Cyber security for Electric Vehicle Smart Charging Energy Network

Safa Hamdare

Department of Computer Science

A thesis submitted in partial fulfillment of the requirements of
Nottingham Trent University for the degree of
Doctor of Philosophy

May 2025

Copyright Statement

Dedication

I dedicate this thesis to my husband and my daughter for their unwavering support and understanding through every challenge, achievement, and moment of stress. Your patience, love, and encouragement have been my anchor, and I could not have done this without you both.

Acknowledgements

This thesis work was conducted at Nottingham Trent University from April 2022 to February 2025, while I was a full-time doctoral research student. The research presented in this dissertation would not have been possible without the help, patience, and support of numerous individuals.

Throughout my three years of Ph.D. research, I have been privileged to have **Dr. Omprakash Kaiwartya** as my esteemed director of studies. I am deeply grateful for the time, effort, encouragement, and knowledge he has generously shared with me. Despite his many responsibilities, his office was always open for any queries. His unwavering dedication and insightful suggestions were fundamental to the success of this research.

I would like to express my sincere thanks to **Prof. David Brown** for his guidance, support, and expertise throughout the development of my thesis. His assistance during both the research and revision stages has been invaluable. His deep understanding of written expression allowed me to articulate the ideas and concepts behind this research.

My gratitude also goes to **Dr. Pratik Vyas** for his constructive feedback during supervisory meetings and annual reviews, which greatly contributed to the progress of this work.

This research could not have been completed without the support of **JMVL Ltd**, whose collaboration was vital in constructing the experimental setup and establishing the foundation for the cybersecurity analysis. A special thank you to **Manish Jugran** for his guidance and support in developing the prototype.

I am grateful to the members of my advisory committee—**Al-Hadhrami Tawfiks, Mahmud Mufti, and Yue Cao**—for their valuable feedback, constructive criticism, and time spent reviewing my work at each stage. Their collective expertise has enhanced the quality of this research.

I would also like to express my deepest gratitude to my husband, **Salman Hamdare**, and my daughter, **Inaaya Hamdare**, for their unwavering love, support, and understanding, which made the completion of this doctorate degree possible. Thank you, Salman, for believing in me and supporting my decision to pursue a Ph.D., for standing by my side when

we made the difficult choice to leave our family and our country. Your faith in me never wavered, even when I doubted myself. This journey was ours, and I am forever grateful for your sacrifices, love, and endless support.

A special thanks to my parents, **Shakoor and Shama Manikware**, for their continuous encouragement and motivation. Their sacrifices have been instrumental in shaping who I am today, and I am beyond grateful for everything they have done for me.

Lastly, I am thankful to my friends and extended family for their endless support, patience, and encouragement throughout this journey. Their belief in me has been a constant source of inspiration.

I am also grateful to Nottingham Trent University for funding my Ph.D. studies through a fully-funded scholarship. The financial support from the School of Science and Technology to attend various international conferences has been invaluable. Without this scholarship, I would not have been able to reach this academic milestone.

Safa Hamdare
May 2025

Abstract

The Electric Vehicle (EV) charging infrastructure plays a vital role in advancing sustainable transportation and achieving global net-zero emission targets. The increasing adoption of EVs has led to a rising demand for charging stations, creating both opportunities and challenges. Optimizing Electric Vehicle Charging Stations (EVCS) is crucial for scalability, reliability, and user trust while addressing security vulnerabilities. In response to the growing EV market in the UK, **JMVL Ltd (industry partner and funder of this research)** has initiated research on cybersecurity challenges in EV networks to develop a secure EV charging infrastructure. This research contributes to optimizing and securing the EV Charging Management System (EVCMS), aligning with JMVL's objectives.

This research addresses the identified challenges through six key contributions, each grounded in real-world validation and industry integration. **Unlike most prior studies that offer post-event analytics or focus on power optimization**, the 1st contribution is analysing real-time EV charging data, uncovering anomalies in connection durations, charge times, and energy consumption. These insights emphasize the need for operational enhancements and targeted security measures to strengthen the resilience of EV charging infrastructure. **Departing from conventional theoretical models or simulation-only approaches**, the 2nd contribution is the development of an EVCMS framework as a test bed. This framework integrates an application for booking and reserving charging sessions, enabling optimized charging plans, along with a charge box simulator that utilizes the Open Charge Point Protocol (OCPP) to replicate real-world charging interactions and security scenarios.

In contrast to static or centralized charging systems in literature, the 3rd contribution is scaling the framework into a distributed Hybrid-EVCMS, which dynamically allocates charging stations based on booking loads. This approach enhances scalability, efficiency, and adaptability, ensuring the system can meet real-world demands. **While prior work has acknowledged OCPP vulnerabilities, this research uniquely provides a practical security analysis using real-world test bed data**. The 4th contribution assessed Hybrid-EVCMS security against Man-in-the-Middle attacks, focusing on vulnerabilities in the OCPP communication protocol. The analysis revealed plaintext transmission of sensitive data

in OCPP, stressing the need for stronger encryption. **Unlike studies that overlook low-rate attacks**, the 5th contribution tested resilience against Slow Denial-of-Service attacks, identifying abnormal traffic patterns and resource exhaustion risks, emphasizing the need for robust mitigation strategies.

To our knowledge, no existing work has implemented proactive protocol-level defences within OCPP transactions on a live test bed. The 6th contribution is enhancing the OCPP communication protocol with improved security. Key improvements in OCPP included two-step verification for Boot Notifications, enhanced session management for Start Transactions, and integrity checks for Stop Transactions. By addressing both operational inefficiencies and security vulnerabilities, this research provides a comprehensive framework for resilient EV charging system. The key output of this research (EVCMS), has been integrated into **JMVL Ltd.'s (Industry partner) product for EV charging**.

Publications

As a result of the research presented in this thesis, the following publications have been published:

Refereed Journal Papers:

Hamdare S., Kaiwartya O., Aljaidi M., Jugran M., Cao Y., Kumar S., Mahmud M., Brown D., Lloret J. "Cybersecurity Risk Analysis of Electric Vehicles Charging Stations." **Mdpi Sensors**, 23(15):6716 (2023). https://doi.org/10.3390/s23156716

Hamdare S., Brown D.J., Cao Y., Aljaidi M., Kumar S., Alanazi R., Jugran M., Vyas P., Kaiwartya O. "A Novel Charging Management and Security Framework for the Electric Vehicle (EV) Ecosystem." **World Electric Vehicle Journal**, 15(9):392 (2024). https://doi.org/10.3390/wevj15090392

Hamdare S., Brown D.J., Cao Y., Aljaidi M., Kaiwartya O., Yadav R., Vyas P., Jugran M. "EV Charging Management and Security for Multi-Charging Stations Environment." **IEEE Open Journal of Vehicular Technology**, 5:807–824 (2024). https://doi.org/10.1109/OJVT.2024.3418201

Hamdare S., Brown D.J., Jha D.N., Jugran M., Kaiwartya O. "Cyber Defense in OCPP for EV Charging Security Risks" **International Journal of Information Security, Springer** https://link.springer.com/article/10.1007/s10207-025-01055-7.

Hamdare S., Brown D.J., Cao Y., Aljaidi M., Kumar S., Jha D.N., Alanazi R., Jugran M., Kaiwartya O. "Future-proofing Electric Vehicles Charging Management with Cyber Resilience." **Institute of Energy and Technology (IET) Journals** . [Manuscript submitted for publication.]

Refereed Conference Papers:
Hamdare S., Kaiwartya O., Jugran M., Brown D., Vyas P. "Analysis of EV Charging Infrastructure and its Impact on Public Adoption: Examining the Critical Role of Charging Stations in the Acceleration of Electric Vehicle Market Growth." **Proceedings of the 16th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '23)**, pp. 550–556 (2023). Association for Computing Machinery. https://doi.org/10.1145/3594806.3596586

Hamdare S., Brown D.J., Kaiwartya O., Cao Y., Jugran M. "MitM Cyber Risk Analysis in OCPP-Enabled EV Charging Stations." **Proceedings of the 2024 IEEE International Conference on Smart Cities (ICSC)**.

Refereed Poster:
Hamdare S., Kaiwartya O., Brown D.J.,Vyas P. "Impact of Optimization in EV Charging Infrastructure", NTU School of Science and Technology, **Annual Research Conference (STAR 2024)**

Hamdare S., Kaiwartya O., Brown D.J.,Vyas P. "Is OCPP Safe for Electric Vehicle Charging?" NTU School of Science and Technology, **Computer and Informatics Research and Industry Showcase (CIRIS 2024)**

Hamdare S., Kaiwartya O., Brown D.J.,Vyas P. "Analysis of Cyber security Threats to Electric Vehicle Charging Infrastructure", NTU School of Science and Technology, **Annual Research Conference (STAR 2023)** (selected as one of the three best posters)

Hamdare S., Kaiwartya O., Brown D.J.,Vyas P. "Cyber Risk Analysis of Electric Vehicle Charging Station (EVCS) Infrastructure" NTU School of Science and Technology, **Computing and Informatics Research Centre (CIRC 2023)**

Nomenclature

Acronyms

ACM	Availability Cost Multiplier
ACM	Availability Cost Multiplier
CAN	Controller Area Network
ChgSrv	Charging Point Server
CP	Charging Point
DCFC	DC fast charging
DoS	Denial-of-Service
DREAD	Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability
ECUs	Electronic Control Units
EVCMS	Electric Vehicle Charging Management System
EVCS	Electric Vehicle Charging Systems
EVSE	Electric Vehicle Supply Equipment
EVs	Electric Vehicles
H-EVCMS	Hybrid EV Charging Management System
IAAM	Identify, Analyze, Assess, Mitigate framework
IEC61851	International Electrotechnical Commission 61851
IEA	International Energy Agency
ISO 15118	International Standard Organization 15118
JSON	JavaScript Object Notation
kW	kilowatts
kWh	kilowatt-hours
MITM	Man-in-the-Middle
MEC	Mobile Edge Computing
OCPP	Open Charge Point Protocol
OCPP-RPC	Open Charge Point Protocol-Remote Procedure Call
PEV	Plug-in Electric Vehicle
RKE	Remote Keyless Entry
RUDY	R-U-Dead-Yet

SCMSs	Smart Charging Management Systems
SOAP	Simple Object Access Protocol
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, DoS, Elevation of Privilege
TPMS	Tire Pressure Monitoring System
TLS	Transport Layer Security
V2G	Vehicle-to-Grid
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle

Table of contents

Copyright i

Dedication iii

Acknowledgements v

Abstract vii

Publications ix

Nomenclature xi

List of figures xix

List of tables xxiii

1 Introduction 1

1.1 EVs Charging Ecosystem 1
1.2 OCPP in EV Charging Ecosystem 3
1.3 Motivation-Cybersecurity Challenges of EV Charging 4
1.4 Research Objectives 5
1.5 Contributions of the Thesis to the Knowledge 6
1.6 Scope 8
1.7 Thesis Structure 8

2 Literature Review 11

2.1 Current State of EV Charging Infrastructure 11
2.1.1 Advancement in EV Ecosystem 13
2.1.2 EV Charging Types 15
2.1.3 EV Charging Station 16

2.1.4 EV Charging Usage 19
2.2 EVCS Infrastructure 20
2.2.1 EVCS Optimization 21
2.2.2 EVCS Communication Security 21
2.3 Scalability of EVCS Infrastructure 22
2.3.1 EV charging infrastructure 23
2.3.2 EV charging control strategies 25
2.4 Cyber Vulnerability in EVCS 26
2.4.1 Infrastructure Centric 28
2.4.2 Protocol Centric 31
2.5 Cyber Security Analysis in EVCS 34
2.6 Cyber Threats to EVCS 37
2.6.1 Overview of DoS Attacks 37
2.6.2 Impact of DoS on EVCS 40
2.6.3 Overview of MitM on EVCS 41
2.6.4 Impact of MitM on EVCS 42
2.6.5 Identified Research Gaps and Methodological Baseline 43

3 Cyber Threat Analysis in EVCS 45
3.1 Experimental Methodology 45
3.2 Result and Discussion 47
3.3 Chapter Summary 49

4 EVCMS Framework 53
4.1 Background 53
4.2 System Design 55
4.2.1 EVCMS Framework 55
4.3 System Methodology 57
4.3.1 Formulating Charging Prices 57
4.3.2 Charging Price Optimization 59
4.4 System Implementation 64
4.4.1 Client Server 65
4.4.2 OCPP Chargebox Simulator 67
4.5 System Security 74
4.6 Performance Evaluation 74
4.6.1 Result Analysis 74
4.6.2 Comparative Analysis 79

4.7 Chapter Summary 80

5 Hybrid-EVCMS Framework 81
5.1 Background 81
5.2 System Design 83
5.2.1 EVCMS framework 84
5.2.2 H-EVCMS framework 85
5.3 System Methodology 86
5.3.1 CS Load Balancing 87
5.3.2 CS Allocation 88
5.3.3 Charging Price Optimization 90
5.4 System Security 94
5.5 Performance Evaluation 98
5.5.1 Case Study: Implementation 98
5.5.2 Result Analysis 102
5.5.3 Comparative Analysis 106
5.6 Chapter Summary 108

6 Slow DoS attack on H-EVCMS 111
6.1 Background 111
6.2 Experimental Methodology 113
6.2.1 SlowLoris Attack 113
6.2.2 Rudy Attack 114
6.2.3 Slow Read Attack 114
6.2.4 Range Attack 115
6.3 Performance Evaluation 116
6.3.1 Simulation Environment 116
6.3.2 Result Analysis 117
6.4 Mitigation:Detect and Stop Slow DoS attacks 124
6.5 Chapter Summary 127

7 MitM attack on H-EVCMS 129
7.1 Background 129
7.2 Experimental Methodology 131
7.2.1 OCPP Attack Model 131
7.2.2 MitM Attack Analysis on OCPP 133
7.3 Performance Evaluation 137

7.3.1 Simulation Environment . . . 137
7.3.2 Result Analysis . . . 138
7.4 Chapter Summary . . . 140

8 Cyber defense in OCPP 141
8.1 Background . . . 141
8.2 Proposed Security Framework . . . 143
8.2.1 OCPP Transaction . . . 143
8.2.2 IAAM Framework . . . 145
8.3 Identify Threat: STRIDE . . . 146
8.4 Analyse Threat . . . 153
8.4.1 Attack Tree . . . 153
8.4.2 Attack Workflow . . . 157
8.5 Assess Threat:DREAD . . . 160
8.6 Mitigate Threat: Upgrade in OCPP . . . 164
8.6.1 OCPP BootNotification Request . . . 165
8.6.2 OCPP Start Transaction Request . . . 167
8.6.3 OCPP Stop Transaction Request . . . 169
8.7 Result Analysis . . . 172
8.7.1 Environmental Setup . . . 172
8.7.2 Updated OCPP Server logs . . . 173
8.7.3 Comparative Data Evaluation . . . 177
8.8 State-of-the-Art Comparison . . . 182
8.9 Chapter Summary . . . 184

9 Conclusion and Future Work 187
9.1 Thesis Summary . . . 187
9.2 Key Outputs . . . 188
9.3 Limitations . . . 190
9.4 Directions for Further Research . . . 190

References 193

Appendix A State Life Cycle of RPC used for OCPP connection 207

Appendix B Json Schema for OCPP 209

List of figures

1.1 EVs Charging Infrastructure 1
1.2 Thesis structure 10

2.1 Global growth chart of passenger EV 12
2.2 Comparative analysis of slow and fast passenger EV charging points 13
2.3 Roadmap of Technological Enhancements in EVs 13
2.4 Usage trend of EV charging types 15
2.5 EV Private Charging Station 17
2.6 EV Public Charging Station 18
2.7 Trend of Public and Private EV Charging Infrastructure 19
2.8 EV charging infrastructures 24
2.9 Cyber-attack vulnerabilities on EV charging use cases 27
2.10 Communication in EVCS Infrastructure 28
2.11 EVCSs with Vulnerable points 29
2.12 Roadmap of EVs Communication protocol 33
2.13 Significant attacks on EVCSs 35
2.14 Classification of DoS attack 38

3.1 Threat analysis of EV charging sessions 48
3.2 Cyber-attacks on EV charging network 49
3.3 Lack of standardization issue in EV charging network architecture 50

4.1 EVCMS communication protocol architecture 55
4.2 EVCMS Framework 56
4.3 Charging plan optimization flowchart 63
4.4 Client Side User Interface for cloud ready EVCMS framework 65
4.5 Server Side User Interface for cloud ready EVCMS framework 66
4.6 Chargebox Simulator User Interface for cloud ready EVCMS framework . . 67

4.7 Integration of OCPP in Cloud ready EVCMS framework with Chargebox Simulator Functions 68
4.8 Request and Response for Connect function 69
4.9 Request and Response for Authorize function 70
4.10 Request and Response for Start Charging function 71
4.11 Request and Response for Stop Charging function 72
4.12 Request and Response for Meter Values function 73
4.13 Total charging cost comparison of original and optimized w.r.t system cases 76
4.14 Discount and profit in optimized system w.r.t system cases 77
4.15 Charging and power demand comparison w.r.t to before and after optimization 78
4.16 Charging and power demand after optimization w.r.t to Peak off/PeakOn . . 78

5.1 H-EVCMS Framework 83
5.2 EVCMS layered framework 84
5.3 H-EVCMS layered framework 85
5.4 Process of determining CS weight and assigning EV 90
5.5 H-EVCMS Framework Flowchart 93
5.6 Visualization of Compromised Session Management in the OCPP Chargebox Simulator 95
5.7 Visualization of Secure Session Management in the OCPP Chargebox Simulator 96
5.8 Server side user interface of H-EVCMS 99
5.9 Spread measure vector representation 100
5.10 CS availability among EVCMS and H-EVCMS approach 104
5.11 EV charging demand among EVCMS and H-EVCMS approach 104
5.12 EV Charging demand during Peak On/Off period among EVCMS and H-EVCMS approach 105
5.13 Power consumption of CS among EVCMS and H-EVCMS approach 105
5.14 Efficiency of CS among EVCMS and H-EVCMS approach 106

6.1 DoS attack scenario on EVCMS 112
6.2 Implementation of SlowLoris Attack 113
6.3 Implementation of Rudy Attack 114
6.4 Implementation of Slow Read Attack 115
6.5 Implementation of Range Attack 116
6.6 TCP window size in normal scenario 120
6.7 TCP window size in SlowLoris attack 121
6.8 TCP window size in rudy attack 121

6.9 TCP window size in slow read attack . . . 122
6.10 Average Delta time in normal scenario . . . 123
6.11 Average Delta time in SlowLoris attack . . . 123
6.12 Average Delta time in Rudy attack . . . 124
6.13 Average Delta time in slow read attack . . . 124
6.14 Evidence of DoS attack detected and prevented . . . 127

7.1 MitM attack in EVCS using OCPP . . . 130
7.2 OCPP Client and Server Communication in EVCS . . . 132
7.3 Network traffic capture using tcp dump . . . 138
7.4 Client Hello message analysis . . . 138
7.5 Server Hello message analysis . . . 139
7.6 Application data protocol analysis . . . 139

8.1 OCPP Communication Architecture . . . 142
8.2 Transaction message flow in OCPP . . . 144
8.3 IAAM Framework . . . 146
8.4 OCPP Attack Tree . . . 155
8.5 Attack workflow of OCPP vulnerabilities . . . 159
8.6 Flowchart to show updated OCPP Mitigation Logic . . . 171
8.7 Server log for Initial connection validation . . . 174
8.8 Server log for BootNotification . . . 175
8.9 Server log for Start Transaction . . . 176
8.10 Server log for Stop Transaction . . . 176
8.11 Server log for Stop Transaction . . . 177
8.12 (a) Conflicting Transaction of model 1 w.r.t time (b) Conflicting Transaction of model 2 w.r.t time . . . 180
8.13 (a) Conflicting Transaction of model 1 w.r.t power (b) Conflicting Transaction of model 2 w.r.t power . . . 181
8.14 Fault detection rate . . . 182

A.1 State Life cycle of OCPP . . . 207

List of tables

2.1 Comparison of Private and Public EV Charging Stations 16
2.2 Comparison of Charging Strategies 25
2.3 Impact of MitM Attack on EVCS 43

4.1 Parameters for ToU 58
4.2 Data generated by the EVCMS System 75
4.3 Result of Average charging cost comparison 77
4.4 Table based comparison with state of the artwork 79

5.1 Comparison of H-EVCMS with current state of art 97
5.2 CS Data generated by EVCMS System 102
5.3 CS Data generated by H-EVCMS System 103

6.1 Analysis of Packet Traffic Pattern 118
6.2 Analysis of TCP Window Size 120
6.3 Analysis of Average Delta time between packets 122

7.1 OCPP Messages and Their Vulnerability to Attack Vectors 136

8.1 Analysis of Vulnerabilities in EV Charging System 148
8.2 Threats Mapped to STRIDE Categories 152
8.3 OCPP Attacks mapped to STRIDE Threats 154
8.4 OCPP Attack mapped to STRIDE Categories 154
8.5 Analysis of Attack Vulnerabilities in OCPP 156
8.6 Assessing OCPP Attacks with DREAD 162
8.7 OCPP Attack Assessment: DREAD, Impact, Likelihood, and Risk 163
8.8 EV charging Reservation Details for Charge Point 178
8.9 Charge Point Transaction Data using Model 1 179
8.10 Charge Point Transaction Data using Model 2 179
8.11 State of the Art Analysis for OCPP Communication Vulnerability 183

Chapter 1

Introduction

1.1 EVs Charging Ecosystem

Energy management and transportation systems that use artificial intelligence have become more significant in modern cities as they develop major urban infrastructures. As a result, EVs will be more commonly used as part of private and public transportation fleets in the future (Fig. 1.1).

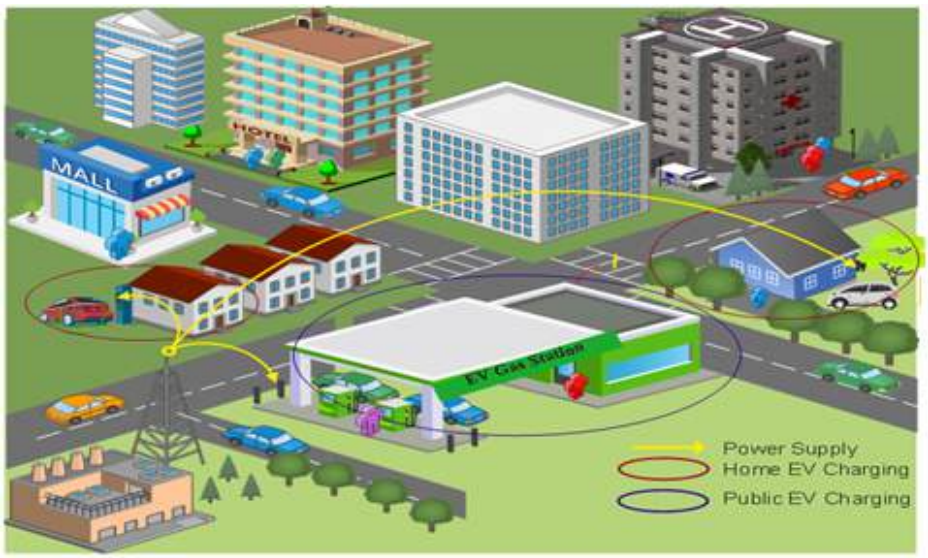


Fig. 1.1 EVs Charging Infrastructure

The government has supported various initiatives to promote the use of EVs, focusing on their contribution to a wide range of future green transportation policy goals[1]. EV usage improves air quality, reduces noise pollution, and reduces carbon emissions by eliminating

road traffic pollution. According to the Accelerating to Zero (A2Z) mandate to reduce carbon emissions, several governments around the world have taken steps to reduce fossil fuel driven vehicles. The UK government has also signed up to work towards bringing in new cars and vans with zero emissions. To help the UK government reach its 2050 "Road to Zero greenhouse gas emission" goal, EVs play an important role. In 2030, the sale of gasoline and diesel cars is proposed to be banned in the UK, with the sale of hybrid vehicles to follow in 2035[2]. This sale analysis follows the suggestion of the Committee on Climate Change[3] that the EVs market is set to reach 100% of all vehicle sales by 2035 to achieve the net zero ambition of the UK.

EVs charging ecosystem is a connected system paradigm at the core of the smart grid, consisting of a complex cyber physical system that compromises linked hardware parts, software elements, and communication protocols. The power from the grid is transferred to EVs using EVCS. The EVCS is a self-contained and Internet of Things enabled infrastructure that operates on its proprietary firmware. The public EVCS is controlled by a cloud server which allows users to be guided in the direction of available EVCS, set up and manages charge sessions and keep track of consumption statistics. Users of public EVCS communicate with the charging management system through the Internet. Usually, users schedule charging sessions, set the charging rate, begin, and end charging, and check on the status of their EVs using these services. The power infrastructure must be functional and connected to charge an EV. Because EVCS is connected into the grid and takes the necessary power from it, they pose a significant threat to the reliability and safety of the power supply. All data exchanged among the user application, EV, and EVCS must be secured to guarantee the safety and reliability of the ecosystem. Equipment manufacturers, national governments, and EVCS operators have their preferred protocols for enabling cyber security. Inconsistencies caused by a lack of standardization of protocols lead to severe cyber security problems[4].

However, EV charging station cyber-attacks have yet to be taken seriously on a wider industry scale or in government policy level. However, the smart application created for Private EV charging was found to have security issues by Kaspersky Lab[5]. The charging process for EVs might be disrupted if an attacker would gain access to the charging equipment via the WiFi connection. Schneider EVs link chargers were also found to have security issues[6], allowing remote attackers to deceive hard-coded passwords, insert malware and deactivate the charger with this vulnerability.

In this context, this research contributes towards the optimization and security of EVCMS framework. It is a part of **JMVL Ltd innovations towards making EV charging ecosystem smart and secure**.

