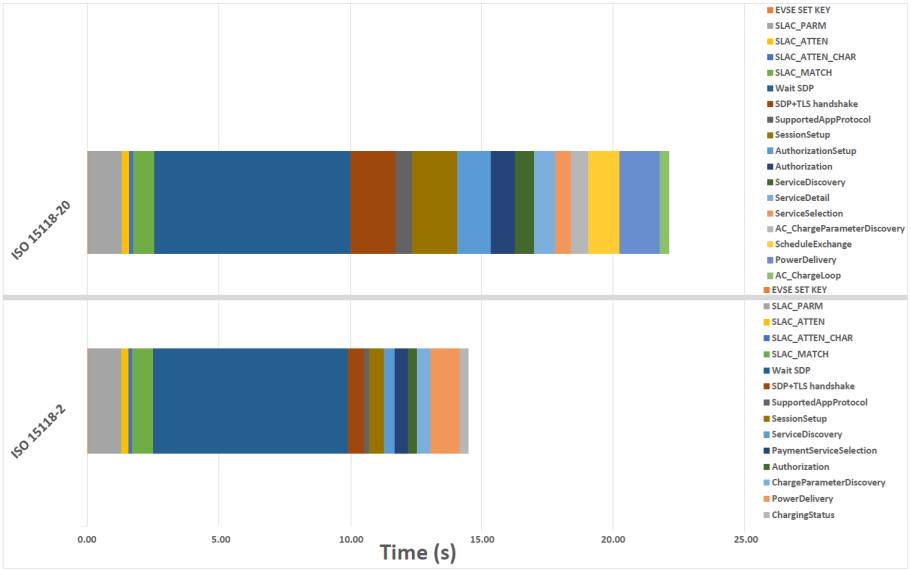


Figure 4.11: Comparison of total communication duration from vehicle plug-in to the start of power delivery: ISO 15118-20 vs. ISO 15118-2.

The comparison contrasts ISO 15118-2 using EIM authentication over TLS 1.2 with ISO 15118-20 using PnC over TLS 1.3. The analysis reveals that ISO 15118-20 requires a longer session setup time compared to ISO 15118-2. This increase is primarily due to the more complex message flow and the inclusion of additional mandatory procedures such as improved certificate handling, enhanced schedule negotiation steps, and stricter security verifications. Notably, PnC flow in ISO 15118-20 is designed to integrate identity authentication and contract verification directly into the message exchange, introducing additional cryptographic processing. However, it was not possible to test PnC with ISO 15118-20 in this work, as detailed at the end of Section 4.1.2.

Moreover, the use of TLS 1.3, although more efficient than TLS 1.2 in terms of handshake performance, still introduces temporal overhead due to the need for mutual certificate validation and key exchange. The cumulative impact of these factors results in an extended time between vehicle plug-in and the onset of power delivery.

While the ISO 15118-20 stack exhibits higher temporal cost in this initial phase, it brings significant architectural improvements that benefit long-term interoperability, security, and smart

charging functionalities. As such, the additional delay during session setup can be considered an acceptable trade-off given the protocol's enhanced capabilities.

4.4 Summary

This chapter presented the performance evaluation of the developed EVSE system through a comprehensive set of simulated and real-world tests. Key aspects analyzed included the impact of different ISO 15118 configurations, namely TLS security (1.2 and 1.3), authentication methods (EIM vs. PnC), and protocol versions (ISO 15118-2 vs. ISO 15118-20), on communication efficiency and reliability. Simulations using the ComboCS EV Simulator validated protocol compliance and allowed precise timing analysis under controlled conditions. Real-world testing on INESC TEC's infrastructure highlighted interoperability challenges, particularly with certain vehicle implementations, and emphasized the critical role of timing and certificate management in successful communication sessions. Plug & Charge (PnC) demonstrated tangible advantages over EIM by removing user-dependent authentication delays, while ISO 15118-20, though slightly more time-consuming during session setup, offers significant long-term benefits in functionality and security.

A noteworthy observation from the evaluation is the variability in how different electric vehicle manufacturers implement and communicate support for the ISO 15118 protocol. Even among vehicles officially supporting ISO 15118-2 or ISO 15118-20, observed behaviors such as SLAC initiation, fallback handling, or the availability of some features can differ significantly. This can result in variations in session initiation times, schedule renegotiation patterns, and error handling across vehicle models. Moreover, OEMs do not always clearly disclose the extent or conditions of their protocol support, which can lead to unexpected outcomes, such as in the case of the Volvo EX30, which bypassed high-level communication entirely despite being expected to support it.