1.2 OCPP in EV Charging Ecosystem

Communication protocols are essential for enabling interaction between EVs, EVCS, and backend platforms. Among these, the Open Charge Point Protocol (OCPP) stands out as a leading standard, ensuring interoperability and compatibility across diverse charging networks[7]. OCPP has undergone significant evolution, with its versions reflecting advancements in functionality and security. OCPP 1.5 is the earliest version set the foundation for communication in EVCS but lacked scalability and robust security mechanisms[8]. OCPP 1.6 is widely adopted due to its simplicity, flexibility, and ease of integration. This version supports WebSocket and SOAP communication protocols and introduces smart charging features like load balancing. Its simplicity has made it a preferred choice for both private and public EVCS networks, even as newer versions have been released[9].OCPP 2.0 and 2.0.1 versions offer substantial improvements in security (e.g., better encryption and authentication) and functionality (e.g., enhanced diagnostic tools and detailed transaction handling)[10]. However, adoption remains limited due to the complexity of implementation, higher infrastructure requirements, and lack of backward compatibility with earlier versions like OCPP 1.6[11].

Although OCPP 2.0 and 2.0.1 are technically superior, the transition to these versions has been slow for several reasons.OCPP 2.0 and later versions are not backward compatible with OCPP 1.6, requiring significant changes in hardware and software for seamless integration. This limitation discourages existing networks from upgrading[11]. Upgrading to OCPP 2.0 involves retrofitting or replacing legacy systems, which can be costly for operators with large-scale deployments[11]. The advanced features and security mechanisms of OCPP 2.0 demand expertise and additional resources, making it less attractive for small-scale operators or those with limited technical capacity[12]. Another limitation is the continuous need for updates and maintenance to stay aligned with evolving standards and security demands, which can be challenging for smaller operators with limited technical resources[11].

OCPP 1.6 continues to dominate due to its simplicity and ease of integration, particularly for operators looking for quick and cost-effective solutions. It allows seamless interaction with existing hardware and is suitable for environments where advanced security or sophisticated diagnostics are not immediate priorities[11], [9], Despite its widespread use, OCPP 1.6 is vulnerable to several cyber threats[13], such as spoofing, tampering, and Man-in-the-Middle (MitM) attacks, due to its weaker encryption and authentication mechanisms. These vulnerabilities require a critical examination and enhancement of its security features, as discussed in later chapters. Balancing the ease of integration of OCPP 1.6 with the advanced capabilities of newer versions like OCPP 2.0 represents a significant challenge for the EVCS ecosystem.

1.3 Motivation-Cybersecurity Challenges of EV Charging

The increasing adoption of EVs has driven the rapid expansion of EVCS, which rely on interconnected networks and protocols like OCPP to facilitate communication between EVs, charging stations, and back-end servers. In addition to ensuring secure communication, managing EV charging operations—such as real-time session monitoring, reservations, and load balancing—is crucial for maintaining efficiency[14]. However, this interconnectivity has exposed EVCS to significant cybersecurity risks, as demonstrated by several real-world security breaches. Recognizing these challenges, **JMVL Ltd sought to invest in an EV charging system** that is both smart and secure. **Funded by JMVL Ltd**, this research addresses critical cybersecurity concerns in the evolving EV landscape, contributing to the development of a robust and secure EVCMS.

Managing EV charging infrastructure presents several operational challenges. Real-time charging session coordination, efficient booking mechanisms, and dynamic load balancing across charging stations are essential for optimizing energy distribution and ensuring a seamless user experience[15]. Without an intelligent management system, charging networks may suffer from congestion, inefficient resource utilization, and service disruptions, negatively affecting both users and service providers[16], [17]. Thus, a smart charging system is necessary to integrate real-time monitoring, optimized scheduling, and secure communication protocols to enhance the reliability and efficiency of EV charging networks.

Beyond managing charging operations, EVCS face significant security challenges. Key threats include MitM attacks, where attackers intercept and alter communication between EVCS components, allowing them to manipulate data, steal sensitive information, and even disrupt charging operations[18], [19]. Another prevalent threat is Denial-of-Service (DoS) attacks, particularly Slow DoS variants. In these attacks, adversaries flood the system with excessive requests or exploit protocol inefficiencies to exhaust server resources, rendering charging stations inoperable and causing widespread disruptions. This type of attack is especially concerning in high-demand charging networks, where service downtime can lead to significant economic and operational consequences[20], [21]. Unauthorized access is another critical vulnerability, often resulting from weak authentication mechanisms or inadequate session management. Attackers exploiting these weaknesses can gain control over charging sessions, tamper with configurations, or even disrupt energy management systems. Such vulnerabilities not only compromise the safety of users but also risk damaging the reputation of charging service providers[22], [10].

The integration of smart technologies and the reliance on cloud-based backend systems further expand the attack surface, making EVCS susceptible to data breaches, information disclosure, and firmware manipulation. These risks are amplified when protocols like

OCPP transmit sensitive data in plaintext, lacking robust encryption measures to ensure confidentiality and integrity[23], [24]. Given the potential impact of these threats, robust cybersecurity mechanisms are imperative to protect the EV charging ecosystem. Measures such as end-to-end encryption, multi-factor authentication, anomaly detection systems, and integrity checks are essential to safeguard communication channels, prevent unauthorized access, and ensure the reliability and resilience of charging networks. Additionally, the adoption of advanced security frameworks, such as STRIDE and DREAD models, can help identify, evaluate, and mitigate risks more effectively, enhancing the overall security posture of EVCS[19], [25]. Addressing these cybersecurity challenges is critical to fostering user trust and ensuring the long-term success of EV charging infrastructure as a cornerstone of sustainable transportation[26].

To address the gap in existing cybersecurity approaches for EV charging systems, this research undertook a practical and layered investigation into vulnerabilities within the OCPP protocol and broader EVCS ecosystem. While many prior studies have identified threats conceptually, very few have simulated these attacks using real-world data. Our work fills this critical void by analysing real-time EV charging session data from ElaadNL [27] to validate actual cybersecurity risks, and proposing a comprehensive EVCMS framework that integrates both cost-efficiency and security enhancements. Furthermore, we introduced a Hybrid-EVCMS model to overcome the scalability and responsiveness limitations of traditional architectures, while ensuring secure and optimized resource allocation. In tackling specific threat vectors, we demonstrated and mitigated MitM attacks on OCPP communication and developed real-time defence strategies against slow DoS attacks using network behaviour analytics. Notably, we advanced OCPP itself by proposing enhanced protocol-level changes, an area that has seen little attention in prior work. These contributions not only bridge the gap between theoretical threat identification and real-world implementation but also establish a foundation for secure and resilient EV charging infrastructure.

1.4 Research Objectives

This research focuses on identifying and mitigating cyber vulnerabilities in EV charging ecosystem communication protocol OCPP, with the aim of creating a secure and scalable EV charging system for JMVL Ltd that integrates seamlessly into energy grids and urban environments.

- **Analyse EVCS for cyber security vulnerabilities:** To identify security flaws within the EVCS infrastructure and communication protocols like OCPP, ensuring the protection of data and system functionality.

- **Design and develop an EVCMS framework:** To create a robust and secure EVCMS framework that addresses the cybersecurity needs of EVCS.
- **Scale the EVCMS framework to a distributed cloud service:** To enhance the scalability of the EVCMS framework, ensuring it can support a growing number of EV charging stations on a global scale while maintaining security.
- **Test, validate, and generate experimental benchmarking performance data:** To validate the performance and security of the EVCMS framework, generating benchmark data to assess its operational efficiency and resilience.
- **Align cybersecurity strategies with JMVL's long-term vision:** To integrate security enhancements in EVCS infrastructure in line with JMVL's commitment to advancing secure and sustainable EV networks.

By meeting these objectives, this research seeks to create a secure, efficient, and scalable EV charging infrastructure that supports broader sustainability goals and contributes to the development of secure, intelligent smart city ecosystems.

1.5 Contributions of the Thesis to the Knowledge

This thesis significantly contributes to the field of EVCS by addressing critical cybersecurity challenges and proposing innovative solutions for resilient and secure charging systems. The key contributions, along with identified limitations in existing literature or industrial practice, are as follows:

- **Cybersecurity Threat Analysis:** Conducted a detailed analysis of vulnerabilities in EV charging infrastructure [28], with a focus on the OCPP protocol. **Existing studies often lack large-scale empirical evaluations using real-world operational data and tend to rely on theoretical threat models without validating them against actual charging session logs.** This thesis leveraged over 10,000 charging sessions from the ElaadNL dataset [27], including timestamps, connector IDs, energy usage, and RFID card activity, to identify unusual patterns such as very long connection times, mismatches between charging time and energy used, and abnormal meter readings. Basic visualizations and rule-based anomaly detection confirmed real-world indications of cybersecurity threats. This practical, data-driven approach fills the gap of limited empirical validation in the literature. (Published in **MDPI Sensors Journal, 2023**)

- **EVCMS Framework Development:** Designed and implemented the EVCMS framework to optimize charging costs, ensure efficient load balancing, and improve security [16]. **Current industrial systems either focus on cost optimization or load balancing but often neglect integrated security considerations in their frameworks. Moreover, many existing solutions lack thorough performance evaluation under realistic operational conditions.** This thesis addresses these gaps by combining cost, load, and security optimization in a unified framework validated with realistic scenarios. (Published in **World Electric Vehicle Journal, 2024**)

- **Hybrid-EVCMS Framework Development:** Proposed a scalable and distributed Hybrid-EVCMS (H-EVCMS) framework, combining centralized and distributed models to overcome the scalability and single-point-of-failure issues prevalent in current EVCS frameworks. **Existing literature mostly focuses on either fully centralized or fully distributed architectures, with limited attention to hybrid models that balance flexibility and robustness.** The proposed framework dynamically allocates resources, optimizes charging plans, and integrates OCPP security enhancements to address these limitations [17]. (Published in **IEEE Open Journal of Vehicular Technology, 2024**)

- **Analysis of Slow DoS Attacks:** Developed detection and mitigation strategies for slow Denial of Service (DoS) attacks targeting the H-EVCMS framework. **Current EVCS security studies often overlook slow DoS attacks, focusing mainly on high-rate flooding attacks, and lack effective detection mechanisms for low-rate, stealthy attacks that degrade service quality over time.** This research fills this gap by analyzing network-based parameters to identify and prevent such subtle attacks. (Manuscript submitted for publication in **IET ITS Journals, 2025**)

- **Analysis of MitM Attacks:** Identified and demonstrated vulnerabilities in OCPP client-server communication via a Man-in-the-Middle (MitM) attack scenario. **Despite the widespread adoption of OCPP, existing implementations frequently use outdated or weak encryption protocols, and there is a scarcity of practical studies illustrating how these weaknesses can be exploited in real attack scenarios.** This thesis highlights these critical security gaps and stresses the urgent need for upgraded encryption to safeguard communication. (Published in **IEEE International Conference on Smart Cities (ICSC), 2024**)

- **Enhanced OCPP Security:** Enhanced the OCPP communication protocol with improved security measures. **Existing OCPP standards and deployments typically lack**

multi-factor authentication and robust session integrity mechanisms, exposing charge points and management systems to various attacks such as unauthorized access and transaction manipulation. This thesis introduces key improvements including two-step verification for Boot Notifications, enhanced session management for Start Transactions, and integrity checks for Stop Transactions to mitigate these weaknesses effectively.(Published in **International Journal of Information Security, Springer, 2025**)

1.6 Scope

This research focuses on the cybersecurity aspects of EV charging infrastructure, specifically identifying vulnerabilities in the OCPP protocol to support JMVL's objective of developing a smart and secure EV charging ecosystem.

- It evaluates real-time data to identify anomalies and propose targeted security measures.
- The study proposes and implements centralized and hybrid frameworks (EVCMS and H-EVCMS) to optimize charging operations and improve load balancing among charging station respectively.
- Threat detection and mitigation strategies, specifically for slow DoS and MitM attacks, are explored and validated.
- Updates to the OCPP framework include modifications to the Boot Notification, Start Transaction, and Stop Transaction message parameters to enhance security and reliability.

1.7 Thesis Structure

The thesis is organized as follows:

- **Chapter 1: Introduction:**Provides an overview of the EV charging ecosystem, its importance in achieving sustainability goals, and the cybersecurity challenges associated with EVCS.
- **Chapter 2: Literature Review:** Reviews recent advancements in EVCS, the need for optimization, security vulnerabilities, and communication protocols. Explores the transition from centralized to hybrid frameworks and analyses major cyber threats.

- **Chapter 3: Cyber Threat Analysis in EVCS:** Presents a detailed analysis of cybersecurity risks in EVCS using real-time data from charging sessions.
- **Chapter 4: EVCMS Framework:** Introduces the EVCMS framework, highlighting its design, implementation, and performance evaluation in a simulated environment.
- **Chapter 5: Hybrid-EVCMS Framework:** Proposes the H-EVCMS framework, integrating centralized and distributed management models. Details algorithms for load balancing and cybersecurity integration with OCPP.
- **Chapter 6: Slow DoS Attack on H-EVCMS:** Focuses on analysing and mitigating slow DoS attacks targeting H-EVCMS. Evaluates detection and prevention strategies.
- **Chapter 7: MitM Attack on H-EVCMS:** Analyses MitM vulnerabilities in OCPP communication. Demonstrates risks and recommends upgrading encryption protocols.
- **Chapter 8: Cyber Defence in OCPP:** Identifies threats using STRIDE, maps attacks with attack trees, and assesses risks using DREAD. Proposes and evaluates security enhancements for OCPP.
- **Conclusion and Future Work:** Summarizes the research findings, contributions, and limitations. Outlines future directions for enhancing cyber security in EV charging infrastructure.

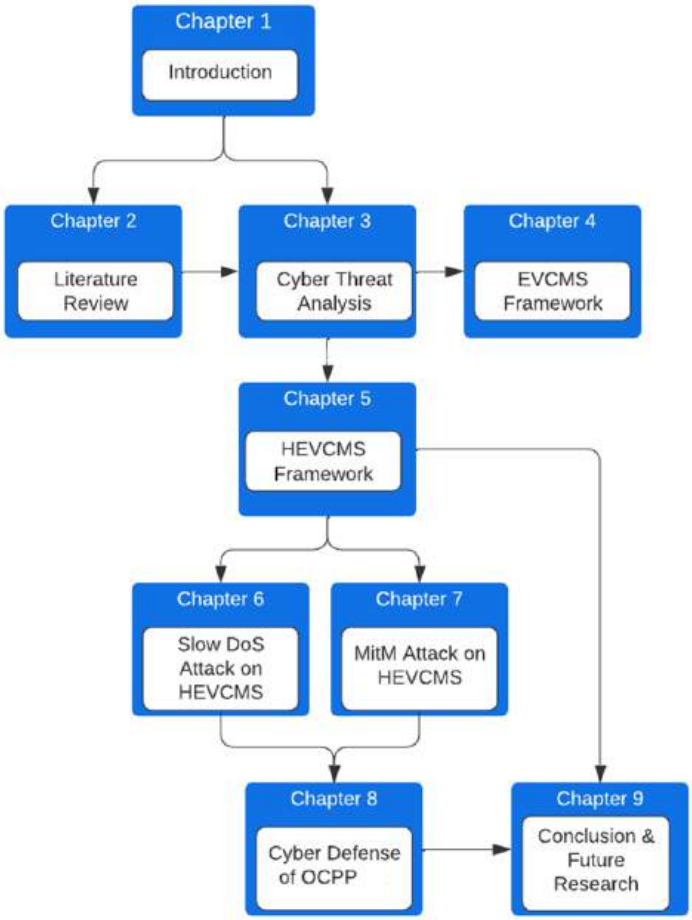


Fig. 1.2 Thesis structure with chapters and their respective dependencies

Chapter 2

Literature Review

This chapter provides a comprehensive literature review of the EVCS network. It begins in **Section 2.1** with an overview of the current state of EV charging infrastructure, followed by **Section 2.2**, which discusses recent advancements in the EVCS network. This section also explores various EV charging types, charging stations, and EV usage patterns to establish a background for the research. Next, the chapter highlights the need for optimization in EVCS, emphasizing the rationale behind developing a smart EV charging infrastructure. It further examines the communication protocols used in EVCS, offering insights into their security aspects.

Considering the scalability of EVCS, **Section 2.3** explores different EV charging infrastructures and control strategies, explaining the shift from a centralized to a hybrid infrastructure. The chapter then addresses cybersecurity concerns in **Section 2.4**, analysing vulnerabilities within both the infrastructure and communication protocols. **Section 2.5** provides an overview of significant cybersecurity attacks on EVCS, followed by **Section 2.6**, which concludes with an analysis of these vulnerabilities.

2.1 Current State of EV Charging Infrastructure

Over the past decade, EVs have experienced significant growth and adoption around the world. This growth has been driven by several key factors, such as advancements in battery technology, government policies aimed at reducing carbon emissions, and consumer demand for more sustainable transportation options. According to the International Energy Agency, there were over 10 million electric cars on the road in 2020, up from just 17,000 in 2010, representing a phenomenal compound annual growth rate of over 60% over the past decade [29] as shown in (Fig.1.2).

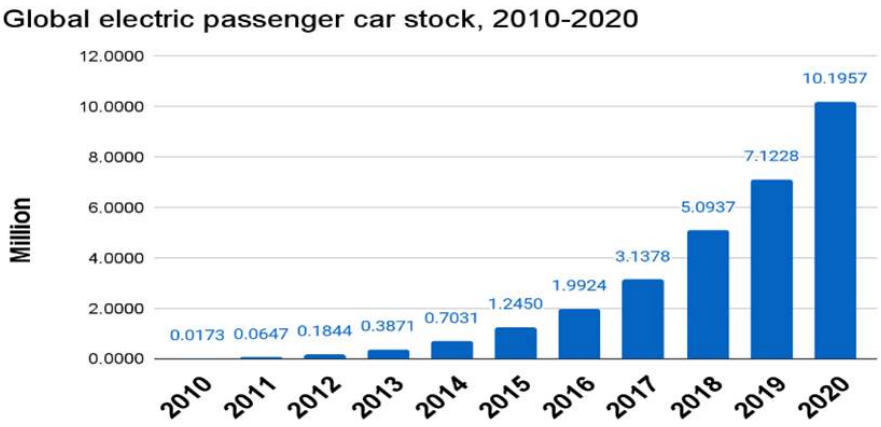


Fig. 2.1 Global growth chart of passenger EV

As EVs have become more popular, there has been a corresponding increase in the installation of EVCS in public places such as parking lots, highways, and commercial areas. Availability of charging infrastructure is critical to the widespread adoption of EVs, as it provides drivers with the convenience and confidence to make the switch from traditional gas-powered vehicles. The current state of EV charging infrastructure varies across different regions and countries. In some areas, the infrastructure is well-established, while in others, it is still in the early stages of development. According to the International Energy Agency (IEA), the number of public EV charging points worldwide reached over 1.4 million in 2020, up from 1 million in 2019 [29]. In Europe, the number of public charging points increased from around 100,000 in 2017 to over 225,000 in 2020, with a higher concentration of charging points in countries such as Norway, the Netherlands, and Germany [29]. In Asia, China leads the way with the largest number of public charging points, accounting for over 80% of the total number of charging points globally [29]. In 2020, China had over 1 million public charging points, compared to just 300,000 in Europe and 200,000 in the United States. Following bar chart shown in (Fig. 13) gives a comparative analysis of slow and fast light duty EV charging points available from 2015 to 2020 for major countries like China, Europe, United States, and rest of the world [29].

However, the availability and accessibility of charging infrastructure remain key factors affecting the widespread adoption of EVs. There are several other factors that can influence the use of EV charging infrastructure, including: (1) Advancement in EV Ecosystem, (2) EV charging types (3) EV charging Station Types (4) EV charging usage Pattern.

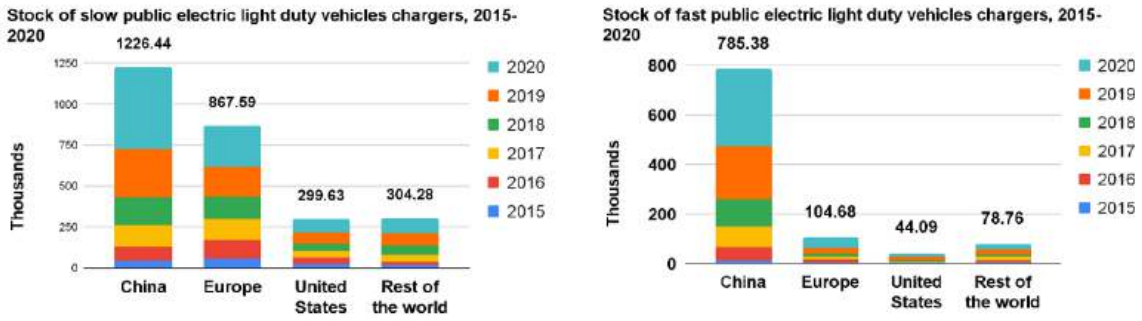


Fig. 2.2 Comparative analysis of slow and fast passenger EV charging points

2.1.1 Advancement in EV Ecosystem

The use of EVs has yet to be common in the car industry. Some people are worried about running out of power while driving because there are not enough places to re-charge their vehicles. That is why the government offers money-saving incentives like tax breaks and rebates to encourage people to buy EVs. Advancements in EVs are highlighted in (Fig. 2.1), which explains various aspects of EV technology and infra-structure from 2010 onwards till date with significant years highlighted.

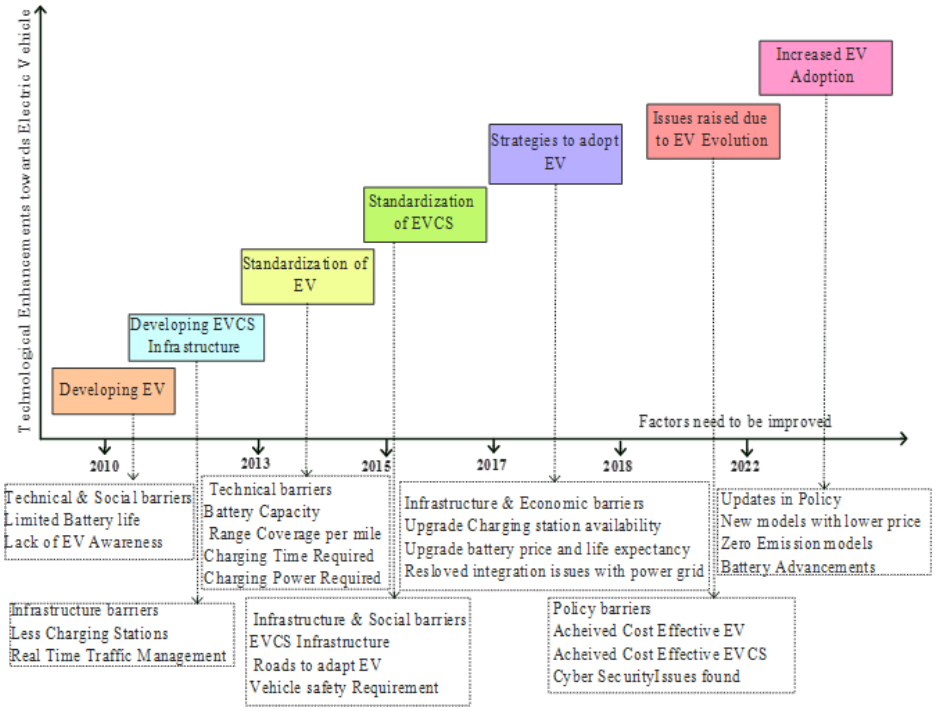


Fig. 2.3 Roadmap of Technological Enhancements in EVs

The decreasing cost of EVs can be attributed to the advancements in technology utilized in their production and the scaling up of their manufacturing processes. This advancement has helped change people's opinions about EVs. The technological revolution in battery development has been identified as a potential catalyst for promoting EV adoption[30]. The cost of batteries has also gone down, which has made up about a quarter of the total cost of an EV. EVs have problems, like needing more charging stations, which is challenging for people who own them [31]. Efforts to address this challenge necessitate the establishment of an efficient and intelligent net-work of charging stations. Leveraging algorithms to optimize charging station allocation for individual users could significantly reduce wait times and boost productivity.

Additionally, promoting Private charging can aid in prolonging battery life while also contributing to grid stability [31]. This study compares EVs' extant production and testing with new designs currently in the prototype phase [32]. Researchers have examined how far an EV can go on one charge, how big the battery is, how powerful the charger is, and how long it takes to charge the vehicle. They also discussed the charging stations' and vehicles' specifications and how this affects the power grid. An-other study also looked at the safety requirements for the vehicles and the roads they drive on [33]. The recommendation is to build more charging stations quickly to cope with the growing number of EVs on the road. The study found that different chargers are available for EVs with different power levels and interfaces. The common barriers to EV station advancement include issues such as cost, regulatory permissions, and theft. Recommendations for overcoming these hurdles were made, including in-creasing the attractiveness of EV ownership by making EV charging stations accessible to the public. Proper placement of charging stations to ensure widespread EV adoption is critical to mitigate some of the inherent risks associated with this technology [34]. Some of these issues include factors such as battery price, battery life expectancy, the availability of charging stations, integration issues with the smart grid, range, and coverage. This research was carried out from three perspectives: charging stations, batteries, and vehicle types. In recent work [35] the difficulties that have arisen throughout the evolution of EVs in recent years are examined. The total expense of owning a battery-operated EV has decreased significantly due to lower installed battery prices, and this trend is expected to continue. The efficient and cost-effective deployment of charging infrastructure is significantly more critical for the long-term growth of EV ownership.

2.1.2 EV Charging Types

The main factor responsible for EV adoption is the types of charging used. The three main types of EV charging are Level 1, Level 2, and DC fast charging (DCFC), and they have different impacts on the EV charging experience.

1. Level 1 charging uses a standard household outlet (120 volts) and provides a slow charge rate of around 2-5 miles of range per hour of charging [36]. This type of charging is best suited for overnight charging at Private and is convenient for EV owners with low daily driving needs.