Such variability complicates interoperability assessments and underscores the importance of real-world validation when evaluating ISO 15118 deployment readiness. In these cases, differences may be due to specific software configurations, firmware versions, or optional features requiring activation. These observations highlight the need for more robust conformance testing frameworks and clearer implementation guidelines to foster consistent behavior across manufacturers and vehicle platforms.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This work aimed to integrate ISO 15118-based communication into a real-world EVSE environment, combining both simulated testing and live validation. The developed architecture successfully enabled V2G communication using both ISO 15118-2 and ISO 15118-20, supporting various security configurations, authentication methods, and simulated BPT.

The use of Trialog's ComboCS simulator was instrumental in the initial stages, allowing controlled validation of protocol support and timing behavior under various configurations. Key insights were drawn from this simulator-based environment, including the performance impact of TLS versions, the behavior of message exchanges under different authentication mechanisms, and the limitations of EV-side implementations.

Real-world testing was conducted using two upgraded chargers from INESC TEC's infrastructure. Over the course of a month, several EVs were connected, though only a few supported HLC. Among these, only one vehicle, the BMW i5, completed a full ISO 15118 session, revealing that protocol support among current EVs remains limited. Despite this, the developed implementation proved to be robust—after a few initial adjustments during deployment, it operated reliably across all vehicles, correctly identifying unsupported configurations and falling back to basic charging when necessary. Timing data and fallback behaviors were analyzed to assess the practical implications of integrating ISO 15118 with the infrastructure.

The results confirmed that ISO 15118-2 with EIM authentication is already usable in real infrastructure, but the integration of ISO 15118-20 faces significant practical challenges, primarily due to the mandatory use of TLS1.3 and the absence of a certificate infrastructure. Without valid PKI roots and trusted intermediates, mutual authentication fails during the initial handshake, rendering it impossible to verify if a vehicle supports ISO 15118-20 or to complete a secure session.

5.2 Future Work

A critical next step is the development of a custom EV simulator to replace or complement existing tools like the ComboCS. A dedicated simulator would grant full control over the EVCC software stack and hardware integration, enabling deeper exploration of the ISO 15118 protocol. This includes testing edge cases, simulating faults, and introducing complex timing variations to assess EVSE robustness.

More importantly, the simulator should support real energy transfer, both consumption and BPT, allowing for the evaluation of real-time energy negotiation logic. This approach removes the dependency on certificate authorities, enabling meaningful closed-loop tests even in the absence of a full PKI infrastructure. The simulator under development at INESC TEC already aims to support these functionalities, including a web-based GUI for switching between different modes of operation and a physically present DC load for realistic battery charging behavior.

Complementing the simulator, future work should also focus on implementing DC charging capabilities on the EVSE itself. Unlike AC, DC charging introduces more stringent safety, control, and protocol requirements, as it directly interfaces with the EV battery management system. These requirements are formally specified in the IEC 61851-23 protocol, which defines the electrical characteristics and safety protocols for a DC EV charging infrastructure. Implementing and testing ISO 15118 DC-specific flows, especially in combination with BPT and TLS1.3, will require more careful design and hardware preparation. The simulator must also be capable of safely emulating DC loads to facilitate this stage.

An additional direction would include enhancing the charger's interoperability with external energy sources, such as photovoltaic (PV) systems, using standards like OCPP and incorporating smart charging logic. This integration would enable dynamic load management, optimized energy usage based on solar generation, and alignment with grid conditions.

Another important avenue is the integration of real certificate handling. The absence of a valid PKI severely limits the scope of testing, especially for ISO 15118-20 where TLS1.3 is mandatory. Without trusted root certificates and a compliant chain, it is impossible to confirm a car's support for newer standards, since the handshake fails at the cryptographic level. Either acquiring access to a V2G-compliant certificate authority or deploying a controlled PKI test environment will be essential. Certification would validate that the EVSE behaves according to the standard, increasing trust and opening possibilities for deployment in production-grade infrastructures. Certification would also enable participation in interoperability events and access to broader EV testing pools.