2. Level 2 charging uses a dedicated charging station that operates on 240 volts and provides a faster charge rate of 10-60 miles of range per hour of charging, depending on the vehicle and the charging station's power output [36]. Level 2 charging is commonly found in public locations such as shopping centres, work-places, and public parking facilities, and it is suitable for daily charging needs.

3. DC fast charging (DCFC) is the fastest type of EV charging. It can provide up to 80% of a vehicle's battery capacity in around 30 minutes, depending on the vehicle and the charging station's power output [37]. DCFC stations are typically located along highways and major travel routes, making them ideal for long-distance travel.

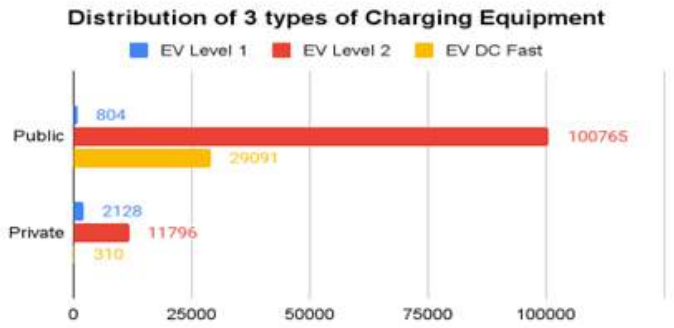


Fig. 2.4 Usage trend of EV charging types

The EV charging industry trend has been toward expanding Level 2 and DCFC charging infrastructure. This trend is because Level 2 charging provides a faster and more convenient charging experience than Level 1 charging, and DCFC stations are essential for long-distance travel and reducing range anxiety. This trend can be explained by the analysis shown in (Fig. 2.2).

In recent years, many public charging stations and auto makers have invested heavily in DCFC infrastructure to make long-distance travel in EVs more practical and convenient.

As a result, the number of DCFC stations has grown significantly, making it easier for EV owners to travel longer distances without worrying about running out of charge. The impact of Level 1, Level 2, and DCFC charging on the EV charging experience varies based on charging speed, charging location, and the driver's charging needs. The EV charging industry trend has been toward expanding Level 2 and DCFC infrastructure, as these types of charging provide a faster and more convenient charging experience for EV owners.

2.1.3 EV Charging Station

An EVs charger open to the public is called a "public charging station". A charging station designed for residential use is typically installed permanently at one's home, with the user only charged for power consumed. Due to an increase in the use of EVs worldwide, more and more private and public charging stations are being built. (Table.2.1) summarizes the pros and cons of the presented EV charging station types.

Category	**Private EVCS**	**Public EVCS**
Ownership	Private	Public
Access	Limited to owners	Open to public
Cost	Owner pays for charging	May charge a fee
Convenience	Convenient for owners	May be less convenient
Advantages	Convenient for owners	Promotes EV adoption
Disadvantages	Not accessible to the public	Less convenient, limited maintenance
Gaps to Improve	Increase public awareness, incentives for installations	Increase number of stations, improve reliability

Table 2.1 Comparison of Private and Public EV Charging Stations

As illustrated in (Fig. 2.3), the private charging use case describes the home charging use case where EVs can be recharged at home. It is safer and less hazardous to charge an EV at home since the EV is connected to an established network, making it more secure. Charging an EV at home might have some positive effects as well as limitations as follows:

1. More convenient: Installing a Private charging station presents benefits in terms of saved time and reduced reliance on gasoline. Charging an EV at home can eliminate waiting in parking areas for charging opportunities.

2. Increased savings: According to US Department of Energy estimates, charging an EV with a 33-kWh capacity at 0.13 cents per kWh costs only $0.04 per mile [38].

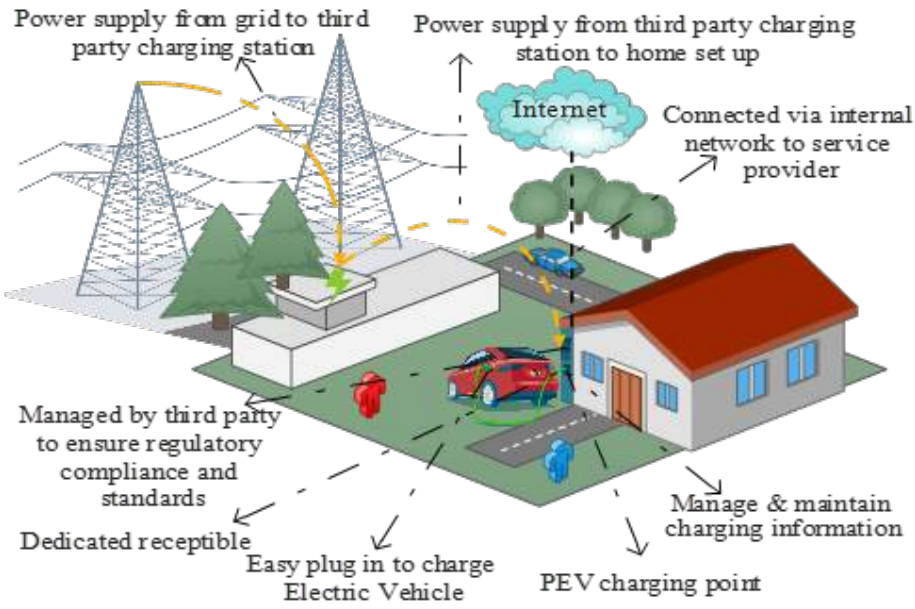


Fig. 2.5 EV Private Charging Station

3. Private charging saves money by avoiding high public charging rates. Residential EV charging stations pay for themselves via cost savings.

4. Greater home value: EVCSs could increase property value and save time/costs. Home buyers with EVs seek residences with pre-installed EV charging, boosting demand. This setup could lead to faster sales at higher prices, recouping the initial EVCS investment.

5. Less wear and tear and safe: Fewer people use it, which reduces wear and tear and repair costs. We know who uses the chargers despite not being connected to any external network.

6. Longer charging time: Many owners wish to keep the Level 1 charger with their EVs. These chargers charge slower than public ones. A Level 2 charger helps speed up charging. Level 2 chargers are faster than Level 1 chargers, charging batteries up to 30 to 44 miles per hour [38].

7. Higher Upfront Cost: If they do not have the money to pay for their own, they will have to rely on public stations. In contrast, individuals with the money to do so will gain long-term savings and increased property value by charging at home.

The public charging use case illustrated in (Fig. 2.4) presents the second use case in which EVs can be charged in public charging stations. This use case is more vulnerable to assault since it is connected to an external network and can be readily altered.

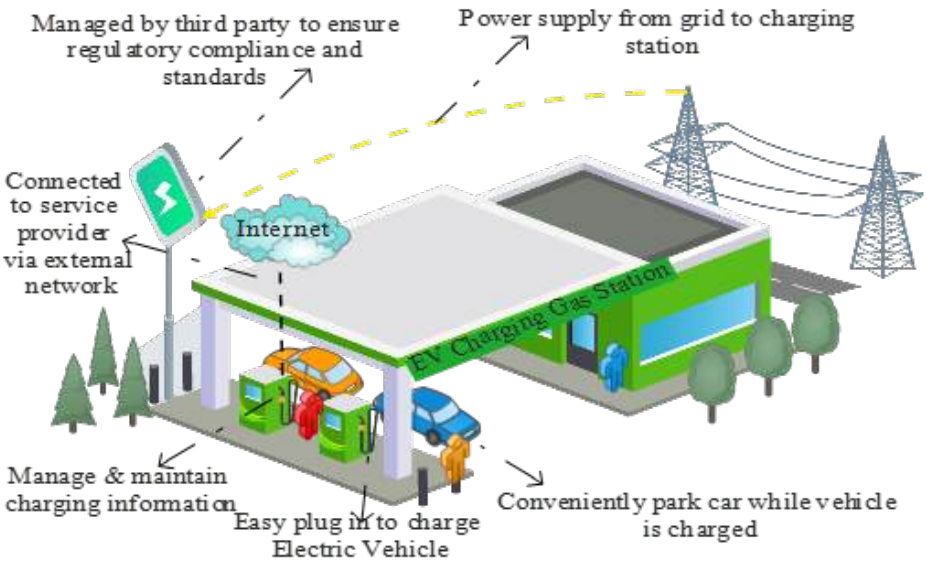


Fig. 2.6 EV Public Charging Station

1. Using public EV charging instead of home infrastructure has the following advantages and some associated challenges:

2. Less investment cost: Public EV charging stations do not require substantial initial expenditure. Since the company paid for and built the station, individuals only need to pay for the electricity they use for charging their EVs.

3. Economical for the public: Public EVCSs may offer faster charging than most homes can afford. Some public chargers are ideal for needing to charge a vehicle urgently during a journey.

4. Longer public waiting time: When a person arrives at a charging station, they may have to wait a long time. Thus, Private charging can be preferred over public charging to minimize long delays.

5. Inconvenient searching for EVCSs: Finding one may be difficult if someone is unfamiliar with their neighbourhood's charging stations. Even if a person has discovered all the public charging stations on their normal route, having an at-home charger is a smart choice.

6. Potential damage to battery: Utilizing Level 3 fast chargers may cause an EV's battery to deteriorate more quickly than with Level 1 or Level 2 chargers [38]. For those who want to increase their battery life, it is advisable to use rapid charging stations rarely and home chargers frequently.

2.1.4 EV Charging Usage

Private charging is the most convenient and cost-effective way to charge an EV. Many EV owners install a Level 2 charging station at home, allowing them to charge their vehicle overnight while sleeping. So, they can wake up to a fully charged vehicle each morning and start their day without worrying about finding a charging station. However, in some cases, drivers may need access to Private charging due to a lack of dedicated parking or the inability to install a charging station. In these cases, public charging infrastructure becomes essential for EV adoption. The U.S. public and private EV charging infrastructure graph on the alternative fuels data centre website visually represents the growth of EV charging infrastructure in the United States over time [39].

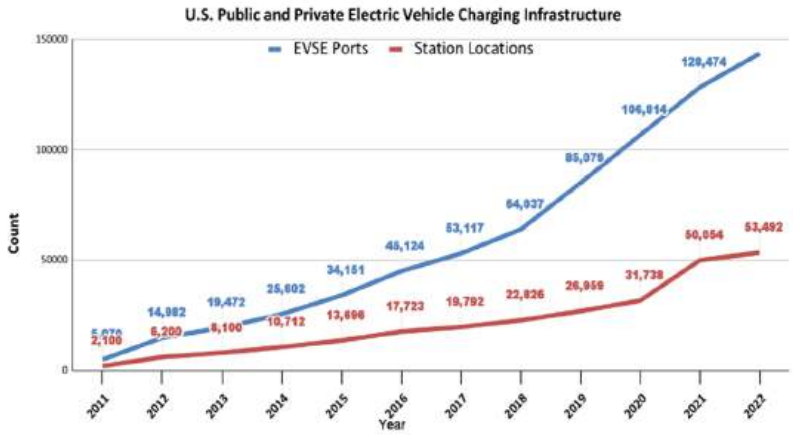


Fig. 2.7 Trend of Public and Private EV Charging Infrastructure

The graph (Fig. 2.5) shows two lines, one representing the number of public EV charging stations in the United States and the other representing the number of private EV charging stations. The data in the graph covers the period from 2011 to 2021[39]. The trend in the graph shows a steady increase in the number of EVCS in the United States over the years. Public charging stations have grown significantly since 2011, from just over 1,000 to over 40,000 stations as of 2021. Similarly, private charging stations increased from just over 600 stations in 2011 to over 10,000 in 2021.

The growth in EV charging infrastructure can be attributed to several factors, including government incentives and policies that encourage the adoption of EVs, advancements in technology that have made EVs more affordable and practical for consumers, and the increasing demand for sustainable transportation options. As the number of EVs on the road continues to grow, the need for charging infrastructure will also continue to in-crease. The trend in the graph shows that the United States is making significant progress in expanding its EV charging infrastructures, which is essential for promoting the widespread adoption of EVs and reducing carbon emissions in the transportation sec-tor.

Public charging stations are often located in areas where drivers spend time, such as workplaces, shopping centres, and public parking facilities. This convenience makes it easy for drivers to top off their vehicle's charge during the day while running errands or working. Additionally, some drivers may need to use public charging infrastructure for long distance travel or to supplement their Private charging. For example, drivers may need to use fast charging stations located on highways for quick charging during long road trips. In short, while Private charging is often considered the most convenient and cost-effective option for EV owners, public charging infrastructure is essential for EV adoption, particularly for drivers who do not have access to Private charging or need to supplement their charging needs for long-distance travel.

2.2 EVCS Infrastructure

Several research studies and commercial implementations, are aiming at modernizing the charging procedure for EV. Enhancement in fast charging technology was incorporated to reduce charging time and improve EV user convenience [40]. The EV user experience was also improved by the means of user-friendly mobile interface and customer oriented features to increase user demand [40]. The public found inconvenience with wired charging infrastructure so implementing wireless charging technology to eliminate the need for cords and improve user convenience was a progress[41]. There were many research showing problems of low EV battery in long distance journeys. This therefore prompted the use of advanced battery management system for improved and longer life batteries [42].Optimized charging station and cost effective charging methods were adapted to minimize production costs and maximize profitability [43]. For instance, [44] suggests some sustainably rechargeable options in an effort to minimize the carbon emissions and support cleaner transportation system. In [45],authors suggested charging stations to be integrated with smart grid in order to enhance power usage optimisation and address imbalances in networks. All these studies and implementations demonstrate the importance of improving the EV charging process to

provide a more convenient, efficient, and cost-effective EV charging experience. Through this proposed work the intention is to improve the EV charging with improved security.

2.2.1 EVCS Optimization

Several studies and commercial implementations have focused on enhancing the real-time operation as well as optimizing EV charging plans.Authors in [15] have used real-time data and optimization algorithms which lead to improved performance by dynamically adjusting the EV charging prices.As indicated in [46], the predictive maintenance algorithms may help avoid possible charging station failure and enhance general reliability.As the number of EVs is increasing more EVCS and better load balancing schemes are needed, to manage the charging needs. Authors in [47], have provided Smart routing of EVs for Load Balancing in Smart Grids. It also helps in balancing the charging demand on different charging stations which increases efficiency thereby reduces or limits grid imbalances. The emergence of smart EVCS in recent year have raised various privacy and security issues related to the smart grid which may generate legal challenges [48]. These challenges affect how private sensitive information is transmitted among EVs, EVCS and smart grid. Therefore, securing communications in EVCS infrastructure is crucial.

2.2.2 EVCS Communication Security

Safe communication channel between the charging station and the cloud-based management system should be ensured, preventing malicious users. Moreover, there should be use of secure authentication processes to allow only legitimate users to access charging facilities. Encryption of sensitive information like billing information and energy data could prevent data leakage and unwanted data disclosure. Therefore, it is crucial to have a knowledge of various EVCS communication protocol utilized for effective EV charging infrastructure.

A connection is needed between an EV's charger and the EV itself. Commonly used EV communication standards are International Standard Organization 15118 (ISO 15118), International Electrotechnical Commission 61851 (IEC61851), and OCPP [49]. These standards have different uses and applications. ISO 15118 is a communication protocol to secure and streamline communication between the EV and EVCS. It's used to confirm the EV's identity and manage charging details [50]. IEC61851 defines the general requirements, testing methods, and test procedures for EV charging systems and equipment [51]. It also defines AC and DC charging modes and the requirements for each mode. OCPP is an open-source communication protocol that is widely used in EVCS to communicate with EV charging stations [10]. It is a widely adopted communication protocol for EVCS as it is an

open standard [8], and can be used by any EV charging station manufacturer. OCPP has been designed with simplicity and efficiency. It is built on a client-server architecture. OCPP can be used to exchange basic information like charging station availability, charging status, and the real time pricing [10].

One of the main differences between these protocols is that IEC61851 is focused on the technical aspects of the EV charging equipment and the communication between the EV and EVCS, whereas OCPP and ISO15118 are more focused on the communication protocols and the information exchanged between the EV and EVCS [49]. Both standards are important to ensure the compatibility and safety of the EV charging systems. OCPP in EVCS provides a convenient, efficient, and cost-effective solution for managing and controlling EV charging process [8]. OCPP also ensures security in EVCS by providing access to real time information. As the objective of the proposed system is to ensure real-time performance along with security, we intend to opt for OCPP communication protocol for this design. There are certain measures that need to be adapted to ensure security in EVCS with the use of OCPP.

These studies and implementations demonstrate the importance of optimizing EV charging plans, improving real-time performance, and ensuring security in EV charging management systems. The proposed system aims at optimizing the EV charging plans with real-time optimization in compliance with industry standards and providing improved security by incorporating secure communication, customer authentication, and Data encryption.

2.3 Scalability of EVCS Infrastructure

The evolution of EV charging infrastructure is the result of a diverse range of research work, encompassing both technological advancements in EV and enhancements made in CS infrastructure. In [52], have provided insights on enhancements made in EV technology, charging standards and power grid integration. They have also highlighted the challenges and opportunities related to the increasing demand in EV. This focus has drawn the attention of researchers, as evidenced by [53], proposes the integration of wireless charging technology in EV charging infrastructure. By doing so, they examine the potential benefits and obstacles of incorporating wireless charging within the EV ecosystem. In [54], authors have proposed a centralized management framework for streamlining EV charging processes. It aims to enhance efficiency and optimization by central coordination. But this solution addresses challenges of load management aligning with issues of managing too many EVs at the same time. On the other hand, authors of [55] has also introduced a centralized charging approach to manage large number of EV at same time while maintaining grid balance. This strategy

seeks to optimize EV charging schedules centrally, showcasing its potential to outperform the proposed framework in [54].

The system presented in [56] addresses the growing need for efficient charging solutions in EV industry. By leveraging Mobile Edge Computing (MEC) architecture, the system provides intelligent EV charging recommendations while minimizing network traffic overhead and ensuring data security. It operates through a centralized cloud server that analyses CS data to predict availability and utilizes road-side units as MEC servers for disseminating information to EVs. On the other hand, [57] delves into the impact of Plug-in Electric Vehicle (PEV) charging on power system dynamics. It proposes an optimized solution for dynamically allocating available power to manage PEV charging efficiently, employing a stochastic decentralized control strategy.This research offers valuable insights into how distributed charging practices influence the grid. To make infrastructure more reliable, [58] proposes a distributed failure-tolerant optimization strategy for EV charging. This approach makes sure each EV charging plan schedule is reliable even in the event of charging infrastructure failure. In [59], authors have introduced a distributed cooperative method for EV charging, allowing EVs to collaboratively optimize their charging schedules considering user preferences and grid constraints. This cooperative approach is a blend of centralized and distributed, proving to be more efficient and sustainable. The above enhancements made by researchers in EV charging infrastructure, necessitates a detail understanding of different EV charging infrastructures and charging control strategies used in them.

2.3.1 EV charging infrastructure

EV charging infrastructure has evolved based on different user constraints, CS and grid requirements. Among these, three common approaches are: centralized, distributed, and hybrid charging infrastructure [60].(Fig.2.6) shows the different EV charging infrastructure.

1. **Centralized charging:** In centralized charging, EV CSs are positioned throughout the grid with careful planning. A central controller is employed to gather data from individual EVs. This data is then processed centrally to create an optimal solution that considers both grid requirements and user constraints. This approach involves making informed decisions about the optimal charging rates and schedules for the EVs. While efficient for managing optimized charging plans, the challenge arises as the number of EVs increases [60]. Also, this approach yields efficient solutions by considering complete system data at one place. But also raises concerns related to system failure due to a single point of control.

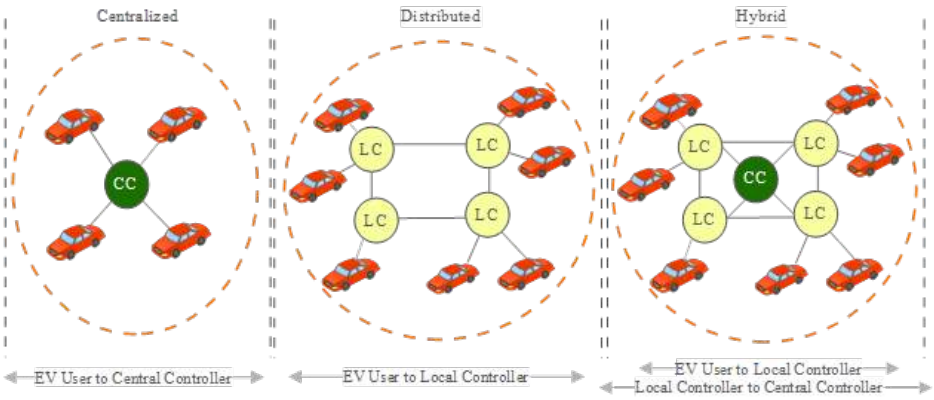


Fig. 2.8 EV charging infrastructures

2. **Distributed charging:** Contrasting the centralized approach, distributed charging empowers EV users with more autonomy. Distributed charging allows individual EV owners to independently determine their own charging schedules [60]. Each EV acts as an autonomous decision-maker coordinating with local controller, considering factors such as electricity prices, availability, and personal preferences. This approach is practical and scalable, as it reduces the need for centralized decision-making and extensive communication networks. However, since each EV acts individually, the system may not always achieve globally optimal solutions, particularly when there's a lack of complete information [60]. Distributed control is well-suited for scenarios where EV users have diverse charging needs and where user autonomy is prioritized.

3. **Hybrid charging:** A hybrid charging strategy strikes a balance between a centralized and distributed approach by employing multiple levels of control entities [60]. The hybrid structure consists of a central controller controlling multiple local controllers, and each local controller coordinating a specific group of EVs. These controllers collaborate to influence each other's decisions and manage EV charging schedules. Hybrid control reduces the need for network-wide communication, increases scalability, and enhances system resilience. This approach is especially useful when dealing with large numbers of EVs and potential communication failures.

Each of these models offers unique advantages and considerations, shaping the way EVs are charged and integrated into the existing energy ecosystem, as shown in (Table.2.2). To design an optimal charging infrastructure that considers the optimization of charging plans, user preferences and load balancing, then hybrid approach looks more suitable. As these features align with the proposed framework, a hybrid EV charging infrastructure was selected.

Table 2.2 Comparison of Charging Strategies

Aspect	Centralized	Distributed	Hybrid
Decision Authority	Central entity makes decisions	Individual EV owners decide	Combination of both approaches
Load Balancing	Efficient load balancing	Limited load balancing	Balanced load distribution
Scalability	May face scalability challenges	Highly scalable	Optimized scalability
Optimization	Optimized global solutions	Limited optimization	Hybrid optimization
Grid Impact	Concentrated load on the grid	Distributed load on the grid	Balanced grid impact

2.3.2 EV charging control strategies

EV charging control strategies manage and optimize the process of charging EVs within EV infrastructure. They encompass a wide range of methods, ranging from simple time-based charging to more complex algorithms that consider real-time data and dynamic grid conditions. So in order to optimize charging plans and load balance the power consumption of CS, its essential to understand these strategies. There have been lot of research carried out in this direction to improve EV charging infrastructure w.r.t to charging dynamics.

Most of the research in EV charging control strategies focuses on minimizing the charging price by optimizing the charging plans.In [61], authors have intelligently adapted charging time and associated price to minimize charging costs. By doing so they aim to minimize the strain on grid during peak hours. Also in [62], authors have tried to implement the same strategy but for multiple EVs within a fleet. Here fleet operators minimizes the operational cost and ensures there is no overloading of local grid. Other research is indirection of load balancing the grid infrastructure. Authors in [63] discusses a coordinated charging approach involving charging multiple EVs with intention to prevent overloading. By spreading out the charging load and timing, this strategy mitigates potential strain on the grid's infrastructure. The authors in [64] develop a queuing based load-balancing model for public EV CSs, enabling EVs to communicate their charging status with the grid to minimize waiting times and optimize efficiency. In [65], authors have proposed an approach that encourages users to shift their charging time preference to off peak hours. This helps balance the load on the grid and can lead to more efficient energy distribution. Whereas in [66], authors focuses on signals from power grid. If the grid is facing high demand, EV chargers can temporarily increase their charging rates or pause charging for sometime. This helps manage peak loads and maintain grid stability during Peak demand hours.

Some of the research focuses on improving grid performance to improve energy ecosystem. In [67], authors have used a strategy to charge EVs when renewable energy sources like solar and wind are generating excess power. By doing so there would be less dependency on on-renewable sources and contributes to a more sustainable energy ecosystem. On the other hand,authors in [68] focuses on Vehicle-to-Grid (V2G) Charging. This enables EVs to not only draw energy from the grid but also send energy back into the grid.This bi-directional flow allows EVs to act as mobile energy storage units. They can store excess energy when the grid has surplus capacity and feed it back into the grid when demand is high, aiding in load balancing and providing grid support services. There is also research done to improve the overall user experience by introducing wireless concept to EV charging infrastructure. Authors in [69], introduces a cutting edge approach that enables EVs to charge wirelessly while in motion, using embedded charging infrastructure along roadways.As EVs drive over these charging lanes, they receive power through induction. This innovation aims to eliminate the need for scheduled charging stops and extend the range of EVs.

Each of these strategies offers unique advantages and challenges, contributing to the evolving landscape of EV charging control schemes. As technology and research progress, these approaches continue to shape the integration of EVs into our EV charging infrastructure. The proposed system incorporates some of the above charging control strategies to support charging cost minimization, load balancing by shifting to Peak off time and coordinated charging approach to spread out charging time and load for better performance.

2.4 Cyber Vulnerability in EVCS

There is growing concern that EVs may adversely affect power grid reliability due to their unpredictable charging behaviour. Some potential attacks on the PEV charging systems have been addressed [70] and are discussed above. The potential negative effects of PEVs integrated components, including a risk to the general public's safety for nearby residents and those operating nearby vehicles, have been highlighted. The prevalence of cyber attacks against EVs is growing. More charging stations provide more potential targets for cyber attacks on EVs. A hacker could exploit many security holes across brands to gain unauthorized access to user accounts, disrupt charging, or even transform one of the chargers to gain access to an owner's home network [71]. The battery management system of an EV is vulnerable to attack if an attacker gains access to it through a hacked website or by downloading malware to the EVs systems. Researchers have addressed how attackers might harm EV batteries by altering the charging current and avoiding safeguards [72]. Infrastructure for EVs is inherently vulnerable since IoT devices use various web-based communication and application

services [4]. In haste to get their products to market quickly and at a lower cost, operators and manufacturers sometimes compromise on security. The safe functioning of EVs charging is deemed crucial for the security of the new smart grid because of the interconnected nature of the EVs infrastructure and the power grid. The United States Department of Transportation has identified the following as some of the security concerns related to the EV charging ecosystem, and these concerns are discussed in detail [70]:

1. The EVs sector needs safer software development practices.
2. Communication between EVs and EVCSs is not standardized on a secure communication protocol,
3. Insufficient data integrity controls and cyber security monitoring systems exist.
4. While the physical characteristics of EVCSs vary, many are easily accessible and modifiable.

A proposal from the National Institute of Standards and Technology (NIST) identifying attacks [73] on the EVs infrastructure is summarized in (Fig.2.7) as follows:

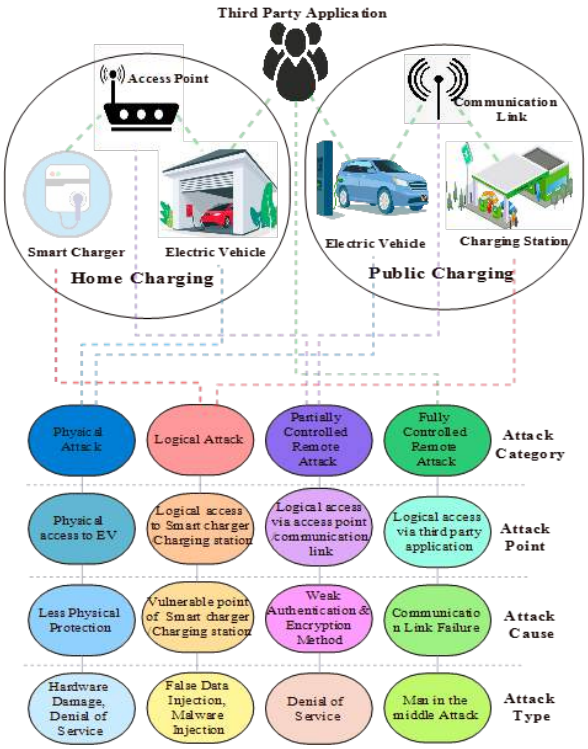


Fig. 2.9 Cyber-attack vulnerabilities on EV charging use cases

1. Physical Attack: Due to its lack of physical protection, EVCSs are vulnerable to attacks that disable the system, steal power, or infect it with malware via accessible USB ports.

2. Logical Attack: The EVCSs are compromised in such an attack by exploiting a flaw in the firmware, which allows the attacker to acquire logical access to the system. Some suppliers' firmware upgrades, including those released by Schneider Electric, may be downloaded from the internet and dissected by attackers to discover security flaws and potential entry points [74]. Kaspersky laboratories could also crack the Charge Point home charger firmware by local attack [75].

3. Partially controlled Remote Attack: Local Area Networks at charging points may be used by attackers to gain access to the EVCSs. Weak authentication and outdated encryption methods are typical of such systems. Over the charging line, the EVs and EVCSs communicate via a series of protocols, which leaves the system open to attack.

4. Fully controlled Remote Attack: Users of EVs interact with the EVs management system through an online user interface. Such interaction opens potential vulnerabilities, whether a website or a mobile application.

2.4.1 Infrastructure Centric

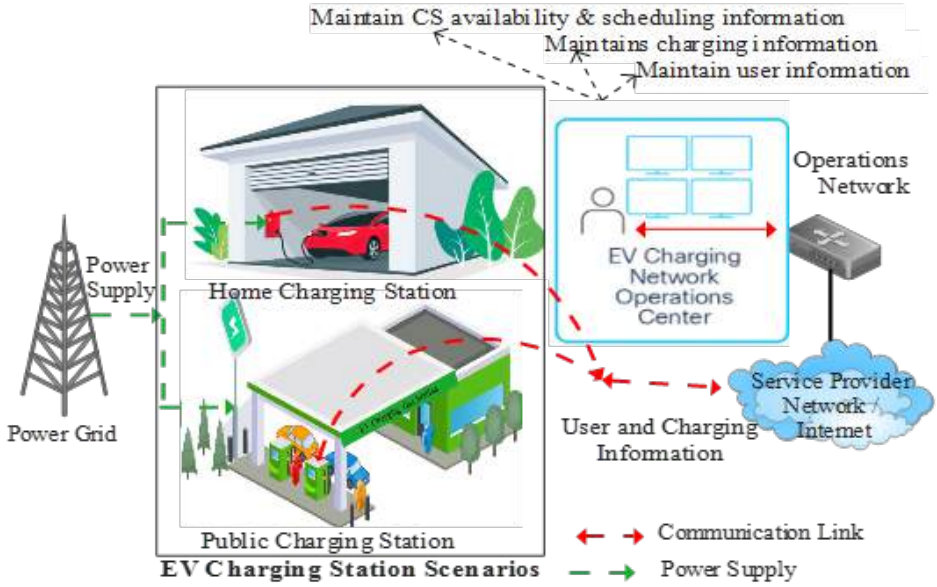


Fig. 2.10 Communication in EVCS Infrastructure

EVCS infrastructure comprises a power grid, charging station, service provider, and EVs user connected. There is communication among them in the network to maintain data

related to charging (Fig.2.8).The service provider relates to the operator network to maintain information on energy and time required for specific EVs. The Service provider also relates to charging stations to check their availability so accordingly they can schedule EVs visits and connect to EVs for user information related to payment. Researchers have identified vulnerabilities in EVCS devices and their communications among networks, including the cloud services involved. EVCSs security evaluations and vulnerabilities are described by inter-face type use case [25].

(Fig.2.9) depicts the four probable entry points used by attackers to compromise EVCSs. Potential security vulnerabilities can arise through various ports of entry in EV charging systems, including EV connectors, user terminals, internet connections, and maintenance terminals. These ports allow attackers to exploit weaknesses and compromise the security of the EV charging infrastructure.

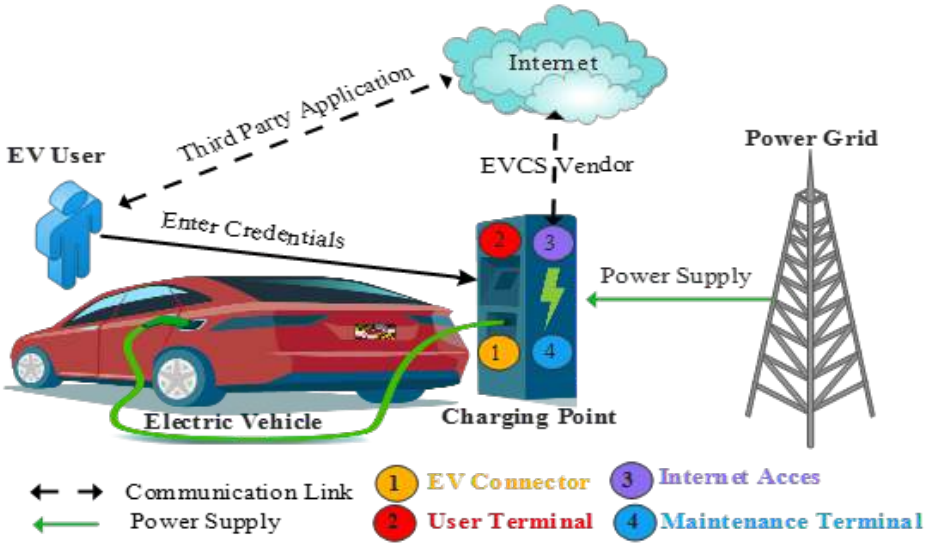


Fig. 2.11 EVCSs with Vulnerable points

1. EV connectors: EV connectors serve as potential targets for attackers due to their communication protocols and connectivity capabilities. Attackers may exploit these vulnerabilities to introduce malware or manipulate charging settings, gaining unauthorized access to the EVCS. This exploitation can have severe security implications, as malicious actors' unauthorized access to the EVCS opens the door for further compromise and control. Moreover, side-channel threats pose a significant concern during the charging process.

   Attackers may leverage these vulnerabilities to gather sensitive information or indirectly manipulate the EVCS. This attempt compromises the privacy and integrity of

the charging system and creates potential risks for the connected vehicles and their owners. Robust security measures should be implemented to ensure the security of EV connectors. These security measures include rigorous testing and validation of communication protocols, implementation of secure coding practices, and continuous monitoring for any signs of malicious activity.

2. User terminals: Public EV charging stations commonly rely on authentication methods such as RFID, NFC, or credit card chips/swipes to connect charging sessions and user accounts, facilitating billing and tracking. However, the security of these authentication systems is crucial, as any compromise could lead to significant consequences for both users and the charging infrastructure.

   If attackers manage to compromise these authentication systems, they gain the ability to carry out various malicious activities. They can deactivate charging sessions, causing inconvenience and potential disruptions for EV owners. Furthermore, attackers can manipulate pricing mechanisms, leading to financial losses for users or the charging station operator. Unauthorized modifications to the equipment could introduce safety risks, impacting not only the charging infrastructure but also the vehicles being charged. Implementing strong encryption and secure communication protocols, regularly updating and patching authentication systems, and conducting thorough vulnerability assessments are essential to mitigate the risks of compromise.

3. Internet connections: Integrating internet connections in modern EVCSs brings convenience and advanced services operators, or EVCS providers offer. Nevertheless, it is essential to acknowledge the associated security risks that arise from this connectivity. Breaching the EVCSs compromises the charging infra-structure and allows attackers to exploit the system as an access point for launching broader attacks on critical infrastructure.

   By infiltrating the EVCSs via an internet connection, attackers can gain unauthorized access to the connected network, extending beyond the charging infrastructure. This entry point could enable them to target and manipulate the critical components of the power grid or transportation network. The consequences of such attacks could be severe, leading to disruptions in power supply and transportation systems or even compromising public safety. Robust security measures should be implemented, including strong access controls, encryption protocols, intrusion detection systems, and regular security updates.

4. Maintenance terminals: EVCSs typically comprise multiple circuit boards communicating through Ethernet, serial, or analogue interfaces [76]. One significant concern is the lack of encryption in module communications, which leaves these communications vulnerable to eavesdropping or tampering by unauthorized individuals. Additionally, the presence of physical ports intended for maintenance purposes can inadvertently create potential access points for attackers if overlooked during production or not properly secured.

   Exploiting these overlooked openings, attackers could gain unauthorized access to the EVCSs, compromising the integrity and security of the entire system [13]. They may monitor sensitive information exchanged between the maintenance terminal and the EVCSs. Moreover, malicious actors could disrupt the operation of the EVCSs, leading to service disruptions, financial losses, or even safety hazards. Implementing robust encryption protocols for module communications, employing secure authentication mechanisms, and conducting regular security audits to address these security concerns is essential. Physical security measures such as securing physical ports and access controls should also be implemented to prevent unauthorized tampering or access to maintenance terminals.

Addressing these security aspects is crucial to ensure the integrity and safety of EV charging systems. Robust security measures, including encryption, authentication mechanisms, and regular security audits, should be implemented to mitigate the risks associated with these ports of entry.

2.4.2 Protocol Centric

Data exchange between the EVs and the EVCSs is outlined in the International Electrotechnical Commission's (IEC) and International Organization for Standardization (ISO) standards. Unfortunately, there are flaws and security holes in this mode of communication. Although initially developed for substation control systems, IEC protocols are now a part of the EVs infrastructure. Protocols such as IEC 61850-90-8 and IEC 61851-1 describe several features of EV charging, but we will only be looking at two for now. The needs of smart charging are met by IEC 61850-90-8, which has considered other standardization initiatives from the start. Some fundamental features for EV charging, such as user identification, were found to be lacking for this protocol [4] and instead assigned to others, such as Open Charge Point Protocol (OCPP) or other IEC protocols. As defined in ISO/IEC 15118, which allows for digital communication in both directions, this international standard supplement the existing IEC 61851-1 [4]. ISO 15118 is neither privacy-friendly nor demand response-compliant,

except for a clause specifying that sensitive data should only be disclosed to those who need to know. Previous research raised privacy concerns, and a real-world attack was attempted [77]. Some flaws in the protocol or incorrect use of existing security mechanisms have been brought to light [78] and are addressed as follows:

1. The Signal-Level Attenuation Characterization protocol supports mutual authentication and encrypted communication, allowing it to function securely.

2. Even though this protocol is Transport Layer Security compliant, encryption is turned off after an external authority verifies the charging session as safe.

3. ISO 15118 also facilitates the establishment of public key infrastructure.

Nevertheless, these safeguards are optional; most manufacturers overlook them to save money and effort. This negligence has left plain-text communications open to assault. Real-time remote control of the EVCSs is made possible by OCPP. This feature facilitates the exchange of data and energy between the EVCSs, the EVs management system, the EVs, and the grid. The most widely adopted charging protocol today is OCPP, known for its standardization. As well as initiating and terminating charging sessions and processing bills, OCPP allows online changes to be made to the charging settings. OCPP supports smart charging by regulating session timing, charging rate, and charging time. OCPP uses many communication protocols to control EV charging but only utilizes HTTP for management. Despite the releases of OCPP 2.0 and 2.0.1 in 2018 and 2020, respectively, OCPP 1.5 and 1.6 are still widely used [13]. Over Web Sockets, OCPP 1.6 supports various communication frameworks, including SOAP/XML and JavaScript Object Notation. Most manufacturers and operators have disregarded OCPP's optional TLS layer for a secure connection to save costs. Extra security was available through an optional hash function in OCPP 1.5. However, OCPP 1.6 removed this option instead of requiring it in further releases. EVs communication protocol road map is presented in (Fig.2.10).

Verifying a charging session's legitimacy with a billing system is the primary focus of OCPP's security measures [8]. As a result, attackers may easily hijack the transmission and take control of EV charging since it is conducted in plain text and encryption is not widely used. Even if an attacker cannot decipher the transmitted data, they may still be able to interrupt an EVs charging process by intercepting and replaying communications, such as those that initiate and end charging sessions. The local authorization list features available in OCPP versions 1.5 and 1.6 ensures that an EVCS can continue to serve its customers during a network outage [79]. The series of IEEE 1547 tackles many of the technical integration difficulties for a mature smart grid, including high penetration of distributed generators, grid

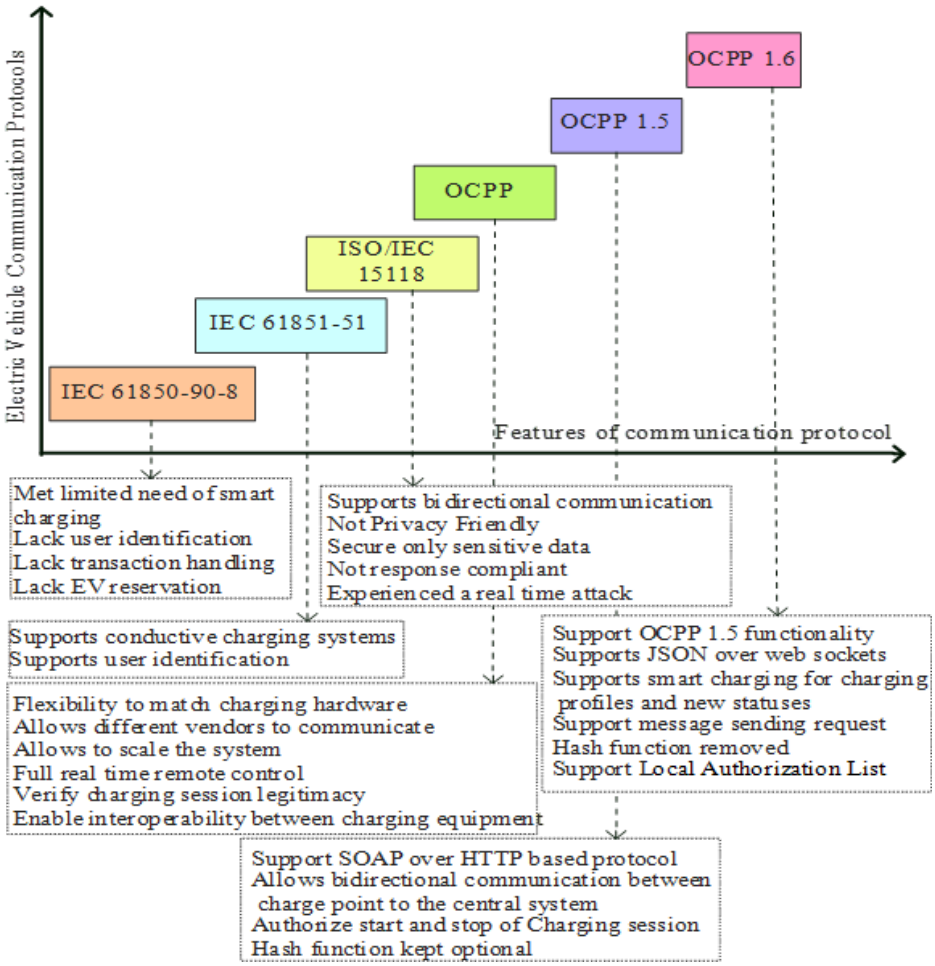


Fig. 2.12 Roadmap of EVs Communication protocol

support, and load control. These are some of the issues that are addressed [80]. The smart grid interoperability standard requires more of a layered security approach. Cybersecurity solutions supplier C2A Security [81] has introduced a new cybersecurity management system called EVSec, which automates EV's infrastructure security. By providing an automated and centralized solution, EVSec can meet the cybersecurity demands of the entire electric vehicle infrastructure.

2.5 Cyber Security Analysis in EVCS

EV charging infrastructure relies heavily on hardware components, software systems and communication networks making it susceptible to various vulnerabilities. Physical tampering with EV charging equipment poses a significant threat to security. Attackers may tamper with charging cables, connectors, or payment terminals to steal electricity, damage infrastructure, or deploy malicious hardware implants [82]. Weak authentication mechanisms can be exploited by attackers to gain unauthorized access to EV charging stations or the charging network. Additionally, inadequate authorization mechanisms may allow attackers to manipulate charging sessions or steal user credentials [83]. The communication protocols used in EV charging stations, such as OCPP and ISO 15118, are susceptible to attacks. For instance, Man-in-the-Middle attacks can intercept communication between the charging station and the backend systems, leading to unauthorized access or data manipulation [84]. Power grid, charging stations, service providers, and EVs users are all linked to Smart Charging Management Systems (SCMSs) and Electric Vehicle Supply Equipment (EVSE). As a result, the power grid might be affected, and PEV batteries can be easily damaged. An EVSE system's accessibility and power consumption might be used to interrupt a building's power supply to a specified region. The disruption would be more severe if the hacker also placed persistent malware in the EVSE, which then propagated to the SCMS and the power grid. This means that SCMS and its interconnected system will fall short of meeting the CIA requirements. The following is a breakdown of the most significant attacks that could be carried out on SCMS and its network, summarized in (Fig.2.11).

1. False Data Injection: False data injection in an EVCS involves an attacker gaining unauthorized access to the communication channels within the system. By exploiting vulnerabilities in the communication protocols, the attacker intercepts data transmission between the EVSEs, smart measuring equipment, and the SCMS. They then manipulate or inject false data related to PEV charging and discharging, such as altering charging rates or battery status [13]. This exploitation can deceive the SCMS, leading to incorrect decision-making and potentially harmful consequences. The impact includes

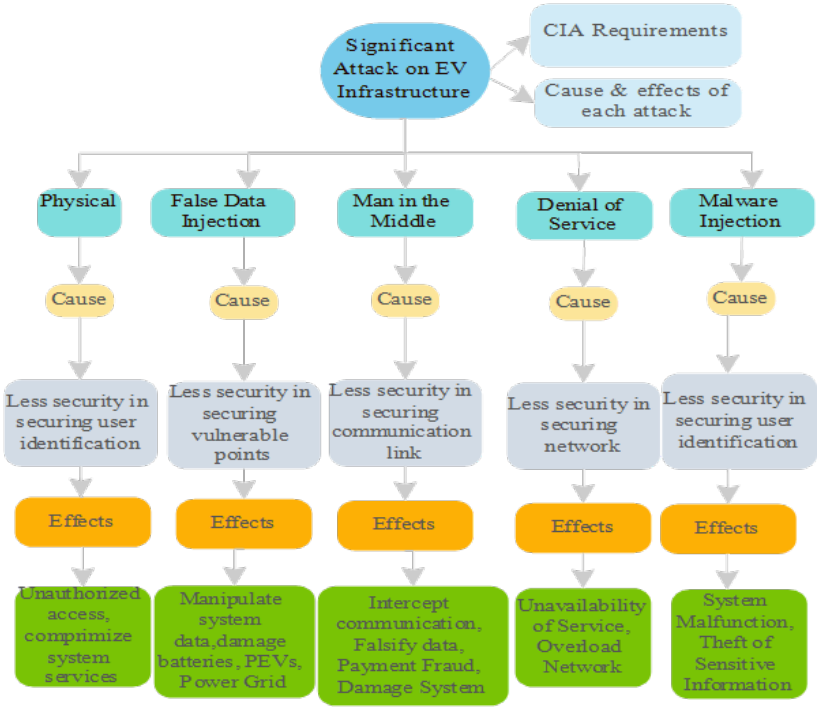


Fig. 2.13 Significant attacks on EVCSs

overcharging batteries, compromised vehicle performance, and disruptions to grid stability. Preventive measures such as secure communication protocols, encryption, and authentication mechanisms are necessary to miti-gate this attack and ensure the integrity of charging data in the EVCS.

2. Man-in-the-Middle: In an EVCS, a Man-in-the-Middle (MITM) attack occurs when an unauthorized attacker inserts themselves between the communication channels of the system. The attacker intercepts and manipulates the data transmission be-tween the EVSE, the PEVs, and the SCMS [79]. By gaining access to the communication flow, the attacker can alter, discard, or misrepresent the data exchanged, leading to various malicious outcomes [4]. For instance, the attacker can tamper with charging requests, leading to overcharging or over-discharging PEV batteries, potentially damaging or reducing their range [4]. Additionally, the attacker can exploit this position to breach privacy by accessing sensitive information exchanged between the PEVs and the SCMS. To mitigate MITM attacks in the EVCS, robust encryption, authentication mechanisms, and secure communication protocols should be implemented to ensure the integrity and confidentiality of the data transmission [85].

3. Denial of Service: In an EVCS, a Denial-of-Service (DoS) attack aims to disrupt the system's normal functioning by overwhelming it with excessive traffic or requests

[74]. In this attack, an adversary targets the SCMS or associated components to overload the network, making it unable to provide services to legitimate users. The attacker may flood the system with high f charging requests, exhaust system resources, or exploit vulnerabilities to crash the SCMS. As a result, the system may become unresponsive, preventing PEVs from accessing the charging services. Such an attack can have severe consequences, particularly for critical emergency vehicles that require charging, potentially compromising their availability and hindering emergency response efforts [86]. To counter DoS attacks in EVCS measures such as traffic filtering, rate limiting, and anomaly detection techniques can be implemented to identify and mitigate abnormal traffic patterns, ensuring uninterrupted and reliable charging services for PEVs [87].

4. Malware Injections: In an EVCS, the Malware Injection attack involves introducing malicious software into the system, mainly targeting the EVSE units. Since EVSEs are often publicly accessible at charging stations, they can become vulnerable to malware infections. Attackers can exploit these vulnerabilities to inject malware into EVSEs, which can then spread to other units within the network. Once infected, the malware can compromise the security of the entire EVCS ecosystem, including the PEVs, the SCMS, and even the power grids. This attack can result in the theft of sensitive data, such as credit card information and personal details, from unsuspecting users [85]. Implementing robust cyber security measures to mitigate Malware Injection attacks, including regular security testing and assessment of EVSEs, is crucial to ensure their integrity and protect the overall EVCS infrastructure from potential malware threats.

5. Physical Attack: A physical attack in an EVCS refers to any deliberate act of damaging or tampering with the system's physical components, such as the EVSE or the PEVs. This type of attack can have severe consequences, including personal safety risks and threats to the integrity of the power grid network [88]. For example, an attacker may physically manipulate the charging infrastructure to disrupt the charging process or cause damage to the electrical system. By compromising the synchronized charging activities, the attacker can create disturbances in the grid's stability and potentially disrupt the overall functionality of the EVCS. Safeguarding against physical attacks in the EVCS requires implementing physical security measures, such as surveillance systems, access controls, and tamper-resistant designs for the charging equipment, to deter and mitigate potential physical threats.

2.6 Cyber Threats to EVCS

The EV charging ecosystem consists of various cyber and physical components, each with distinct vulnerabilities. Previous research has demonstrated the potential exploitation of EV charging applications as attack vectors against the power grid [89]. Thus, the security and integrity of EV charging infrastructure face several significant threats, particularly from cyber attacks. Common attack vectors include malware injections, DoS attacks, and MitM attacks, all of which can disrupt charging services, compromise user data, and even damage charging equipment [4]. A DoS attack can overwhelm a server with illegitimate requests, preventing it from responding to genuine ones and thus disrupting communications within the EV charging infrastructure [90]. Another concerning attack is the MitM attack, which could lead to unauthorized access to sensitive data such as user identities, location information, and charging history collected during EV charging sessions. Insufficient data protection measures can result in privacy breaches, exposing users to risks like identity theft, surveillance, and social engineering attacks [91]. Studies have highlighted vulnerabilities that could be manipulated to affect both EV charging and the broader power grid [4]. For instance, during the Russia-Ukraine conflict, attackers exploited a manufacturer backdoor to display anti-war messages on Russian EV charging stations while executing DoS attacks [92]. In the UK, third-party applications on EV charging stations were compromised to display inappropriate images and conduct further DoS attacks [93]. Furthermore, a security researcher managed to breach an Electrify America EV charging station using Team Viewer, gaining remote control and access [94].These studies collectively enhance our understanding of the cybersecurity landscape surrounding EV charging systems, underscoring the critical need for robust security measures to protect against identified threats such as DoS and MitM attacks.