Security testing and validation also deserve further attention. This includes targeted assessments of TLS handshake behavior, fault injection, and attempts to simulate timing or message manipulation. Preliminary attempts at this were limited due to the encrypted nature of the message flow

once TLS is engaged, but the development of internal logging within a custom simulator could enable deeper insights into how encryption affects the protocol's resilience.

In summary, this dissertation established a working foundation for ISO 15118 implementation in INESC TEC's EVSE infrastructure, validated core communication sequences, and identified key limitations. The next phase should focus on expanding testing capabilities through custom simulator development, enabling DC testing, implementing certificate-based security mechanisms, and pursuing compliance certification. These steps are essential to unlock the full potential of ISO 15118 and enable advanced services such as secure Plug & Charge and bidirectional energy management.

References

[1] Oktavia; Catsaros. “Electric Vehicle Sales Headed for Record Year but Growth Slowdown Puts Climate Targets at Risk, According to BloombergNEF Report | BloombergNEF”. In: *BloombergNEF* (June 2024). Accessed: 2024-10-28. URL: https://about.bnef.com/blog/electric-vehicle-sales-headed-for-record-year-but-growth-slowdown-puts-climate-targets-at-risk-according-to-bloombergnef-report/.

[2] Stephen Brown, David Pyke, and Paul Steenhof. “Electric vehicles: The role and importance of standards in an emerging market”. In: *Energy Policy* 38 (7 July 2010), pp. 3797–3806. ISSN: 03014215. DOI: 10.1016/j.enpol.2010.02.059.

[3] Sanchari Deb et al. “Impact of electric vehicle charging station load on distribution network”. In: *Energies* 11 (1 2018). ISSN: 19961073. DOI: 10.3390/en11010178.

[4] Yanning Li and Alan Jenn. “Impact of electric vehicle charging demand on power distribution grid congestion”. In: *Proceedings of the National Academy of Sciences of the United States of America* 121 (18 Apr. 2024). ISSN: 10916490. DOI: 10.1073/pnas.2317599121.

[5] Olalekan Kolawole and Irfan Al-Anbagi. *The Impact of EV Battery Cycle Life on Charge-Discharge Optimization in a V2G Environment*. IEEE, July 2018.

[6] István Szúcs et al. “Analyzing EV Users’ Charging Patterns for Estimating Session Parameters and Optimizing Load Management”. In: *2025 IEEE 23rd World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, Jan. 2025, pp. 000521–000526. ISBN: 979-8-3503-7936-5. DOI: 10.1109/SAMI63904.2025.10883303. URL: https://ieeexplore.ieee.org/document/10883303/.

[7] R. Hari Raghav et al. “Establishing communication between EV - EVSE - CSMS based on IEC61851-1 and OCPP 2.0.1 Standard”. In: *2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023*. Institute of Electrical and Electronics Engineers Inc., 2023. ISBN: 9798350335095. DOI: 10.1109/ICCCNT56998.2023.10308172.

[8] IEC. *IEC 61851-1: Electric vehicle conductive charging system. Part 1, General requirements*. Ed. by Peter; Van Den Bossche. International Electrotechnical Commission, 2010, p. 99. ISBN: 9782889122226.

[9] Marc Multin. “ISO 15118 as the Enabler of Vehicle-to-Grid Applications”. In: *2018 International Conference of Electrical and Electronic Technologies for Automotive, AUTOMOTIVE 2018* (Oct. 2018). DOI: 10.23919/EETA.2018.8493213.

[10] Nicolas Ginot et al. “Application of power line communication for data transmission over PWM network”. In: *IEEE Transactions on Smart Grid* 1 (2 Sept. 2010), pp. 178–185. ISSN: 19493053. DOI: 10.1109/TSG.2010.2053225.

[11] ISO. *ISO 15118-3: Road vehicles — Vehicle to grid communication interface. Part 3: Physical and data link layer requirements*. International Organization for Standardization, 2015, p. 86.

[12] Mahmood A. Al-Shareeda et al. "IPv6 Link-Local Network SLAAC Attack Detection Mechanisms: A Review". In: *Proceedings - CSCTIT 2022: 5th College of Science International Conference on Recent Trends in Information Technology* (2022). DOI: 10.1109/CSCTIT56299.2022.10145649.