2.6.1 Overview of DoS Attacks

A DoS attack aims to drain the network resources or hardware, to make resources unavailable to legitimate users. There are different types of DoS attacks, all capable of disrupting online services and making systems unresponsive. DoS attacks can be categorized into three distinct categories [95] : (i) volume based attacks; (ii) resource exhaustion attacks; and (iii) application based attacks as shown in (Fig.2.12).

1. **Volumetric Attacks:**These attacks aim to saturate the bandwidth of the targeted site by flooding it with a massive volume of traffic [95]. Volumetric attacks are notably more impactful in terms of the volume of traffic they generate compared to other DoS attack

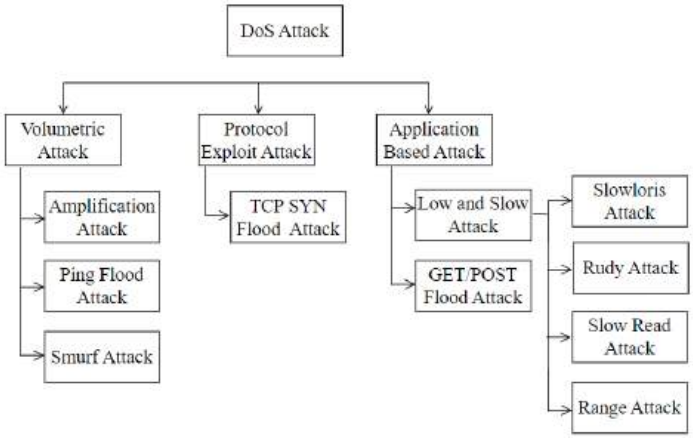


Fig. 2.14 Classification of DoS attack

categories. One prevalent type of volumetric attack involves exploiting an excessive increase in packet size using the UDP or ICMP protocol. One example involving UDP is "Amplification" attack [96].In these attacks, the attacker sends small requests to servers on the internet, typically using UDP packets, with the source address field spoofed to appear as the victim's address. The servers then respond to these requests with much larger responses, which amplifies the amount of data directed towards the victim's address. This can overwhelm the victim's bandwidth and resources, causing a denial of service.

Another example of an attack involving ICMP is the "Ping Flood" attack [97]. In a Ping Flood attack, the attacker sends a large number of ICMP echo request (ping) packets to the target host. The target host must then respond to each ping request, consuming its resources such as network bandwidth and CPU processing power. If the volume of ping requests is sufficiently high, it can overwhelm the target's resources, causing it to become slow or unresponsive to legitimate traffic.The "Smurf Attack" is indeed another example involving ICMP [98]. In a Smurf Attack, the attacker sends a large number of ICMP echo request (ping) packets to a network's broadcast address, spoofing the victim's IP address as the source. The broadcast address causes all devices on the network to respond to the victim's IP address with ICMP echo replies. Since the attacker sends a flood of these requests with the victim's spoofed address, the victim's network becomes inundated with ICMP responses, ultimately leading to a denial of service.

2. **Protocol exploit Attacks:** Attacks in this category aim to drain hardware resources like memory, CPU, and storage, rendering servers unavailable by exploiting protocol vulnerabilities at the network layer [95]. These are also known as protocol-based

attacks, relying on specific message combinations rather than just traffic volume.A common example targets the TCP protocol, called TCP SYN Flood [99], exploit the TCP three-way handshake process by sending a large number of SYN requests without completing the handshake. In this attack, SYN messages are sent to the victim with spoofed source addresses, forcing the target to continuously establish new connections for malicious clients. However, the target server never receives confirmation from the client to complete the connection establishment, leading to depletion of the backlog and rendering the opening of new connections impossible.

3. **Application Based Attacks:** Attacks targeting the application layer aim to exploit weaknesses within an application or service, potentially destabilizing it and preventing legitimate users from accessing it [95]. These attacks are frequently misinterpreted as implementation mistakes because they can mimic the behaviour of legitimate users with minimal malicious traffic. As a result, traditional detection methods often fail to detect such attacks [95]. An example of an application layer attack is the "GET/POST Flood" attack [100]. In this type of attack, the attacker sends a large number of HTTPs GET or POST requests to the target server. By overwhelming the server with these requests, the attacker aims to exhaust its resources, such as processing power, memory, and bandwidth, resulting in service disruption or denial of service for legitimate users. Another common and very famous type of attack on the application layer is the "Low and Slow" attack [100].The "Low and Slow" attack is a method used to exploit vulnerabilities in applications or services by sending traffic at a low rate over an extended period. Unlike traditional attacks that flood the target with a high volume of traffic in a short time, low and slow attacks are designed to fly under the radar of traditional detection mechanisms by mimicking legitimate user behaviour. By gradually and subtly probing for weaknesses, attackers aim to evade detection while causing disruption or gaining unauthorized access to the target system. There are different types of Low-and-Slow Attacks [100], as categorized below:

   (a) **Slowloris Attack:** Slowloris Attack involves keeping multiple HTTPs connections open by sending partial HTTP requests at a slow rate. This tactic exhausts server resources, such as TCP connections and memory, leading to a denial of service for legitimate users.

   (b) **Rudy Attack:** The Rudy attack, also known as a Slow HTTPs POST Attack, involves sending HTTPs POST requests with an exceptionally slow transmission of the request body. This deliberate delay prolongs the server's processing time, tying up resources such as threads and memory. As a result, the targeted service

experiences disruption, potentially leading to denial of service for legitimate users.

(c) **Slow Read Attack:**In a Slow Read Attack, the attacker reads server responses very slowly, extending the duration of each read operation. This action consumes server resources such as network buffers and threads, causing a denial of service for legitimate users.

(d) **Range Attack:** The Range Attack exploits a vulnerability related to the handling of HTTPs Range headers. By sending HTTPs Range requests with overlapping or excessively large ranges, the attacker forces the server to allocate significant memory resources, potentially leading to memory exhaustion and crashes.

These categories offer a structured framework for comprehending the different types of DoS attacks and the tactics employed by attackers to disrupt online services within EVCS platforms.

2.6.2 Impact of DoS on EVCS

The increasing reliance on EVCS has exposed the infrastructure to various cyber threats, with DoS attacks being particularly concerning due to their potential to disrupt services and compromise system integrity. The related work examines the vulnerabilities and consequences of DoS attacks on EVCS.

DoS attacks overwhelm EVCS servers with a flood of malicious traffic, exhausting resources such as CPU, memory, and network bandwidth. The lack of robust rate-limiting mechanisms and request validation in some EVCS implementations exacerbates this vulnerability, making it easier for attackers to disrupt services [89]. Vulnerabilities in communication protocols used within EVCS, such as HTTPs, can be exploited by attackers to initiate DoS attacks. For instance, HTTPs-based EVCS are vulnerable to HTTPs flooding attacks, where attackers flood the system with an excessive number of HTTPs requests, causing service degradation or downtime [101]. Some EVCS lack adequate security measures to detect and mitigate DoS attacks effectively. Without robust intrusion detection systems, and traffic filtering mechanisms, EVCS are more susceptible to prolonged service disruptions and downtime caused by DoS attacks [102].

DoS attacks disrupt the availability and reliability of EV charging services, causing inconvenience to users and potentially damaging the reputation of service providers. Extended periods of service downtime can lead to financial losses and customer dissatisfaction, hindering the adoption and acceptance of EVs [102]. The long-term repercussions of Dos attack

extend beyond immediate service disruptions, affecting trust in the security and reliability of EV charging infrastructure[102]. A DoS attack on EVCS can be especially damaging if it targets the power grid. It could cause major power outages, affecting hospitals and emergency services [103]. The shortage of vital supplies during blackouts could lead to panic buying and unrest. The attack could also create political tensions, raise doubts about the ability of the government and national security policies [103].

2.6.3 Overview of MitM on EVCS

A MitM attack is a significant cyber security threat that occurs when an attacker secretly intercepts and relays communications between two parties who believe they are engaging directly with each other. In EVCS, such attacks can involve intercepting communications between EVs and charging stations, as well as between charging stations and backend servers. Attackers utilize various techniques, including spoofing and packet sniffing, to gain unauthorized access to communication channels. Once in position, they can manipulate the data being transmitted, leading to unauthorized access to sensitive information, such as user credentials, charging session details, and financial data [13] [104]. The inherent vulnerabilities within EVCS, particularly those stemming from the OCPP, create opportunities for attackers, especially if secure communication measures like encryption and authentication are not adequately implemented [105][106]. For instance, weak firmware or inadequately secured wireless communications can serve as entry points for attackers, allowing them to intercept communications during critical phases of the charging process.

The openness of the OCPP presents opportunities for enhancing EV charging solutions but also raises significant security concerns, particularly regarding OCPP 1.6. Research by Conti et al. [105] identified additional vulnerabilities in the communication between EVs and charging stations. Nasr et al. [106] discussed several critical and high-severity vulnerabilities in existing EV charging management systems that could lead to remote cyber-attacks. These vulnerabilities highlight the potential risks associated with the integration of EVCS into smart city infrastructures, as unauthorized access can compromise user privacy and system integrity. Vaidya and Mouftah [107] introduced SecCharge, a management system aimed at supporting the deployment of charging infrastructure in smart cities. Their work detailed numerous security issues present in OCPP 1.6 and outlined several security requirements for improvement. Notably, one significant risk in OCPP 1.6 is the potential for MitM attacks, where an attacker intercepts communications between EVs and charging stations, threatening the confidentiality and integrity of user data. The research conducted by SaiFlow identified weak authentication policies in OCPP, indicating that connections between CPs and CSs could be disrupted by falsifying additional connections to the CS [108]. Friedland's examination of

potential issues when an attacker gains network access to OCPP 1.6 highlights the systemic vulnerabilities that could be exploited by cybercriminals [109]. Alcaraz et al. [13] identified specific threats targeting OCPP CP, including unauthorized access to private messages and data manipulation, which could result in denial-of-service attacks. Furthermore, Rubio et al. [104] specifically emphasized the consequences of successful MitM attacks, suggesting that these vulnerabilities could enable fraudulent charging or disrupt power system operations on a large scale. As a mitigation strategy, they proposed using the OCPP Data Transfer method to exchange secret data between CP and a central system, thereby enhancing communication security.

Building on these insights, Morosan and Pop [110] further explored the security landscape by proposing a neural network approach to classify OCPP traffic (i.e., request/response pairs) into malicious and benign categories. This classification system aims to enhance threat detection within the OCPP ecosystem, addressing the pressing need for improved security measures in the face of evolving cyber threats. The collective findings from these studies underscore the critical importance of implementing robust security protocols to safeguard EVCS against MitM attacks and other cybersecurity threats, ensuring the integrity and reliability of electric vehicle charging infrastructure in the growing landscape of smart cities.

2.6.4 Impact of MitM on EVCS

The (Table 2.3) below summarizes the key impacts of MitM attacks on EVCS.It provides an insight into how these impacts can compromise user privacy, disrupt services, and affect the overall charging ecosystem. Understanding these impacts is crucial for developing robust security measures to mitigate the risks associated with MitM vulnerabilities in EVCS.

The impact of MitM attacks on EVCS is both profound and multifaceted. One of the primary consequences is the compromise of user privacy, as attackers can access sensitive user data, including personal information and charging habits, potentially leading to identity theft and privacy breaches [107]. Additionally, unauthorized access to charging sessions can result in financial losses for users and service providers alike, as attackers may initiate unauthorized charging or alter billing information. The disruption of charging services can lead to DoS conditions, preventing legitimate users from charging their vehicles, which not only frustrates users but also can harm the reputation of charging service providers [108]. Moreover, the consequences of MitM attacks extend beyond individual users to the broader power grid, as disruptions in communication between EVCS and the grid can impact load balancing and demand management, leading to inefficiencies and increased strain during peak usage periods [109]. Regulatory and compliance issues may also arise, as breaches can lead to legal ramifications for service providers failing to implement adequate security

Table 2.3 Impact of MitM Attack on EVCS

Impact	Description
Compromised User Privacy	Access to sensitive user data can lead to identity theft and privacy breaches, exposing users to various risks.
Unauthorized Access and Control	Attackers may initiate unauthorized charging sessions, alter billing information, and disrupt legitimate services, resulting in financial losses.
Disruption of Charging Services	Legitimate users may face Denial-of-Service conditions, leading to service disruptions and reputational damage for providers.
Impact on the Power Grid	Disruptions can affect load balancing and demand management, creating inefficiencies and straining the grid during peak usage periods.
Regulatory and Compliance Issues	Security breaches can lead to legal ramifications and scrutiny for service providers, emphasizing the need for robust security measures.

measures to protect user data. To mitigate these risks, it is crucial to employ robust security measures such as end-to-end encryption, secure authentication protocols, and regular security assessments of EVCS components[110].

2.6.5 Identified Research Gaps and Methodological Baseline

Despite growing attention to cybersecurity in EVCS, several critical research gaps remain unaddressed in the current literature.

- **Absence of Real-World Testing Frameworks:** Most existing studies focus on simulated attack models or theoretical frameworks, lacking practical validation through real-world deployments or emulated test beds [10, 111]. For instance, while Garofalaki et al. [10] propose potential cyberattacks in EV infrastructures, they rely on abstract models rather than emulated OCPP environments. This creates a gap between proposed security mechanisms and their operational effectiveness. This thesis addresses this gap by implementing and testing attacks like BootNotification spoofing and StartTransaction manipulation in a practical test bed using OCPP 1.6 tools.
- **Hybrid Infrastructure Security Challenges:** Limited studies analyse the full security landscape involving EVs, charge points, backends, and the power grid as an integrated system. Most works consider isolated components, lacking a holistic approach to cross-domain threats [24, 20]. For example, Carryl et al. [24] focus on grid vulnerabilities,

while Woo et al. examine vehicle communication. In contrast, this research models a multi-layer threat where a malicious charge point could manipulate backend logic and influence grid-side load forecasts.

- **Unaddressed OCPP Vulnerabilities:** Although the OCPP is widely adopted, specific vulnerabilities such as weak session management, spoofing via boot notifications, and insecure firmware updates are either under explored or treated superficially in literature [19, 25, 84]. For example, Gebauer et al. [19] briefly mention the risk of Charge Point impersonation but do not demonstrate a full exploit. This thesis expands on this by simulating a cloned Charge Point sending a spoofed BootNotification to hijack backend sessions, thereby showcasing real attack feasibility and impact.

To systematically assess and classify threats, several prior works employ established threat modelling frameworks:

- **STRIDE:** The STRIDE framework is commonly used to identify and categorize threats such as spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege [112]. For example, spoofing can occur when a phantom Charge Point pretends to be legitimate during the BootNotification phase, while tampering can involve manipulating values in MeterValues messages. This thesis uses STRIDE to map vulnerabilities at each message exchange phase in the OCPP workflow.
- **DREAD:** The DREAD model allows for semi-quantitative risk analysis by assessing factors including damage potential, reproducibility, exploit ability, affected users, and discoverability [113]. For instance, a token replay attack during the `AuthorizeRequest` phase—where a previously captured valid `IdTag` is reused by an attacker—can score high in terms of damage and exploit ability, as it enables unauthorized charging, but may score low in discoverability. This thesis applies DREAD scoring to prioritize vulnerabilities identified in the OCPP 1.6 test environment.

These tools provide structured benchmarks against which the methodologies proposed in this thesis will be compared in the upcoming chapters.

Chapter 3

Cyber Threat Analysis in EVCS

This chapter presents a cybersecurity threat analysis of the EVCS network. Cybersecurity threat analysis involves evaluating the security risks associated with EV charging infrastructure, including potential cyberattacks targeting charging stations, electric grids, and EV batteries. By analysing key parameters in EV charging, cybersecurity experts can identify vulnerabilities and develop security measures to mitigate these risks. Additionally, threats in EVCS have been validated through real-time, data-centric analysis of EV charging sessions.

Section 3.1 outlines the experimental methodology, detailing the factors considered in detecting abnormal behaviour. **Section 3.2** presents the results and discussion, highlighting instances of abnormal behaviour detected from real-time EV charging data collected from ELadNL. Finally, **Section 3.3** provides a chapter summary, identifying risks based on the experimental findings and proposing mitigation strategies for further research.

3.1 Experimental Methodology

EV network security stands apart from classical network security due to EVs unique characteristics and requirements. The utilization of specialized communication protocols, the integration of physical and cyber components, battery security, the protection of charging infrastructure, and the handling of privacy concerns necessitate tailored security approaches. By acknowledging and addressing these distinctive security aspects, we can build a robust and resilient EV network infrastructure that ensures EV users' safety, privacy, security, and the overall ecosystem. Transactional data shared by ElaadNL [27], which depicts the increase in the use of EVs and their respective rise in usage of Charging Stations for the year 2019, is used to deploy the charts in (Fig. 3.1). The data provided by ElaadNL [27] contains information about various parameters related to charging sessions at different charging stations in the Netherlands. These parameters include total energy consumed, maximum charging power,

connected time, charging time, UTC transaction start and stop timestamps, energy interval, and average power.

The selection of these specific parameters is based on their direct relevance to security-critical aspects of EV charging. For example, unusually long connected times with short charging durations can indicate potential misuse or attacks such as charger hogging or spoofed session initiation [111, 25]. Total energy and power metrics help identify energy theft or unauthorized high-energy transfers [89]. UTC timestamps help track suspicious transaction timing, such as midnight connections or inconsistent duration patterns [18]. These anomalies, if left unchecked, can compromise the integrity and availability of EV infrastructure [24]. To uncover such irregularities, this study employs statistical techniques, particularly regression analysis. Regression analysis helps quantify relationships between parameters (e.g., between charge time and connected time or between energy interval and average power). This allows detection of anomalies that deviate from expected behavioural patterns [19]. Such statistical base lining is essential in cybersecurity contexts, where early detection of data-driven deviations could indicate spoofing, tampering, or denial-of-service activities [20, 10]. Thus, regression and anomaly detection serve not just for operational insights but also for identifying potential cyber threats embedded in transaction data.

Based on these parameters, statistical techniques—particularly regression analysis—can be used to detect abnormal behaviour in EV charging sessions. Regression models quantify the relationship between parameters, allowing the identification of sessions that significantly deviate from expected trends. These deviations may indicate security issues such as energy theft, charger misuse, or protocol-level attacks. Some of the factors considered for detecting abnormal behaviour include:

1. **Total energy consumed:** Sessions consuming unusually high or low energy compared to the typical range may indicate tampering or misreporting.

2. **Maximum charging power:** Abnormally high peak power may suggest unauthorized use of high-capacity charging.

3. **Connected time:** Excessively long connected durations with minimal charging may imply charger hogging or spoofed session initiation.

4. **Charging time:** Very short or unusually long charge durations may point to unexpected interruptions or inefficient charging patterns.

5. **UTC transaction timestamps:** Transactions occurring at unusual times (e.g., late-night or irregular intervals) can signal suspicious scheduling or timing attacks.

6. **Energy interval:** Sudden jumps or drops between consecutive meter readings may suggest tampering or anomalies in energy reporting.

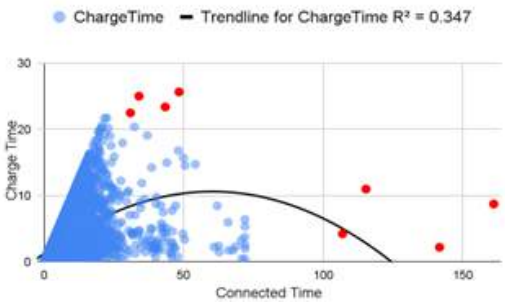
7. **Average power:** Discrepancies where average power diverges significantly from expected norms can indicate manipulation or misconfiguration.

This study employed regression analysis as the core statistical method to assess relationships between these parameters. For instance, regression between Charge-Time and Connected-Time or between Energy-Interval and Average-Power revealed how deviations from established patterns could highlight abnormal behaviours. The analysis was conducted on ElaadNL transaction data, with outliers identified using statistical thresholds (e.g., 95th percentile) and visualized through scatter plots and bar chart. This process allowed the detection of potential anomalies that could pose cybersecurity or operational risks in EV infrastructure.

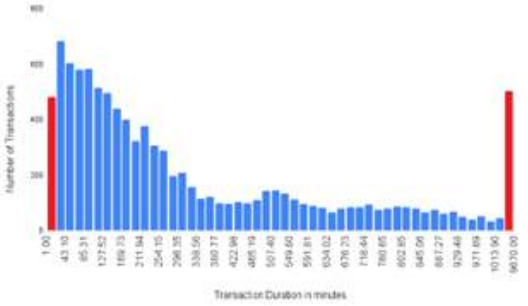
3.2 Result and Discussion

Based on the regression analysis done on the data, the following result shown in (Fig. 3.1 (a)-(d)) explains the abnormal behaviour detected in various charging sessions. Charge-Time is the energy transfer duration, while Connected-Time is the difference between the start and end of a transaction. The graph in (Fig. 3.1 (a)) shows the relationship between Charge-Time and Connected-Time, with an R-squared value of 0.347, indicating that Connected-Time can explain 34.7% of the variation in Charge-Time. Outliers, marked in red, fall outside the 95th percentile range and may warrant further investigation to determine if they are due to a genuine data error or malicious activity. Some records in the dataset exhibit abnormal usage behaviour where the Connected Time is 150+ hours, but the Charge-Time is less than 10 hours. The UTC-Transaction-Start is the start time of a transaction, and the UTC-Transaction-Stop is the stop time of a transaction. The histogram plot (Fig. 3.1 (b)) displays the Transaction Time in minutes, equivalent to Connected-Time. The chart suggests that transactions typically take between 30 and 1035 minutes to commence. Transactions that take over 1200 minutes (20 hours) may be due to connection timeouts or suspicious activity. However, for Level 1 or Level 2 infrastructure, it may take 24+ hours to charge, whereas DC fast charging usually takes 15 minutes to 3 hours.

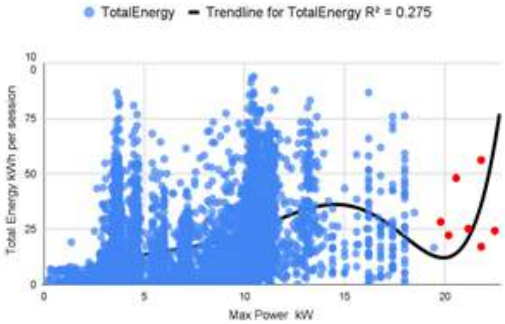
The graph in (Fig. 3.1 (c)) shows the correlation between Max-Power and Total Energy. The L1 charger's output is between 1.3 and 2.4 kW, while the L2 charger is between 3 and 19 kW of AC power. Values beyond these ranges for Max-Power could indicate suspicious activity. In addition, Total-Energy consumption per session beyond 75 kWh may indicate



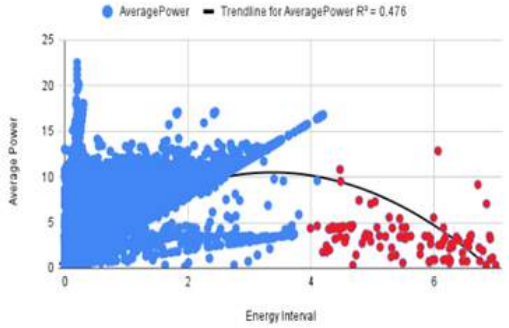
(a) Scatter plot for Connection Time and Charge Time



(b) Bar Chart representing Transaction time-frame



(c) Scatter plot for max power and total energy



(d) Scatter plot for energy interval and average power

Fig. 3.1 Threat analysis of EV charging sessions

heavy battery requirements, which could be a concern. Energy Interval is the total energy (kWh) transfer between two consecutive meter readings, and Average-Power is the average power in kW between two consecutive meter readings. (Fig. 3.1 (d)) shows a graph plot of Energy-Interval and Average-Power, where a change in Energy-Interval explains 48% of the change in Average-Power. Some unusual logs where Average-Power is lower than Energy-Interval beyond 4 kWh is marked red and need further investigation.

The threat analysis suggests that the dataset's charging session data are vulnerable to potential attacks or anomalies. The relationships and patterns between charging session attributes indicate potential issues or suspicious activity. Abnormal usage behavior with extremely long Connected Time but short Charge Time, transactions taking longer than expected, and values beyond normal ranges for Max Power and Total Energy consumption raise concerns. Logs also indicate anomalies where Average Power is lower than Energy Interval beyond a certain threshold. These findings highlight the need for further investigation to determine if these anomalies result from genuine data errors, malicious activities, or potential vulnerabilities in the charging sessions that attackers could exploit.

3.3 Chapter Summary

The rapid adoption of EVs has brought about significant advancements in transportation and energy sectors. However, this growth also introduces various cyber security challenges that must be addressed to ensure the safe and reliable operation of EV charging infrastructure. This section explores these challenges and outlines potential research directions to mitigate cyber threats and improve the overall security posture of EV charging networks.Following are some of the key research areas:

1. Cyber-attacks on EV Charging networks: The issue of cyber-attacks on EV charging networks involves abnormal behaviour in charging sessions that can impact EVs, Charging Stations, and EV Servers [86], as shown in (Fig.3.2). This issue poses real-time problems such as modification, interruption, Interception, and Interference [114]. To address this issue, research directions should focus on Authentication, backup, encryption, security monitoring using firewall, user education, and collaboration for establishing standards. These efforts aim to mitigate cyber threats, enhance security, and ensure the reliable and secure operation of EV charging infrastructure.