[13] Ed Watt, Mayuresh Savargaonkar, and Pranav Gadamsetty. *TIMEOUTS Best Practices*. Tech. rep. ChargeX consortium, Mar. 2025.

[14] ISO. *ISO 15118-20: Road vehicles — Vehicle to grid communication interface. Part 20: 2nd generation network layer and application layer requirements*. International Organization for Standardization, 2022.

[15] Ahmet Kilic. "TLS-handshake for Plug and Charge in vehicular communications". In: *Computer Networks* 243 (Apr. 2024). ISSN: 13891286. DOI: 10.1016/j.comnet.2024.110281.

[16] Rainer Falk and Steffen Fries. *Electric Vehicle Charging Infrastructure-Security Considerations and Approaches*. Siemens AG, Corporate Technology, 2012. ISBN: 9781612082042. URL: https://www.researchgate.net/publication/279852975.

[17] Mauro Conti et al. "EVExchange: A Relay Attack on Electric Vehicle Charging System". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 13554 LNCS. Springer Science and Business Media Deutschland GmbH, 2022, pp. 488–508. ISBN: 9783031171390. DOI: 10.1007/978-3-031-17140-6_24.

[18] Samrat Acharya, Yury Dvorkin, and Ramesh Karri. "Public Plug-in Electric Vehicles + Grid Data: Is a New Cyberattack Vector Viable?" In: *IEEE Transactions on Smart Grid* 11 (6 Nov. 2020), pp. 5099–5113. ISSN: 19493061. DOI: 10.1109/TSG.2020.2994177.

[19] Ross Porter et al. "Extending ISO 15118-20 EV Charging: Preventing Downgrade Attacks and Enabling New Security Capabilities". In: *2024 21st Annual International Conference on Privacy, Security and Trust (PST)*. IEEE, Aug. 2024, pp. 1–9. ISBN: 979-8-3503-6709-6. DOI: 10.1109/PST62714.2024.10788058. URL: https://ieeexplore.ieee.org/document/10788058/.

[20] K. I.L.I.C. Ahmet. "Smart, Secure and Interoperable Charging Infrastructure with Plug and Charge". In: *12th International Conference on Smart Grid, icSmartGrid 2024*. Institute of Electrical and Electronics Engineers Inc., 2024, pp. 174–180. ISBN: 9798350361612. DOI: 10.1109/icSmartGrid61824.2024.10578287.

[21] Gaurav Kumar and Suresh Mikkili. "Critical Review of Vehicle-to-Everything (V2X) Topologies: Communication, Power Flow Characteristics, Challenges, and Opportunities". In: *CPSS Transactions on Power Electronics and Applications* 9 (1 Mar. 2024), pp. 10–26. ISSN: 2475742X. DOI: 10.24295/CPSSTPEA.2023.00042.

[22] Josef Meiers and Georg Frey. "A Case Study of the Use of Smart EV Charging for Peak Shaving in Local Area Grids". In: *Energies* 17 (1 Jan. 2024). ISSN: 19961073. DOI: 10.3390/en17010047.

[23] Kristian Sevdari, Peter Bach Andersen, and Mattia Marinelli. "Aggregation and Control of Electric Vehicles AC Charging for Grid Services Delivery". In: *IEEE Transactions on Smart Grid* (2024). ISSN: 19493061. DOI: 10.1109/TSG.2024.3492391.

[24] EV Database. *EV Database*. Accessed: 2025-04-13. Apr. 2025. URL: https://ev-database.org/.

[25] Trialog. *EV ComboCS - The CCS Tester Datasheet v5 2023-02-06*. Feb. 2023.

[26] Raspberry Pi Ltd. *Raspberry Pi 3 Model B+ Product Brief*. Jan. 2025.

[27] Chargebyte. *PLC Stamp mini 2 Datasheet*. chargebyte GmbH, Aug. 2023.

[28] EcoG-io. "EcoG-io ISO15118". In: (June 2024). Accessed: 2024-12-30. URL: https://github.com/EcoG-io/iso15118.

[29] EcoG-io. "EcoG-io pyslac". In: (Aug. 2022). Accessed: 2024-12-30. URL: https://github.com/EcoG-io/pyslac.

[30] Wireshark Foundation. *Wireshark*. Apr. 2025. URL: https://www.wireshark.org/.