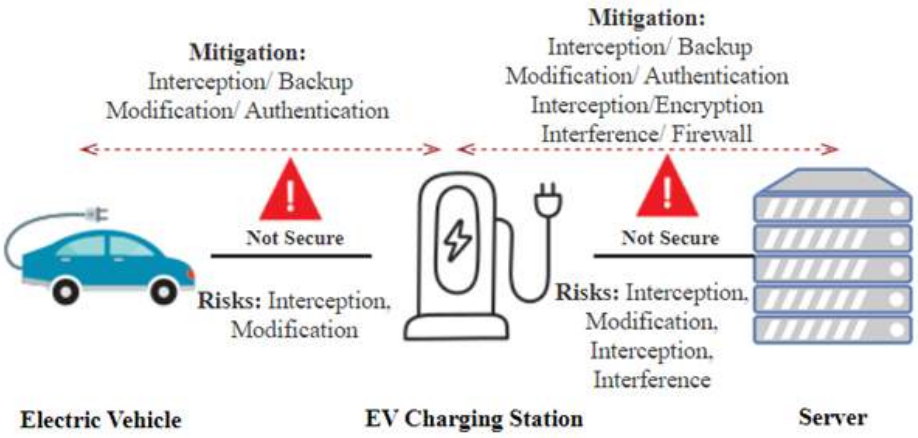


Fig. 3.2 Cyber-attacks on EV charging network

2. Insecure End-to-End Communication for EVs Charging: Communication from end to end relies on a trust paradigm still in its formative phases [78]. In the current state, the majority of the PEV and charging infrastructure sectors need more accessible access to cyber security testing and assessment [8], [13]. This issue creates a real-time problem, raising concerns about the reliability and security of communication channels between EVs, EV charging stations, and servers, as shown in (Fig.3.2). To address this issue, research should focus on developing robust communication protocols, encryption

mechanisms, and authentication frameworks to ensure secure and trustworthy end-to-end communication. Additionally, exploring advancements in energy forecasting and communication technologies can enhance the efficiency and reliability of EV charging systems [77].

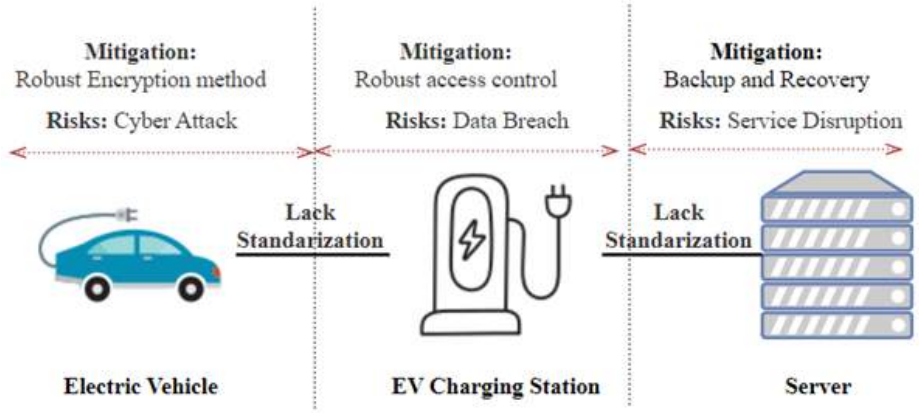


Fig. 3.3 Lack of standardization issue in EV charging network architecture

3. Lack of Standardization in EV Charging Network: The current PEV and EVSE charging infrastructures must meet the best cyber security standards already in place [13],[115]. The existing infrastructure falls short of established cyber security standards and lacks standardized development processes for security software [79], as shown in (Fig.3.3). This issue poses a real-time problem as it leaves the charging infrastructure vulnerable to cyber attacks, potentially leading to unauthorized access, data breaches, and disruptions in charging services. To ad-dress this issue, research should focus on developing and implementing robust security frameworks, improving encryption methods, access control, and providing data backup. Additionally, exploring centralized or distributed cloud services can offer potential solutions for improving the security of the charging infrastructure [116] [117].

4. Define-Test-Validate Charging Security Guidelines: Existing EV infrastructures have not kept pace with the latest technology advancements [77], and accessible EVSEs still struggle with insufficient physical security standards [118][18][119]. As a result, consumer trust in PEVs has been shaken. To address this issue, it is crucial to establish comprehensive and standardized security solutions for existing EVs, as shown in (Fig.3.3). These guidelines should be tested and validated in real-world scenarios to ensure their effectiveness in mitigating cyber security risks and restoring consumer confidence in PEVs. Research efforts should focus on developing robust security

frameworks, physical security standards for EVSEs, and conducting practical tests to validate the effectiveness of recommended security measures [120][121].

5. AI-enabled EV Charging: Considering the capability of AI in detecting or predicting future events, the application of AI for addressing charging related cyber risks is another future direction of EV cyber research themes [122] [123]. Some existing research has shown the potential of using AI in EV research [124]. However, its potential impact on EV cyber research is yet to be explored with specific use cases in securing EV charging networks and detecting risks in EV charging networks.

Chapter 4

EVCMS Framework

This chapter presents the proposed EVCMS framework, detailing its system design, implementation, and performance evaluation. The chapter is structured as follows: **Section 4.1** provides an overview of the EVCMS communication protocol architecture, emphasizing the need for such a system within the EVCS infrastructure. **Section 4.2** introduces the proposed EVCMS framework and its system design, while **Section 4.3** explains the system methodology, focusing on the formulation of charging prices and the optimization of charging plans. **Section 4.4** presents the system implementation, including screenshots of the client- and server-side interfaces, along with an OCPP charge box simulator demonstrating communication within the OCPP framework. **Section 4.5** discusses the security measures integrated into the system design. **Section 4.6** provides a performance evaluation, analysing the system's efficiency by comparing it with conventional charging models. This section also includes a comparative analysis of the proposed model against the current state-of-the-art. Finally, **Section 4.7** summarizes the key findings of the chapter.

4.1 Background

Governments all over the world are supporting EVs to achieve cleaner transportation goals [2]. EVs play a vital role in lowering air pollution, reducing noise, and cutting carbon emissions compared to traditional vehicles running on fuel. For instance, the UK Government plans to stop selling fuel-driven vehicles by 2030, with hybrids following in 2035, aiming to reach net-zero emissions by 2050[1]. This shift towards EVs aligns with global efforts to decrease fossil fuel-driven vehicles and create a more sustainable future [3]. The increase in EV sales has been driven by notable enhancements in the technology and design of EVs [125]. These improvements include the development of more efficient and high-capacity batteries, which have increased the energy density, extended driving range, and significantly reduced charging

times, addressing some of the initial limitations of EVs[126]. Furthermore, advancements in design and features, along with an increased emphasis on sustainability, have made EVs more attractive to environmentally conscious consumers [127]. However, the surge in EV sales has also exposed the need for a more extensive and reliable charging infrastructure [128]. To meet this demand, governments and private entities have invested in expanding the charging network, with the establishment of fast charging stations and the introduction of smart charging management systems[128]. Fast charging stations play a pivotal role by significantly reducing charging times, while smart charging management systems ensure efficient resource utilization and cost minimization for EV users. Leveraging cloud services in EV charging infrastructure enables real-time monitoring and data analysis, elevating user experience and infrastructure efficiency[129]. These enhancements to EV infrastructure have played a vital role in addressing the challenges associated with EV adoption, further promoting the growth of the EV market.

The rapid expansion of EV charging infrastructure has brought forth a new set of challenges, such as user inconvenience, cyber security, optimal scheduling, and poor real-time performance. According to [130], the use of an online reservation system can significantly reduce long wait times, thereby preventing congestion and improving customer satisfaction. With the integration of internet-connected systems, charging stations are increasingly susceptible to cyber threats, such as hacking and data breaches [130]. Safeguarding the personal and financial data of EV users is paramount, requiring robust cyber security measures to protect against potential attacks. In addition to security concerns, ensuring the optimal scheduling of charging sessions is another critical aspect. Smart charging management systems are being developed to address this challenge, allowing for efficient resource allocation and the distribution of charging demand [131]. By analysing charging data, these systems aim to minimize costs for EV users while balancing the grid's load. Furthermore, the use of software systems with an OCPP protocol is considered a suitable solution for large-scale infrastructures that can enhance real-time performance and security requirements [8]. Consequently, addressing the above issues will be pivotal in ensuring the seamless and secure operation of EV charging infrastructure.

Driven by these issues and challenges, this paper introduces the EVCMS communication protocol architecture as shown in (Fig.4.1). The proposed architecture addresses the issues of optimizing charging schedules at an EVCS by allowing EV users to make reservations through a web application using HTTP. Utilizing an optimization algorithm, charging plans are dynamically generated based on user preferences and EV charging requirements. Once the user selects a plan and confirms the reservation, a booking is made. Upon arrival at the charging station, EV drivers can connect to chargers using the ISO 15118 protocol.

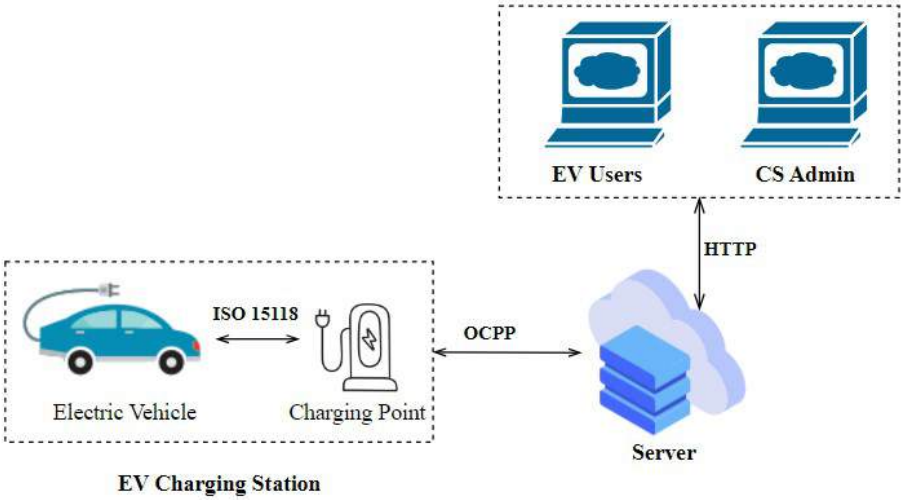


Fig. 4.1 EVCMS communication protocol architecture

Once the user enters the booking ID, the charging connection is remotely controlled via the OCPP protocol to manage each EV charging point. The aim is to enable users to reserve charging outlets in advance and charge at charging stations without human intervention. The optimization method applied in the framework aims to minimize the wait time and the total charging cost for both EV drivers and the EVCS, ensuring cost-effective and efficient charging operations. Thus, the objective of this study is to enhance charging infrastructure by optimizing charging plans, facilitating real-time access, and strengthening the security of smart charging management systems.

4.2 System Design

Within the system design, the discussion encompasses the proposed EVCMS framework detailing its structure and components, followed by the system methodology, outlining the implementation of charging price and charging optimization. Additionally, the focus extends to system security, addressing measures to safeguard against potential threats.

4.2.1 EVCMS Framework

A cloud ready smart charging management architecture for EVCMS as shown in (Fig.4.2), designed to manage the charging of EVs through a cloud-based platform. The system provides a centralized platform to manage the charging of EVs and track the reservation for EV users. It utilizes the OCPP protocol to communicate with charging stations and

manage the charging process. It can be used to manage multiple charging points and provide a seamless charging experience for EV users through an easy-to-use interface.

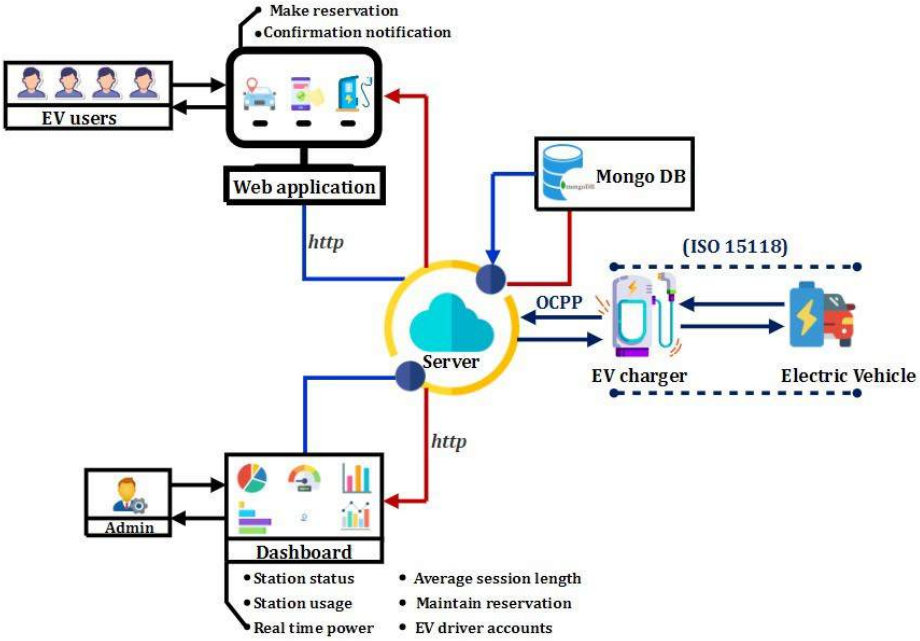


Fig. 4.2 EVCMS Framework

The system aims to enhance the EV charging experience by optimizing charging plans for efficiency, convenience, and cost-effectiveness. This is achieved through dynamic pricing based on real-time energy demand, supply, and pricing data, enabling adjustments to charging prices for improved cost efficiency. Additionally, customer engagement features offer personalized charging experiences with web charging control, customized plans, and money-saving suggestions, all geared toward enhancing user satisfaction. Another key objective is the automation of the EVCMS to efficiently manage all reservations through smart charging operations conducted remotely. This is achieved by implementing the OCPP protocol, enabling automatic and smart operation of the charging station based on scheduling results. This integration of the OCPP protocol enhances the charging system's intelligence and efficiency.

The client-side web application enables users to reserve charging slots for EVs, specifying parameters like booking date, arrival time, duration, and charging power. It optimizes reservations to prevent queuing and provides features to pair with reserved charging units. Post-charging, the application displays essential information such as the time taken, current battery status, power consumed, and session cost. On the server side, a dashboard using the OCPP protocol remotely manages the charging station. It displays booked slots, individual

and overall state of charge, number of vehicles charged, time taken, power consumed, and session cost. The server allows remote start/stop of charging sessions, and notifies users when the EV is connected or fully charged.

4.3 System Methodology

In the system methodology of the proposed EVCMS framework, three crucial components are addressed: charging price, charging optimization for generating optimized plans, and system security. In the EVCMS framework, EV drivers set charging preferences and make reservations. The system dynamically generates charging solutions through optimization, factoring in user preferences, station availability, and Peak OFF/ON times, allowing drivers to confirm or cancel their chosen plan.

4.3.1 Formulating Charging Prices

The cost of a charging session is calculated based on the charging station and the pricing model selected. In general the price per session for a charging station is factor where amount of energy in kilowatt-hours (kWh) delivered during the session is multiplied by the cost per kWh. There are also some instances whereby additional fees may be charged on certain charging stations or even as time-based object prices as well as pricing plans or discounts due to availability that may affect the amount of app credit to be expended per charging session. To derive the formula for the EV charging price for each session of the proposed system, the following parameters have been taken into consideration:

1. Cost of Energy Price: The price per unit of energy (kWh) used during the charging session.

2. Duration Consumed: The total time in hours (h) that the EV was connected to the charging station and consuming energy.

3. Charging rate: It is the rate at which EV battery is charged at the EV charging station.It is measured in kilowatts(kW) and can be considered as the power output of a charging station.

4. Power Output: It is the Maximum Power Output of the Charging Station (in kW) per day.

5. Charging Efficiency: The charging station needs to convert the electrical energy into battery energy for an EV to charge at charging station. The measure with which it is

done is charging efficiency and it is represented as a decimal value between 0 and 1. Here 1 indicates that charging efficiency is 100%, considering all electrical energy supplied to the charging station is utilized to charge the EV batteries.

6. Time-of-Use Charges: This parameter indicates the time when an EV is charged at the charging station. As at some time of day electricity prices are high represented as Peak On time and time of day electricity prices are low represented as Peak Off time. Along with time of day, the availability in charging station is also considered, and the ToU factor is decided as shown in Table 4.1.

7. Peak demand charges: If the EV is charged at charging station when the electricity prices are high then this charge will be taken into consideration. Peak On times are defined in legislation as 8am to 11am and 4pm to 10pm on weekdays.

8. Availability in Station: This parameter indicates whether the charging station is available to use or not at desired time.

9. Service Charge of Station: These are additional charges applied by a charging station for utilizing their service.

Table 4.1 Parameters for ToU

Factor	Availability	Peak Time
0	Available	Peak OFF
0.25	Partially Available	Peak OFF
0.5	Available	Peak ON
1	Partially Available	Peak ON

With these parameters, following formula was derived for the EV charging price for each session:

$$\text{EV Charging Price} = (Ec \cdot Es) + \text{ToU} + \text{SC}_{CS} \tag{4.1}$$

where:
Energy Consumed (in kWh) = Duration Consumed (in hours) x Charging Rate (in kW)

$$i.e. Es = Dh \cdot Crate \tag{4.2}$$

Charging Rate (in kW) = Maximum Power Output of the Charging Station (in kW) x Charging Efficiency (as a decimal)

$$i.e. Crate = Pmax \cdot Ceff \tag{4.3}$$

Time-of-Use Charges= Availability x Peak Demand Charges

$$i.e. ToU = \mathrm{A}_{CS} \cdot PDC \tag{4.4}$$

Substituting (4.2), (4.3) and (4.4) in (4.1) the following formula can be derived:

$$EVChargingPrice = (Ec \cdot Es) + ToU + \mathrm{SC}_{CS} \tag{4.5}$$

$$= (Ec \cdot Dh \cdot Crate) + ToU + \mathrm{SC}_{CS} \tag{4.6}$$

$$= (Ec \cdot Dh \cdot Pmax \cdot Ceff) + ToU + \mathrm{SC}_{CS} \tag{4.7}$$

$$= (Ec \cdot Dh \cdot Pmax \cdot Ceff) + (ACS \cdot PDC) + \mathrm{SC}_{CS} \tag{4.8}$$

4.3.2 Charging Price Optimization

Generating optimized plans for EV users based on various parameters can be a complex task, but by leveraging cloud services and data analytics tools, it can be achieved efficiently and effectively [132]. To generate optimized plans for EV users, parameters such as power requirement, and time requirement of user is taken into consideration along with charging station availability and peak off/on time of charging station.

The (Algorithm 4.1) Check Availability and Assign Time Slot begins by initializing variables such as the start time (startTime), end time (endTime), and a flag (isTimeSlotOverlapping) to track overlap status. It then iterates through existing bookings, comparing their time slots with the user-requested time. If there is an overlap with existing bookings then the algorithm sets the flag to true and searches for an alternative time slot by increasing the start time of requested booking by 15 minutes. This is continued in loop unless a non overlapping slot is found. Once a suitable time is identified, the algorithm finalizes the new date-time (finalDateTime). If there is no initial overlap, the algorithm directly finalizes the date-time to the user's requested time. The algorithm returns a value (isAvailable) indicating whether the requested time slot is available and the finalized date and time for the booking (finalDateTime). Thus, the algorithm efficiently manages booking conflicts, ensuring users are assigned available time slots or proposing suitable alternatives when needed.

Algorithm 4.1 Check Availability and Assign Time Slot

Input : dateTime (givenDate & givenTime), duration, allBookings

Output : isAvailable, finalDateTime

Initialize:
startTime ← givenTime
endTime ← givenTime + duration
isTimeSlotOverlapping ← false **Start:**
for *booking in allBookings* **do**
    bookingStart ← booking.time
    bookingEnd ← booking.time + booking.duration
    if *current booking start and end overlap with user requested time* **then**
        isTimeSlotOverlapping ← true
        break
if *isTimeSlotOverlapping* **then**
    newStartTime ← givenTime + duration
    while *true* **do**
        newStartTime ← previous start time + 15 mins buffer newEndTime ← newStartTime + duration
        isTimeSlotOverlapping ← false
        for *booking in allBookings* **do**
            bookingStart ← booking.time
            bookingEnd ← booking.time + booking.duration
            if *user requested date-time overlaps with new date-time* **then**
                isTimeSlotOverlapping ← true
                break
        if *isTimeSlotOverlapping* **then**
            update the time to the next 15 mins from the original time and continue checking for overlap
        else
            finalize the new date-time
            finalDateTime ← newStartTime
            break
else
    finalize the given date-time
    finalDateTime ← givenDateTime
Return: isAvailable, finalDateTime

The (Algorithm 4.2) Generate Plan algorithm is designed to create effective charging plans for an EV. It begins by determining if the requested charging time falls within a peak period. The algorithm first determines the peak status of the final date time. Based on availability and peak status, the algorithm then calculates cost multipliers. If the isPeakOn is true and there is a timeslot overlap, it assigns a cost multiplier of 1, indicating that the charging is not available in peak on period. Otherwise, if there is no time slot overlap, it sets the cost multiplier to 0.5, signifying availability in the charging station but in peak on period. On the other hand, if isPeakOn is false, the algorithm assigns a cost multiplier of 0.25 in case of a time slot overlap, indicating that charging is not available in peak off period. If there is no time slot overlap during peak off period, the cost multiplier is set to 0, implying that charging is available at no additional cost. The algorithm then returns the computed Availability Cost Multiplier (ACM). Including time-of-use charges, the overall charging price is then computed considering energy cost, duration, charging rate, and additional service charge.

Based on the availability of charging stations and peak off/on time data analysed by the server, further plans will be generated. These plans might involve determining the most efficient time to charge based on the user's power and time requirements and the peak off/on times of the charging station. Based on the parameters taken into consideration following optimized plans can be generated to suit user requirements in consideration with charging station requirements:

1. Plan 1 (Duration): User has entered arrival time, duration, and power required. However the power required is not sufficient to charge the EV as per the time requested. So, provide another plan with updated charging power to meet the time constraints.

2. Plan 2 (Power): User has entered arrival time, duration, and power required. However the required time is not sufficient to charge the EV as per the charging power requested. So, update the time in one plan to meet charging needs.

3. Plan 3 (Duration Eco): Enhancement made to plan 1 by shifting the time slot to peak off period. In this plan, the system suggests the nearest possible peak off time slot w.r.t to the required time duration. This plan minimizes the charging cost compared to the plan 1 charging cost.

4. Plan 4 (Power Eco): Enhancement made to plan 3 by shifting the time slot to peak off period. In this plan, the system suggests the nearest possible peak off time slot w.r.t to power required. This plan minimizes the charging cost compared to the plan 3 charging cost.

Algorithm 4.2 Generate Plan

Input : finalDateTime, isAvailable,CostOfEnergy (Ec), Chargingrate(Crate), peakDemand-Charge(PDC), ServiceCharge(SCcs)

Output : List of Plans

Start:
if *((dT ≥ 8am AND dT ≤ 11am) OR (dT ≥ 4pm AND dT ≤ 10pm))* **then**
    isPeakOn ← true;
isPeakOn ← false; **if** *isPeakOn ==true* **then**
    if *isAvailable* **then**
        availabilityCostMultiplier ← 0.5
    else
        availabilityCostMultiplier ← 1
else
    if *isAvailable* **then**
        availabilityCostMultiplier ← 0
    else
        availabilityCostMultiplier ← 0.25
TimeOfUseCharge (ToU) = availabilityCostMultiplier × PDC
ChargingPrice = (Ec × duration × Crate) + ToU + SCcs
if *isPeakOn ==false* **then**
    Power← duration×chargingRate
    Plan 1: Based on "duration" requirement adjust power
    Plan 2: Based on "power" requirement adjust duration
else
    Power← duration×chargingRate
    Plan 1: Based on "duration" requirement adjust power
    Plan 2: Based on "power" requirement adjust duration
    dateTime ← get a time-slot available in the peakOff period
    Plan 3: Based on "duration" requirement adjust power in peakoff
    Plan 4: Based on "power" requirement adjust duration in peakoff
Return: List of Plans

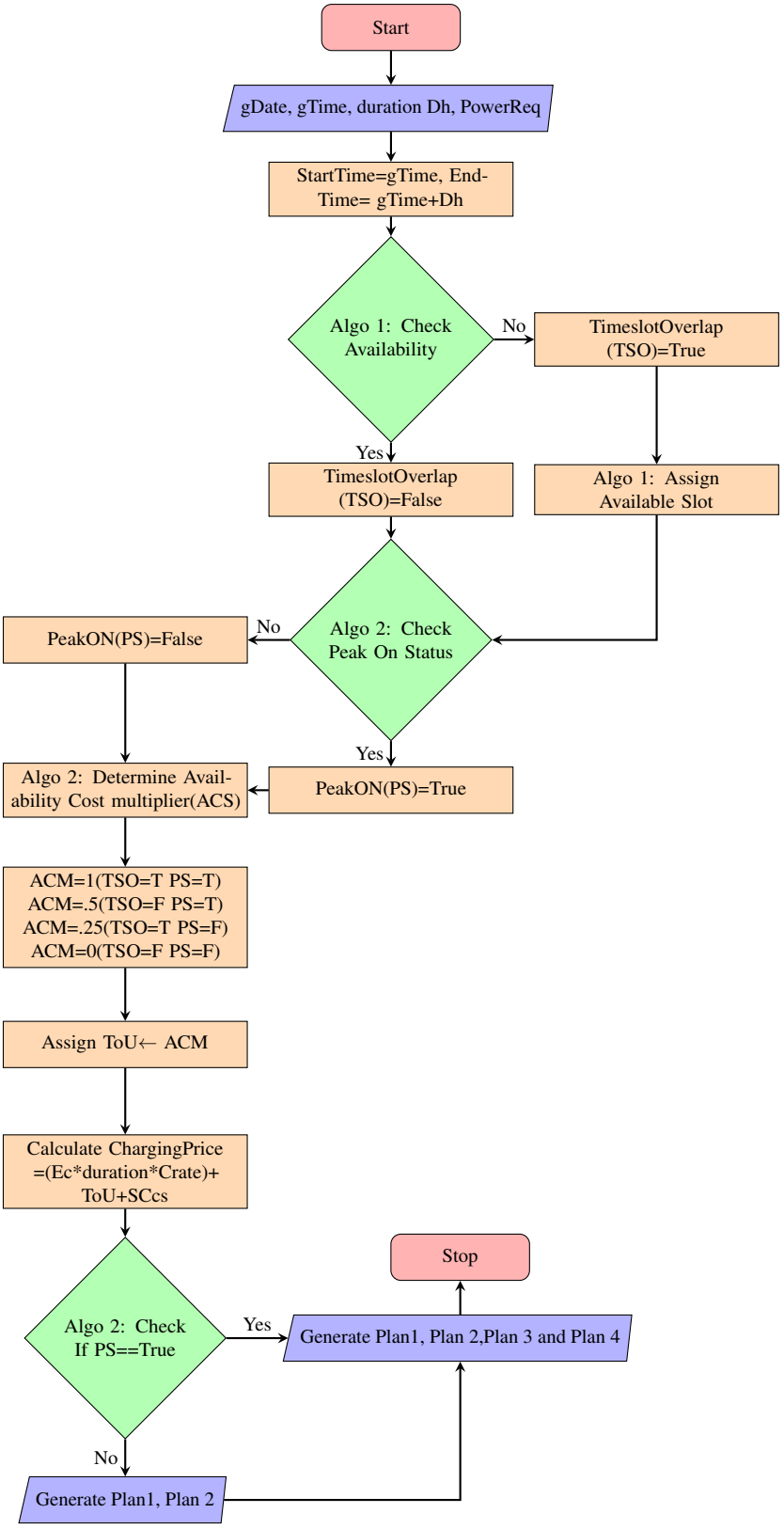


Fig. 4.3 Charging plan optimization flowchart

The first two plans are generated in both statuses (Peak on and Peak Off) by adjusting either power or duration to meet specific user requirements. If the charging time is during the Peak On period, the algorithm generates the first 2 plans in the Peak on period and also seeks an alternative slot in the Peak Off period, and creates two more plans with adjusted power or duration in the Peak off period. These plans are considered as Eco plans. Thus the algorithm outputs a list of plans tailored to user preferences, availability constraints, and cost optimization.