[31] dSPACE. *dsV2Gshark*. 2025. URL: https://github.com/dspace-group/dsV2Gshark.

[32] chrisdotam. *ISO 15118 Git Issue - Communication is closed after ServiceDiscovery #131*. Accessed: 2025-05-15. Sept. 2022. URL: https://github.com/EcoG-io/iso15118/issues/131.

Appendix A

Test Certificates Naming and Conversion

The certificates conversion from Trialog's format (original) to EcoG's format (converted) follows the convention shown in the following tables:

Table A.1: ISO 15118-2 Keys Conversion

Original (Private Keys)	**Converted (private_keys)**
contractCert.key	contractLeaf.key
moSubCA2.key	moSubCA2.key
moSubCA1.key	moSubCA1.key
moRootCA.key	moRootCA.key
seccCert.key	seccLeaf.key
cpoSubCA2.key	cpoSubCA2.key
cpoSubCA1.key	cpoSubCA1.key
v2gRootCA.key	v2gRootCA.key

Table A.2: ISO 15118-20 Keys Conversion

Original (Private Keys)	**Converted (private_keys)**
contractCert.key	contractLeaf.key
moSubCA2.key	moSubCA2.key
moSubCA1.key	moSubCA1.key
moRootCA.key	moRootCA.key
cpsLeafCert.key	cpsLeaf.key
cpsSubCA2.key	cpsSubCA2.key
cpsSubCA1.key	cpsSubCA1.key
oemProvCert.key	oemLeaf.key
oemSubCA2.key	oemSubCA2.key
oemSubCA1.key	oemSubCA1.key
oemRootCA.key	oemRootCA.key
secc20Cert.key	seccLeaf.key
cpo20SubCA2.key	cpoSubCA2.key
cpo20SubCA1.key	cpoSubCA1.key
vehicle20Cert.key	vehicleLeaf.key
vehicle20SubCA2.key	vehicleSubCA2.key
vehicle20SubCA1.key	vehicleSubCA1.key
v2g20RootCA.key	v2gRootCA.key

Table A.3: ISO 15118-2 Certificates Conversion

Original (certs)	**Converted (certs)**
contractCert.pem	contractLeaf.pem
moSubCA2.pem	moSubCA2.pem
moSubCA1.pem	moSubCA1.pem
moRootCA.pem	moRootCA.pem
seccCert.pem	seccLeaf.pem
cpoSubCA2.pem	cpoSubCA2.pem
cpoSubCA1.pem	cpoSubCA1.pem
v2gRootCA.pem	v2gRootCA.pem

Table A.4: ISO 15118-20 Certificates Conversion

Original (certs)	**Converted (certs)**
contractCert.pem	contractLeaf.pem
moSubCA2.pem	moSubCA2.pem
moSubCA1.pem	moSubCA1.pem
moRootCA.pem	moRootCA.pem
cpsLeafCert.pem	cpsLeaf.pem
cpsSubCA2.pem	cpsSubCA2.pem
cpsSubCA1.pem	cpsSubCA1.pem
oemProvCert.pem	oemLeaf.pem
oemSubCA2.pem	oemSubCA2.pem
oemSubCA1.pem	oemSubCA1.pem
oemRootCA.pem	oemRootCA.pem
secc20Cert.pem	seccLeaf.pem
cpo20SubCA2.pem	cpoSubCA2.pem
cpo20SubCA1.pem	cpoSubCA1.pem
vehicle20Cert.pem	vehicleLeaf.pem
vehicle20SubCA2.pem	vehicleSubCA2.pem
vehicle20SubCA1.pem	vehicleSubCA1.pem
v2g20RootCA.pem	v2gRootCA.pem

Table A.5: ISO 15118-2 CSRs Conversion

Original (csrs)	**Converted (csrs)**
contractCert.csr	contractLeaf.csr
moSubCA2.csr	moSubCA2.csr
moSubCA1.csr	moSubCA1.csr
moRootCA.csr	moRootCA.csr
seccCert.csr	seccLeaf.csr
cpoSubCA2.csr	cpoSubCA2.csr
cpoSubCA1.csr	cpoSubCA1.csr
v2gRootCA.csr	v2gRootCA.csr

Table A.6: ISO 15118-20 CSRs Conversion

Original (csrs)	**Converted (csrs)**
contractCert.csr	contractLeaf.csr
moSubCA2.csr	moSubCA2.csr
moSubCA1.csr	moSubCA1.csr
moRootCA.csr	moRootCA.csr
seccCert.csr	seccLeaf.csr
cpoSubCA2.csr	cpoSubCA2.csr
cpoSubCA1.csr	cpoSubCA1.csr
v2gRootCA.csr	v2gRootCA.csr
cpsLeafCert.csr	cpsLeaf.csr
cpsSubCA2.csr	cpsSubCA2.csr
cpsSubCA1.csr	cpsSubCA1.csr
oemProvCert.csr	oemLeaf.csr
oemSubCA2.csr	oemSubCA2.csr
oemSubCA1.csr	oemSubCA1.csr
oemRootCA.csr	oemRootCA.csr

Table A.7: Certificate Chain Creation. Concatenated Certificates in order

Concatenated Certificates	**Chain File**
seccLeaf.pem cpoSubCA2.pem cpoSubCA1.pem	cpoCertChain.pem
oemLeaf.pem oemSubCA2.pem oemSubCA1.pem	oemCertChain.pem
vehicleLeaf.pem vehicleSubCA2.pem vehicleSubCA1.pem	vehicleCertChain.pem

Appendix B

Communication Measurements

Table B.1: ComboCS Packet size measurements in bytes for each ISO 1511-2 message during V2G session setup, with and without TLS1.2. All values are consistent across five test sessions with no observed variation.

Message	**No TLS (bytes)**	**TLS (bytes)**
SupportedAppProtocolReq	130	187
SupportedAppProtocolRes	98	155
SessionSetupReq	108	171
SessionSetupRes	126	187
ServiceDiscoveryReq	107	171
ServiceDiscoveryRes	129	187
PaymentServiceSelectionReq	110	171
PaymentServiceSelectionRes	108	171
AuthorizationReq	107	171
AuthorizationRes	109	171
ChargeParameterDiscoveryReq	132	187
ChargeParameterDiscoveryRes	539	603
PowerDeliveryReq	251	315
PowerDeliveryRes	111	171
ChargingStatusReq	107	171
ChargingStatusRes	129	187

Table B.2: ComboCS Packet size measurements (in bytes) for ISO 15118-20 messages during a V2G session, comparing usage with and without TLS1.3. Values reflect application data record payload lengths.

Message	**No TLS (bytes)**	**TLS (bytes)**
SupportedAppProtocolReq	131	163
SupportedAppProtocolRes	98	120
SessionSetupReq	142	164
SessionSetupRes	125	147
AuthorizationSetupReq	115	137
AuthorizationSetupRes	114	136
AuthorizationReq	115	137
AuthorizationRes	113	135
ServiceDiscoveryReq	118	140
ServiceDiscoveryRes	118	140
ServiceDetailReq	116	138
ServiceDetailRes	260	282
ServiceSelectionReq	118	140
ServiceSelectionRes	113	135
AC_ChargeParameterDiscoveryReq	123	145
AC_ChargeParameterDiscoveryRes	135	157
ScheduleExchangeReq	113	155
ScheduleExchangeRes	523	545
PowerDeliveryReq	125	155
PowerDeliveryRes	113	135
AC_ChargeLoopReq	144	166
AC_ChargeLoopRes	144	166

Table B.3: ComboCS SLAC Message Timing Breakdown

Message	**Average (s)**	**σ (s)**
EVSE SET KEY	0.0005	0.0001
SLAC_PARM	1.269	0.038
SLAC_ATTEN	0.283	0.012
SLAC_ATTEN_CHAR	0.144	0.049
SLAC_MATCH	0.805	0.066
Wait SDP	7.469	2.124

Table B.4: ComboCS Raw Timing Data for ISO 15118-2 Messages. NO TLS + EIM

Message	**Average (s)**	σ **(s)**
SDP+TLS handshake	0.072	0.014
SupportedAppProtocol	0.127	0.028
SessionSetup	0.558	0.162
ServiceDiscovery	0.409	0.147
PaymentServiceSelection	0.483	0.242
Authorization	0.416	0.173
ChargeParameterDiscovery	0.608	0.123
PowerDelivery	1.160	0.157
ChargingStatus	0.378	0.177