(Fig.4.3) illustrates the flowchart of the EVCMS optimization algorithm. This flowchart serves as a road map for EVCMS framework, enabling it to optimize its operations and enhance service efficiency. The flowchart visually guides the process of resource management and overall optimization process.

The complexity of optimization algorithm can be described using two components namely space and time complexity. Time complexity indicates how the algorithm's execution time grows relative to the input size. Lower time complexity signifies better efficiency. Similarly, space complexity reflects how much memory the algorithm requires as the input grows. Lower space complexity indicates more efficient memory usage.

Considering that EVs for charging will be limited per day, time complexity analysis is the major component in the complexity analysis of the proposed optimization technique. Let, Tg is the time gap between each EV booking time, and Nb is the number of bookings per day. To determine the available slot, we linearly scan through all bookings (Nb) and find the available slot. The time complexity of the step is O(Nb). In case there is no available slot, then the algorithm seek for the next available slot in increment of time gap (Tg). If the time gap (Tg) checked for finding the next available slot is constant (15min). For any day the max gap available would be calculated by total time per day in minutes/time gap= 24*60/15 = 96, which can be considered constant. Thus, the worst-case complexity of this step would be O(96*Nb)= O(Nb). Using these notations, the time complexity of algorithm is linear and can be expressed as O(Nb).Plan generation is a constant time operation based on the peak status and the availability it can generate maximum 4 plans. So, the space complexity is also constant and can be expressed as O(1). As the proposed algorithm has lower time complexity as O(Nb) and space complexity as O(1), the algorithm is highly efficient and can process larger inputs quickly while using minimal memory.

4.4 System Implementation

The cloud ready EVCMS framework depicted in (Fig.4.2), comprises two key modules: the client-server and the OCPP Chargebox simulator. These critical components work together

to ensure the system's functionality. Further details on each module will be explored in subsequent subsections.

4.4.1 Client Server

React was chosen as the framework for developing the front end of EVCMS client-server module due to its capability to create intricate user interfaces using reusable components. A variety of additional tools and packages were leveraged alongside React to expedite development and introduce extended functionalities [133]. The unique way React manages updates ensured a swift and seamless user experience, a critical aspect given the frequent UI changes required [134]. Node.js was selected as the back-end technology owing to its scalability, robust security features, and strong performance [135]. Notably, Node.js includes built-in support for secure communication between the server and clients through TLS/SSL encryption [136]. Its adeptness at managing multiple simultaneous connections made it apt for real-time data processing and rapid data retrieval, both essential for the project's needs [135].MongoDB was the preferred database solution due to its adaptability and scalability. Its support for data sharing and replication contributed to high data availability and reliability [137]. MongoDB's dynamic schema allowed for efficient storage and retrieval of complex data structures, aligning well with the evolving data requirements of the project.

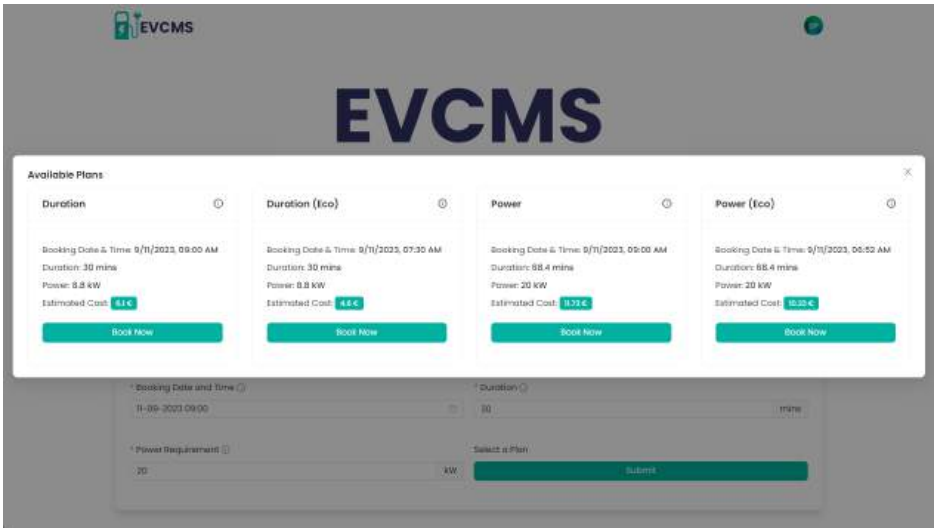


Fig. 4.4 Client Side User Interface for cloud ready EVCMS framework

The integration of various technologies played a pivotal role in achieving project objectives, ensuring a secure and user-friendly charging experience for EV owners. The web application developed for cloud ready client-side EVCMS architecture shown in (Fig.4.4),

proved to be highly effective in enhancing the user experience and streamlining the charging process. The implemented features aimed to enhance user experience and streamline the EV charging process. One significant aspect was the introduction of Reservation Flexibility, allowing users to customize charging parameters and plan sessions in advance. This eliminated the need for queuing, ensuring a seamless charging experience. Additionally, the application offered Slot Reservations, enabling users to secure a charging slot at their preferred time. Pairing this with the reserved charging unit further improved overall convenience. Post-charging, users could access detailed Charging Status and Information, offering insights into the state of charge, charging duration, battery status, energy consumption, and session costs. This transparency empowered users to effectively manage their charging expenses.

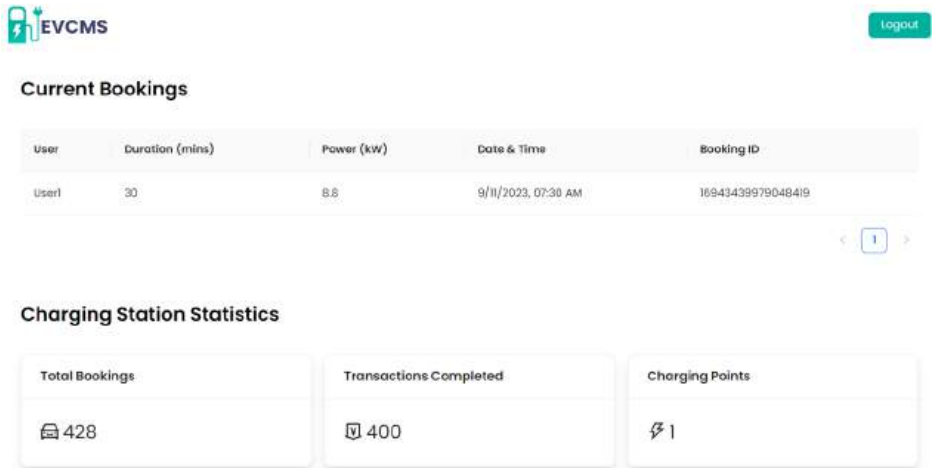


Fig. 4.5 Server Side User Interface for cloud ready EVCMS framework

The cloud ready server-side dashboard for EVCMS framework shown in (Fig.4.5), serves as a central control hub, empowering operators with essential tools to monitor and manage charging stations efficiently. The server-side UI introduces key features that significantly enhance the management and user experience of EV charging stations. A central component is the provision of real-time access to Booking and Charging Data through a comprehensive dashboard. This feature facilitates efficient monitoring of booked slots and vehicle charging status, empowering operators to make informed decisions for optimizing station utilization. Another noteworthy feature is Remote Charging Session Control, which utilizes the OCPP protocol. This capability allows operators to remotely initiate and conclude charging sessions once vehicles are paired with the charging unit. By minimizing the need for manual intervention, this feature not only streamlines the charging process but also boosts operational efficiency. Both charging station operators and EV users benefit, as operators can manage multiple stations remotely, while users enjoy a convenient and seamless charging experience. The implementation also incorporates User Notifications, adding a layer of proactive commu-

nication for EV owners. Automated notifications inform users when their EV is successfully connected to the charging unit and when the charging session is completed. This real-time feedback ensures that users stay informed about their charging progress, contributing to a user-friendly and satisfying overall experience.

4.4.2 OCPP Chargebox Simulator

The framework utilized OCPP to communicate with charging stations, seamlessly managing reservation, charging, and billing operations. The UI for the Chargebox simulator is designed to be user-friendly and intuitive. It provides a clear and straightforward interface that allows users, including EV drivers and operators, to easily interact with the simulator.

Fig. 4.6 Chargebox Simulator User Interface for cloud ready EVCMS framework

(Fig.4.6) showcases the user interface of a Chargebox simulator used within EVCMS framework.The UI is a user-friendly gateway for both EV drivers and operators, providing intuitive controls to configure connections to the EVCMS framework. It includes authentication features for secure access, real-time connection status indicators, and a log panel for messages and error notifications. Action buttons for Connect and Disconnect simplify the connection process, with feedback mechanisms ensuring users are well-informed about the connection status. This UI's role is pivotal, enabling users to seamlessly interact with the charging management system and manage charging sessions effectively, all within an accessible and user-centric environment.

The connection between the EVCMS framework and the Chargebox simulator is made smooth and controlled with the help of OCPP. (Fig.4.7) highlights the central role of OCPP

in the integration of EV charging services within a cloud ready EVCMS and the Chargebox simulator's various functions. This pairing lets users interact easily with the simulator, creating a link between the virtual environment and the real charging setup. To initiate this process, the OCPP client within the Chargebox simulator first establishes a secure connection and obtains authorization from the OCPP server within the EVCMS. This foundational step sets the stage for subsequent interactions between charge point and EVCMS. Once authorization is secured, users can input their booking ID within the simulator's interface, which then leverages OCPP to validate the booking ID by forwarding it to the primary EVCMS server.

Following successful validation, OCPP is employed once more, this time to transmit specific authorization instructions back to the Chargebox simulator. This essential authorization process ensures that the EV is granted access to the charging infrastructure. With authorization in place, users gain the capability to interact seamlessly with the Chargebox simulator via its user interface. OCPP functions for Chargebox simulator such as Connect, Authorize, Start Charging, Stop Charging, Meter Values Request, and Disconnect in an EV charging system involves integrating the protocol into both the client and server EVCMS.

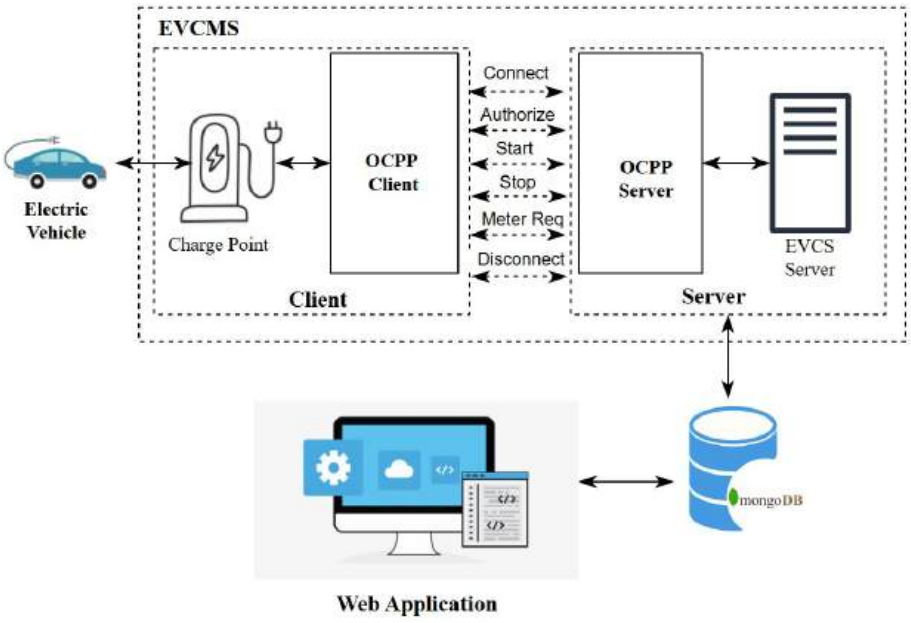


Fig. 4.7 Integration of OCPP in Cloud ready EVCMS framework with Chargebox Simulator Functions

Following is the high-level representation of each function request and response with the charge box simulator:

1. **Connect:** In the Chargebox implementation, the Connect function utilizes the OCPP BootNotification message, facilitating communication between the Chargebox and EVCMS as illustrated in (Fig.4.8). When the EV connects or the Chargebox boots up, it sends a BootNotification request to EVCMS, carrying charge point information. The Chargebox then handles the BootNotification response, containing configuration data of the OCPP server. For EVCMS, the system actively monitors incoming BootNotification requests from various Chargeboxes. Upon receipt, it rigorously validates the charging stations identity. Subsequently, EVCMS promptly sends a BootNotification response, indicating the approval or denial of the connection request. If accepted, the response may include relevant configuration parameters for subsequent setup procedures. This ensures a safe handshake between client and server before actual charging session starts.

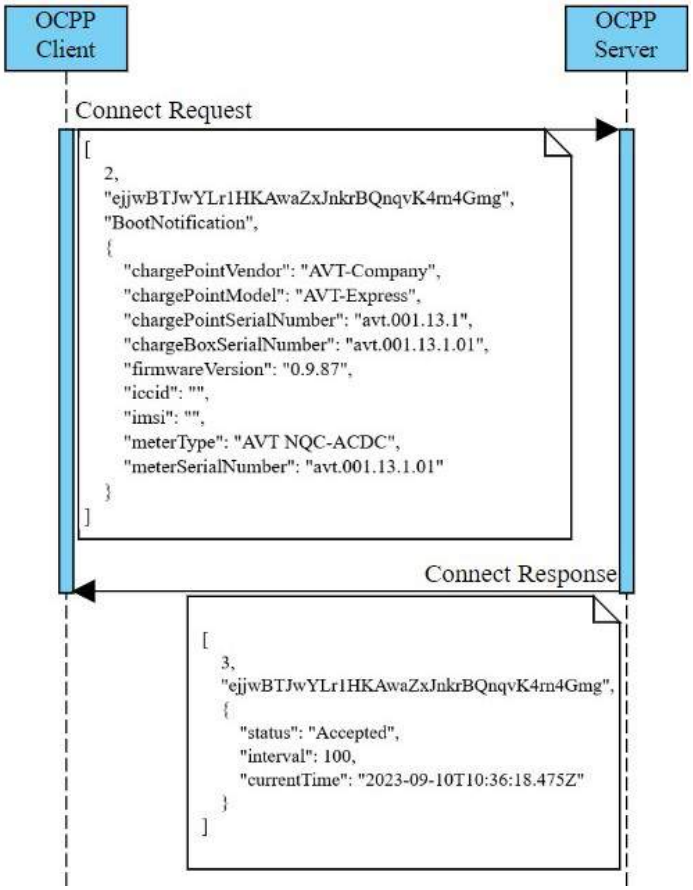


Fig. 4.8 Request and Response for Connect function

2. **Authorize:** In the Chargebox Authorization function, illustrated in (Fig.4.9), the process of handling authorization requests is outlined. When an EV driver seeks

permission to use a specific charging slot, the Chargebox initiates an Authorize request to the EVCMS. This request, including the slot preference, prompts the EVCMS to respond, indicating whether the driver can proceed with the charging session. This authentication step ensures user verification before access to the charging station is granted or denied. In the EVCMS implementation, the system monitors incoming Authorize requests, authenticates the driver, and responds to the Chargebox accordingly.

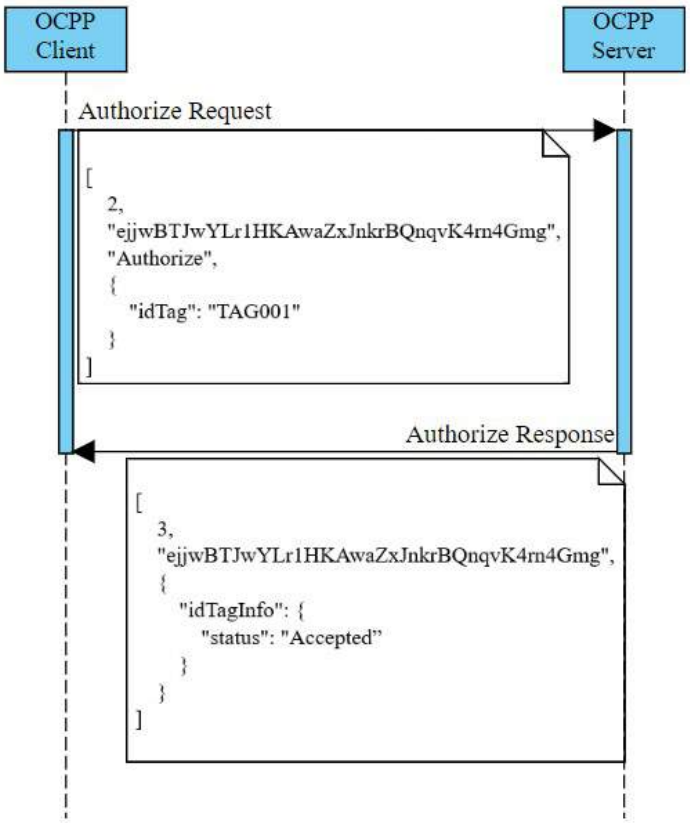


Fig. 4.9 Request and Response for Authorize function

3. **Start Charging:** In the Start Charging function of the Chargebox illustrated in (Fig.4.10), a Start Transaction request is sent to the EVCMS when a charging session begins. This request includes essential transaction details such as connector Id, Slot Id, timestamp, initial meter value set to 0, and reservation ID. The Chargebox then carefully handles the Start Transaction response, ensuring the correctness of the details before responding with an accept message.In the EVCMS implementation, the system listens for incoming Start Transaction requests from the Chargebox, assigning a unique transaction ID upon reception and processing relevant session data. The EVCMS promptly issues an accepted Start Transaction response, by verifying transaction de-

tails to ensure it matches with the reserved charging session. This verification process contributes to a smooth and reliable start for the charging session, enhancing the EV user experience.

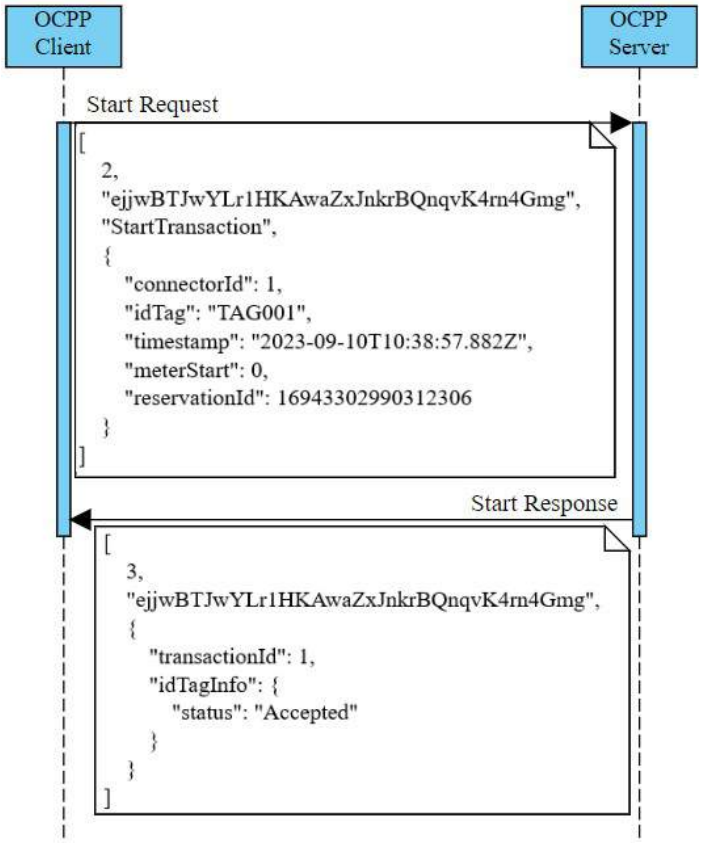


Fig. 4.10 Request and Response for Start Charging function

4. **Stop Charging:**In the Stop Charging function of the Chargebox illustrated in (Fig.4.11), a Stop Transaction request is sent to the EVCMS when a charging session concludes. This request includes transaction ID, Slot ID, timestamp for charging duration, and meter values. The Chargebox then handles the Stop Transaction response from the EVCMS, confirming the end of the session and potentially providing billing information. In the EVCMS context, the system actively listens for incoming Stop Transaction requests from the Chargebox. Upon reception, it efficiently processes the requests, evaluating relevant data. One crucial task is calculating the total energy consumed during the charging session, enabling the system to determine associated costs. The EVCMS then formulates a comprehensive Stop Transaction response, serving as an acknowledgment of the successful completion of the charging session. Stop charging function can be initiated manually by the user or automatically can be initiated by EVCMS

as per reserved charging duration. This process ensures safe end of communication between client and server.

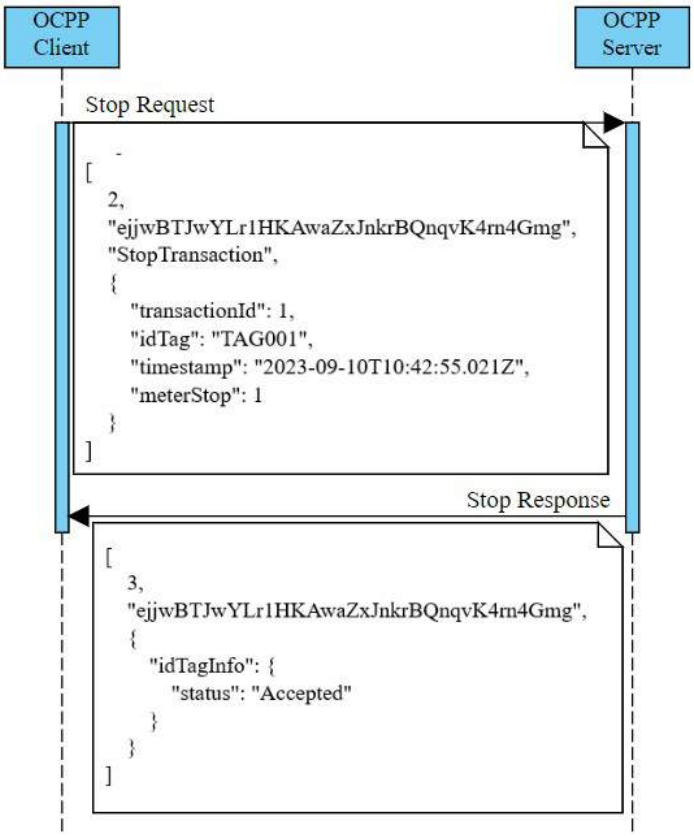


Fig. 4.11 Request and Response for Stop Charging function

5. **Meter Values Request:**In the Meter Values Request function of the Chargebox illustrated in (Fig.4.12), periodic or on-demand meter values requests are sent to the EVCMS. These requests provide real-time data about the ongoing charging session, including connector ID, transaction ID, current meter value, and other session details, enabling comprehensive monitoring. In the EVCMS context, the system actively anticipates and receives incoming meter values requests from the Chargebox simulator. Upon reception, the EVCMS initializes the meter value with the current timestamp and sampled value as a reference. It processes the meter data, storing it if necessary for billing or monitoring. Following data processing, the EVCMS responds with an acknowledgment to the Chargebox, confirming the receipt of meter values. This data exchange ensures accurate relay of essential information, contributing to efficient management and oversight of charging sessions, ensuring well-organized and closely monitored EV charging.