Table B.5: ComboCS Raw Timing Data for ISO 15118-2 Messages. TLS + EIM

Message	**Average (s)**	σ **(s)**
SDP+TLS handshake	0.609	0.198
SupportedAppProtocol	0.194	0.222
SessionSetup	0.576	0.451
ServiceDiscovery	0.408	0.319
PaymentServiceSelection	0.520	0.185
Authorization	0.345	0.198
ChargeParameterDiscovery	0.498	0.290
PowerDelivery	1.124	0.440
ChargingStatus	0.342	0.194

Table B.6: ComboCS Raw Timing Data for ISO 15118-2 Messages. TLS + PnC

Message	**Average (s)**	σ **(s)**
SDP+TLS handshake	0.461	0.007
SupportedAppProtocol	0.320	0.298
SessionSetup	0.875	0.595
ServiceDiscovery	0.554	0.375
PaymentServiceSelection	0.617	0.323
PaymentDetails	0.539	0.276
Authorization	1.039	0.510
ChargeParameterDiscovery	0.697	0.354
PowerDelivery	1.380	0.198
ChargingStatus	0.444	0.176

Table B.7: Renault 5 E-Tech Raw Timing Data for SLAC and ISO 15118-2 Messages before failure. No TLS + EIM. Not enough measurements for a standard deviation.

Message	**Average (s)**
EVSE SET KEY	0.051
SLAC_PARM	0.318
SLAC_ATTEN	0.387
SLAC_ATTEN_CHAR	0.045
SLAC_MATCH	0.795
Wait SDP	11.682
SDP+TLS handshake	0.136
SupportedAppProtocol	0.567
SessionSetup	2.196
ServiceDiscovery	1.151

Table B.8: Renault Megane E-Tech Raw Timing Data for SLAC and ISO 15118-2 Messages before failure. No TLS + EIM. Not enough measurements for a standard deviation.

Message	**Average (s)**
EVSE SET KEY	0.050
SLAC_PARM	1.105
SLAC_ATTEN	0.513
SLAC_ATTEN_CHAR	0.041
SLAC_MATCH	0.810
Wait SDP	9.590
SDP+TLS handshake	0.192
SupportedAppProtocol	0.555
SessionSetup	1.289
ServiceDiscovery	1.180

Table B.9: BMW i5 Raw Timing Data for SLAC and ISO 15118-2 Messages. No TLS + EIM

Message	**Average (s)**
EVSE SET KEY	0.050
SLAC_PARM	1.939
SLAC_ATTEN	0.471
SLAC_ATTEN_CHAR	0.025
SLAC_MATCH	0.873
Wait SDP	5.510
SDP+TLS handshake	0.171
SupportedAppProtocol	0.800
SessionSetup	1.138
ServiceDiscovery	1.063
PaymentServiceSelection	0.809
Authorization	1.131
ChargeParameterDiscovery	0.776
PowerDelivery	3.389
ChargingStatus	0.605

Table B.10: ComboCS Raw Timing Data for ISO 15118-20 Messages. No TLS + EIM

Message	**Average (s)**	σ **(s)**
SDP+TLS handshake	0.089	0.022
SupportedAppProtocol	0.759	0.167
SessionSetup	1.829	0.366
AuthorizationSetup	1.340	0.281
Authorization	0.805	0.169
ServiceDiscovery	0.783	0.157
ServiceDetail	0.682	0.171
ServiceSelection	0.677	0.162
AC_ChargeParameterDiscovery	0.468	0.094
ScheduleExchange	1.626	0.407
PowerDelivery	1.283	0.269
AC_ChargeLoop	0.435	0.100

Table B.11: ComboCS Raw Timing Data for ISO 15118-20 Messages. TLS + EIM

Message	**Average (s)**	σ **(s)**
SDP+TLS handshake	1.695	0.390
SupportedAppProtocol	0.652	0.176
SessionSetup	1.705	0.358
AuthorizationSetup	1.298	0.389
Authorization	0.904	0.298
ServiceDiscovery	0.726	0.160
ServiceDetail	0.804	0.169
ServiceSelection	0.590	0.224
AC_ChargeParameterDiscovery	0.661	0.198
ScheduleExchange	1.189	0.452
PowerDelivery	1.523	0.503
AC_ChargeLoop	0.352	0.095