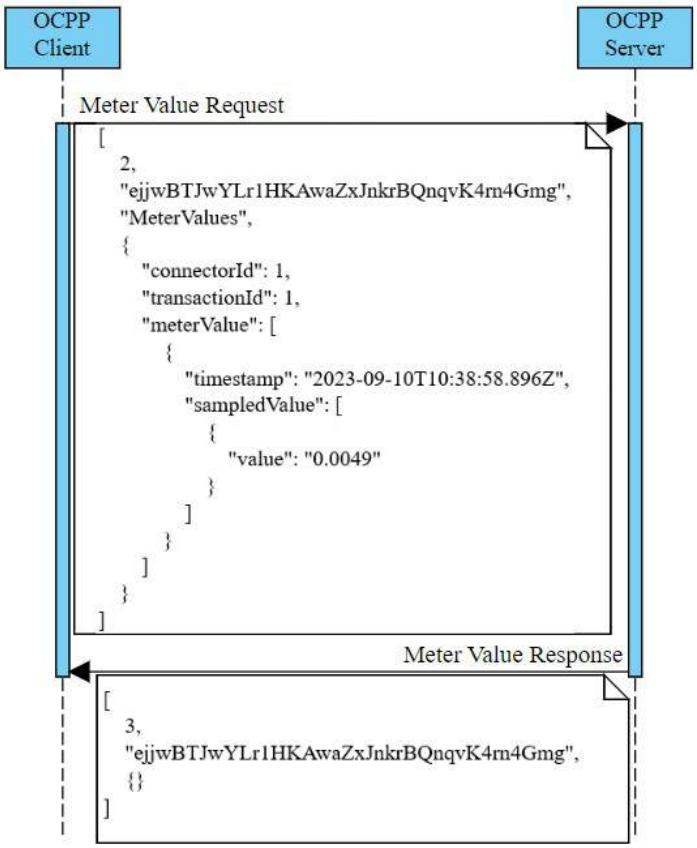


Fig. 4.12 Request and Response for Meter Values function

6. **Disconnect:** In the Disconnect function, both the Chargebox and EVCMS implementations establish procedures for effective disconnection management, catering to various scenarios, whether concluding a charging session smoothly or addressing unexpected errors. The Chargebox ensures a graceful disconnect, smoothly wrapping up sessions, considering factors like session completion or unforeseen issues.On the other hand, the EVCMS proactively monitors the Chargebox connection status, handling disconnections seamlessly for a smooth transition from an active to a terminated charging session. It also efficiently manages the release of allocated resources, crucial for overall system efficiency. Notably, the disconnection process doesn't involve specific user or system requests but centres around the careful closure of the Web Socket connection initially established with the OCPP server.

   All these functions incorporated with OCPP communication improves the security and makes EVCMS robust against cyber attacks.

4.5 System Security

The implementation process of integrating the OCPP-RPC (Open Charge Point Protocol-Remote Procedure Call) library [138] and the OCPP Chargebox Simulator [139] involved several steps to achieve seamless communication and control within the EV charging ecosystem. Initially, the OCPP-RPC library was employed as a foundational component due to its capabilities in managing remote procedure calls in the OCPP. The process commenced by obtaining the library from its GitHub repository and incorporating it into the project. This library served as the bridge for initiating communication between the EVCMS framework and the Chargebox simulator. Simultaneously, the OCPP Chargebox Simulator was sourced from its repository [139]. This simulator, designed for OCPP version 1.6, was selected for its compatibility with the chosen OCPP-RPC library. The simulator's setup involved configuring essential parameters, including the supported OCPP version, charging point details, and communication settings, ensuring alignment with the chosen library. Next, the integration efforts focused on coordinating the functionalities of the OCPP-RPC library and the OCPP Chargebox Simulator. This involved modifying the EVCMS backend to accommodate interactions using the OCPP-RPC library. Communication endpoints were established within the EVCMS framework to facilitate the exchange of OCPP messages with the simulator using an encrypted channel. Instances of the OCPP Chargebox Simulator were then linked to these communication endpoints, enabling a streamlined flow of OCPP messages between the EVCMS and the simulator. The communication protocol was precisely defined, outlining the specific messages required for actions like connecting, authorizing, initiating, and stopping charging sessions.By doing so communications channel can be encrypted,such that charger and EVCMS server have secure connection to avoid any unauthorized access.

4.6 Performance Evaluation

In the performance evaluation section, the system is assessed using specific metrics for result analysis, emphasizing result interpretation. Additionally, a comparative analysis is undertaken against similar existing models to assess effectiveness and identify potential improvements.

4.6.1 Result Analysis

The evaluation also utilized real booking data gathered from the server as shown in (Table 4.2) and concentrated on two main aspects: reducing costs and spreading out charging demand.

Table 4.2 Data generated by the EVCMS System

TID	Dh	Pow	PS	AS	ACM	AC	SPS	OC
0	30	8.8	0	0	0	4.6	0	4.6
1	60	17.6	1	1	1	12.0	0	9.0
2	75	22	0	1	0.25	11.2	0	12.0
3	120	35.2	1	1	1	20.8	1	17.8
4	30	8.8	1	1	1	7.6	0	4.6
:								
495	60	17.6	0	1	0.25	9.0	0	9.8
496	30	8.8	1	1	1	7.6	1	4.6
497	75	22	1	0	0.5	12.7	0	11.9
498	120	35.2	1	0	0.5	19.3	0	18.6
499	120	35.2	0	0	0	17.8	0	17.8

(Table 4.2) provides a detailed breakdown of each charging transaction, including the transaction ID (TID), duration of charging (Dh), power consumption (Pow), the peak status of the charging event (PS), the availability status of the charging station (AS), the availability cost multiplier applied (ACM), the actual cost of the transaction (AC), whether the charging event was shifted to off-peak hours (SPS), and the optimized cost after applying various strategies (OC). The below table presents data for 500 charging transactions, allowing for the assessment of the system's performance, and enabling informed decisions to optimize the charging infrastructure. The above data was collected from the server based on booking time, duration, and power requirements. The booking time was categorized as Peak On/Off for further analysis. Thus, the peak status, Availability status, and shifted peak status can be expressed as given by Eq. (4.9), Eq. (4.10), and Eq. (4.11) respectively.

$$\text{Peak Status (PS)} = \begin{cases} 1, & \text{if Peak On,} \\ 0, & \text{if Peak Off.} \end{cases} \tag{4.9}$$

$$\text{Availability Status (AS)} = \begin{cases} 1, & \text{if available,} \\ 0, & \text{if not available.} \end{cases} \tag{4.10}$$

$$\text{Shifted Peak Status (SPS)} = \begin{cases} 1, & \text{if shifted,} \\ 0, & \text{if not shifted.} \end{cases} \tag{4.11}$$

The ACM is factor representing ToU parameter referred from Table 1. It is taken as per the Peak Status and Availability Status. Based on PS, AS and ACM the result is classified into four cases. In the regular system charging cost was generated based on the booking time

may it be in Peak On/Off, but in the optimized system users are given a preference to shift to Peak off with less price. This data generated by the optimized system was compared against the original system, where charging times were not optimized. The results found were quite interesting.

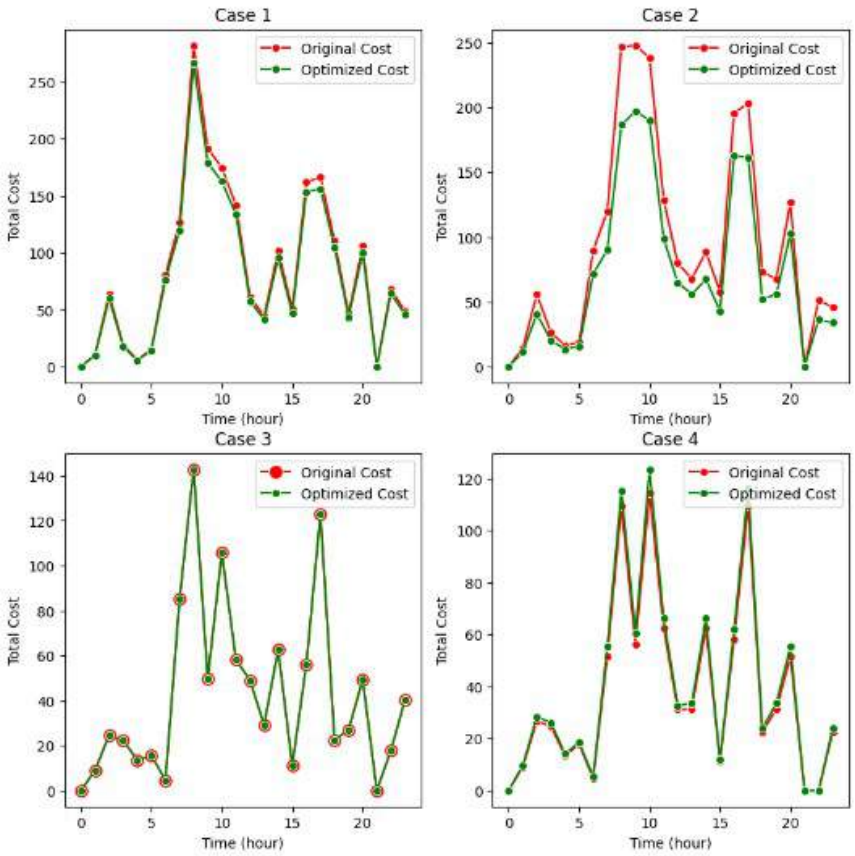


Fig. 4.13 Total charging cost comparison of original and optimized w.r.t system cases

(Fig. 4.13) displays a line plot graph illustrating four distinct system cases. The x-axis represents time in hours, while the y-axis denotes the associated cost. In Case 1 and Case 2, the optimized system demonstrates a noticeable reduction in cost. This reduction is a direct result of efficiently shifting peak-hour bookings to off-peak hours, mitigating costs for users and grid operators alike. Case 3, on the other hand, portrays a scenario where neither the original nor the optimized system incurs a cost change. This situation arises when users schedule their bookings during times when charging stations are both fully available and during off-peak hours. This represents an ideal, cost-neutral scenario. Case 4 presents an intriguing dynamic. Here, bookings are scheduled during off-peak hours in the original system. However, due to limited charging station availability, a cost multiplier of 0.25 is factored into the optimized cost. Importantly, if charging station availability were not an

issue, the original cost in Case 4 would be the same as that in Case 3. Consequently, in Case 4, the optimized cost appears slightly higher, primarily due to the availability constraint imposed by the optimized system.

(Table 4.3) summarizes average cost-related information for the four different cases, each representing a scenario within a system. This table provides a clear comparison between the original cost and the cost after optimization for each case, considering factors like peak demand and system availability. It allows for an assessment of the effectiveness of optimization measures in reducing costs for different system conditions.

Table 4.3 Result of Average charging cost comparison

Case	**Peak Status**	**Availability Status**	**Original Cost**	**Optimized Cost**
Case 1	Peak On	Available	12.44	11.69
Case 2	Peak On	Not Available	13.86	10.86
Case 3	Peak Off	Available	11.45	11.45
Case 4	Peak Off	Not Available	11.39	12.14

(Fig.4.14) illustrates the discount and profit achieved through optimized booking scheduling, with the x-axis representing time and the y-axis indicating costs in GBP. The discounts are presented as cost reductions in GBP for each unique case. Moreover, the figure also showcases profits, expressed in GBP, which are observed exclusively in Case 1 and Case 2. In these cases, bookings are effectively shifted to off-peak hours, resulting in tangible profits. Based on the observed discounts and profits, it becomes evident that optimization has effectively managed costs across multiple transactions, ultimately leading to enhanced system performance and improved cost-effectiveness.

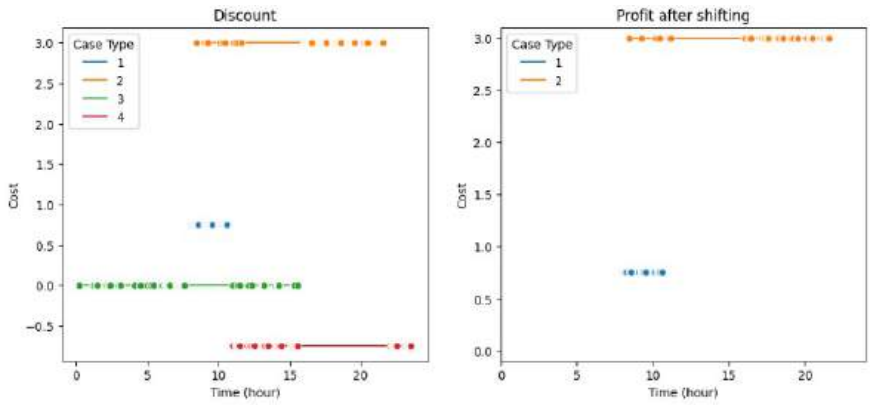


Fig. 4.14 Discount and profit in optimized system w.r.t system cases

To assess the load balancing achieved by the optimized system, a comparison of charging demand and power demand was conducted between two scenarios: before and after optimization. The x-axis represents time in hours, while the y-axis depicts the number of EV bookings for the Charging Demand graph, and the total power demand each hour for the Power Demand graph as shown in (Fig.4.15).

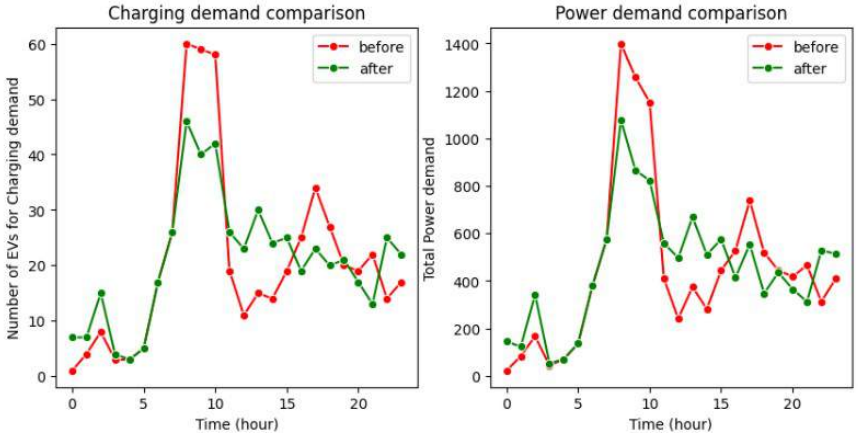


Fig. 4.15 Charging and power demand comparison w.r.t to before and after optimization

It is evident from the graph that before optimization, both the charging demand and power consumption peaked during on-peak hours, potentially straining the grid at these times. However, after optimization, a portion of the EV bookings was intentionally shifted to off-peak hours to reduce costs. Consequently, in the post-optimization scenario, both the charging demand and power consumption also shifted towards the off-peak hours, contributing to a more balanced and efficient utilization of resources.

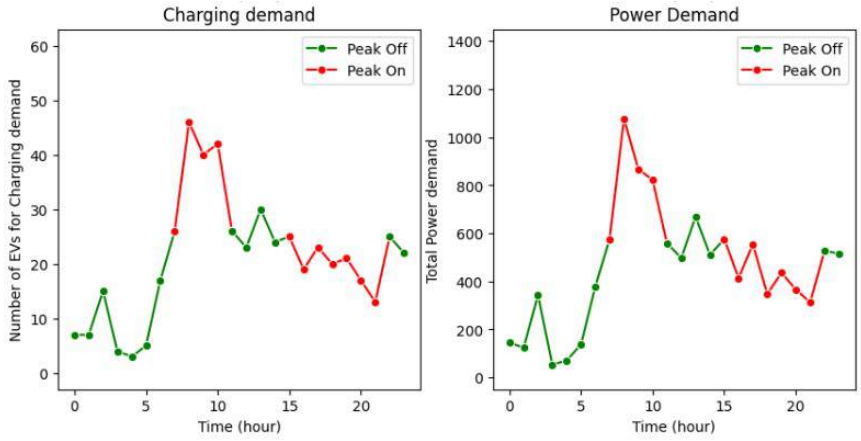


Fig. 4.16 Charging and power demand after optimization w.r.t to Peak off/PeakOn

(Fig.4.16) effectively presents the charging demand and power consumption after optimization, distinctly classifying them with respect to peak on and peak off times. The green colour indicates the charging demand and power consumption during peak off hours, while red signifies the same during peak on hours. This visualization evaluates the performance of the system indicating optimization of resource usage to balance peak demands.

4.6.2 Comparative Analysis

For evaluating the performance of the system against existing systems, the focus was on minimizing the charging cost and balancing charging demands more effectively. One way to balance charging demands in Peak On time is to encourage users to charge their vehicles in Peak Off time by reducing the charging price in Peak Off. Using the current state of art work [140],[141],[142],[129],[8] as a reference point, the proposed EVCMS was compared against these systems. (Table 4.4) provides a comparative analysis between the EVCMS framework and contribution of different researchers, highlighting the advantages of the EVCMS in the context of EV charging systems.

The comparison focuses around five critical properties: cloud infrastructure, OCPP Protocol implementation, optimal charging schedule management, charging cost reduction strategies, and peak load balancing techniques. Upon evaluation, it becomes evident that the proposed EVCMS framework excels in fulfilling all five properties comprehensively. Whereas the other work has incorporated some of the properties from all the mentioned properties. This comparative analysis proves the effectiveness of EVCMS framework with respect to its minimum charging cost, balanced Peak demand loads and enhanced security solution.

Table 4.4 Table based comparison with state of the artwork

Parameters	**EVCMS**	**[140]**	**[141]**	**[142]**	**[129]**	**[8]**
Cloud	Yes	Yes	Yes	Yes	Yes	Yes
OCPP	Yes	Yes	Yes	No	No	Yes
Optimal charge	Yes	No	No	Yes	Yes	Yes
Less charge cost	Yes	No	No	Yes	Yes	Yes
Peak Load Balance	Yes	No	No	No	No	No

The performance evaluation validated the efficacy of the optimization strategy in achieving the dual goals of cost minimization and load balancing. The deliberate shift of Peak ON charging to Peak Off charging not only led to significant cost savings but also fostered better resource utilization and load distribution. These findings provide valuable insights for system enhancement and lay the groundwork for future optimization work.

4.7 Chapter Summary

The EVCMS framework introduced a cloud ready solution for smart charging management that optimizes EV charging dynamics. The system's primary goal is to provide EV users with tailored, optimized charging plans that match their preferences and the charging station's capabilities. This is facilitated through real-time data sharing and an intuitive web interface, streamlining the process of reserving charging slots, thereby minimizing wait times, and ensuring efficient, cost-effective charging strategies. The proposed EVCMS framework allows remote charging process management via OCPP remote procedure calls, providing an additional layer of convenience and control for EV users and charging station operators along with enhanced security. The comparative analysis demonstrated that the EVCMS framework outperforms the existing system, highlighting its potential to reduce charging costs and distribute charging demand more evenly during peak times. This achievement underscores the significance of a cloud ready approach in enhancing the charging experience for EV users and optimizing infrastructure efficiency.

The EVCMS holds promising potential for expansion and improvement. Scaling the system from managing one charging station to multiple stations can greatly enhance its scalability, enabling a wider network of charging infrastructure to be efficiently managed and optimized. This evolution would not only accommodate a larger user base but also facilitate load balancing across various charging points. By aggregating and distributing charging demand intelligently, the EVCMS could contribute to a more evenly distributed load, preventing congestion during peak periods. Expanding the existing EVCMS framework to include multiple charging stations presents a synergistic approach. As the network grows, the model becomes more refined, enhancing load balancing across a broader spectrum of charging points. Users would experience improved convenience due to minimized wait times, while charging infrastructure operators would benefit from optimized resource allocation. Ultimately, this approach aligns well with the primary goal of creating a robust, efficient, and user-centred EV charging ecosystem with improved security.

It is important to note that while the proposed EVCMS framework has been designed with a cloud ready architecture in mind, the current implementation has been deployed in a local environment for development and testing purposes. The modular design ensures that future migration to a cloud infrastructure, such as that required by JMVL, can be achieved with minimal modifications. This planned transition will be explored in future work.

Chapter 5

Hybrid-EVCMS Framework

The chapter proposes an H-EVCMS framework that combines the strengths of both centralized and distributed charging management models. This hybrid approach overcomes the limitations of each paradigm, providing a more robust, secure, and effective EV charging management system. In this chapter **Section 5.1** provides background information on the hybrid EVCMS framework, explaining the need for such an approach. **Section 5.2** introduces the proposed H-EVCMS framework and its system design. **Section 5.3** explains the system methodology, focusing on the development of algorithms for determining weights for charging stations (CS), finding CS availability, and optimizing charging plans within the H-EVCMS framework. **Section 5.4** demonstrates the integration of OCPP into the H-EVCMS framework to enhance cybersecurity in charging management, addressing session management issues in OCPP to improve security. **Section 5.5** provides a performance evaluation, analysing the system's efficiency by comparing it with the centralized EVCMS model. This section also includes a case study analysis of the proposed model across different scenarios. Finally, **Section 5.6** summarizes the key findings of the chapter.

5.1 Background

There has been growing global awareness about the environmental issues caused by vehicle emissions. This is driving the need for cleaner transportation alternatives[2]. EVs provide numerous benefits compared to traditional combustion engine vehicles [3]. In urban settings, air pollution is a significant public health issue, but EVs help mitigate this by running on electricity, which reduces carbon emissions. As a result, EVs are considered a key solution for creating a more sustainable transportation system. Recent advancements in EV technology have spurred widespread adoption [125], with innovations in battery design and materials enhancing energy density, increasing driving range, and reducing charging

times. Furthermore, ongoing improvements in battery technology are lowering the long-term operational and maintenance costs of EVs, making them a more affordable option for consumers [126]. With their environmental advantages and technological progress, the EV sector is rapidly expanding. However, to accommodate the growing number of on-road EVs, efficient and secure charging infrastructure management is essential. Charging stations (CSs) are becoming increasingly widespread and effective, addressing the issue of range anxiety [143], thus making EVs a more practical choice for a wider range of consumers.Overall, technological developments are strengthening the case for EV as a clean, economical, and sustainable transportation solution [144]. However, the rapid adaptation of EV has brought significant cyber security challenge related to EV charging management [128]. If these challenges are not addressed, the EV network system can degrade in efficiency and reliability, causing potential harm to citizens[145].

EV charging infrastructure management can be divided into two broad categories including centralized and distributed. In the centralized charging management approach, a central controller communicates with EV to obtain charging details and generates an optimized solution for managing the charging sequence [146]. However, this approach can become increasingly complex as the number of EVs on the road continues to rise rapidly. It can lead to optimization issues, long waiting times at CSs, load balancing problems, and cyber risks [147]. The distributed charging framework emerged as a promising alternative to address the challenges of the centralized system. It offers a more user-friendly approach, allowing EV drivers to determine their charging slots based on user requirements [148]. While this user-friendly approach has advantages, it also poses serious cyber challenges for the energy distribution network. The distributed framework has limited optimization capabilities, lack of coordinated load management across the grid, and more cyber interfaces. It does not guarantee optimal and secure charging solution due to the autonomy of individual EV in their charging decisions. This can also result in potential energy load imbalances due to inefficient energy consumption [60].

To address these challenges in EV charging, an H-EVCMS framework is proposed in this paper. As shown in (Fig.5.1), this framework is a cloud ready security-centric smart charging management system. Consider multiple CSs coordinated by a central controller, rather than relying on individual stations. When EV users submit charging requests, local controllers communicate with the central controller to identify suitable, secure and optimal charging slots. The framework ensures secure optimization while balancing charging loads across multiple CS. The H-EVCMS framework intelligently distributes the load across CS and leveraging the advantages of both centralized and distributed charging management. The central controller maintains efficient, secure, and coordinated charging, while the distributed

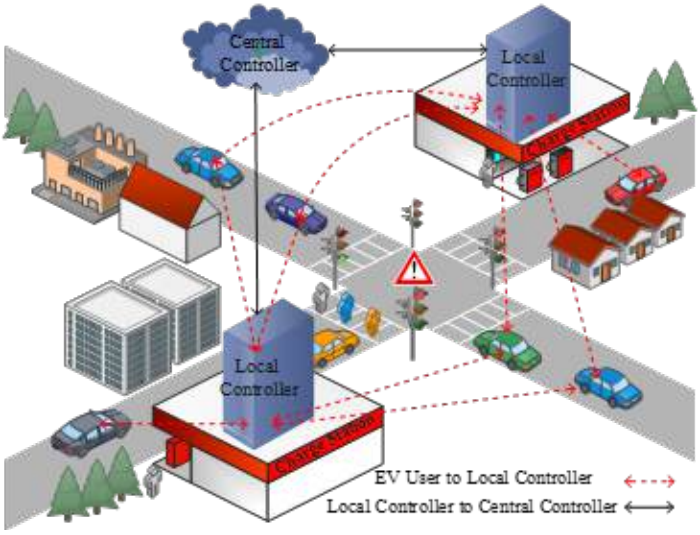


Fig. 5.1 H-EVCMS Framework

architecture allows flexibility and scalability. This approach addresses the drawbacks of both paradigms, leading to a more robust, secure and effective EV charging management.

The proposed H-EVCMS framework aligns with sustainable transportation trends by enhancing the efficiency and reliability of EV charging infrastructure. By optimizing charging slots and balancing loads across multiple CSs, it promotes grid stability, reduces energy waste, and encourages the widespread adoption of EVs. Additionally, its cloud ready, security-centric approach ensures secure and coordinated charging operations, contributing to the overall advancement of sustainable transportation initiatives.

5.2 System Design

The H-EVCMS framework implementation incorporates the system design detailing the enhancements made to single station EVCMS framework to multiple station H-EVCMS framework. Due to increasing demand for EV charging solutions, a hybrid charging infrastructure system was implemented. This system combines elements of both a centralized and a distributed charging infrastructure. It allows users to set their charging preference based on time and power requirement and generate optimized charging plans accordingly. This hybrid approach retains user autonomy by incorporating their preferences, but it also employs centralized optimization to generate plans. It combines the benefits of distributed decision-making with centralized optimization.

5.2.1 EVCMS framework

Initially EVCMS framework was designed to accommodate the needs of a single CS as detailed in chapter 4. (Fig.5.2) shows the single station EVCMS layered framework. Here there are mainly 2 layers, the device layer and Local control layer. The device layer is responsible for connecting EVs to Charge point of CS using ISO15118 protocol. And the Local control layer is responsible to connect Charge point of CS to Local controller using OCPP.

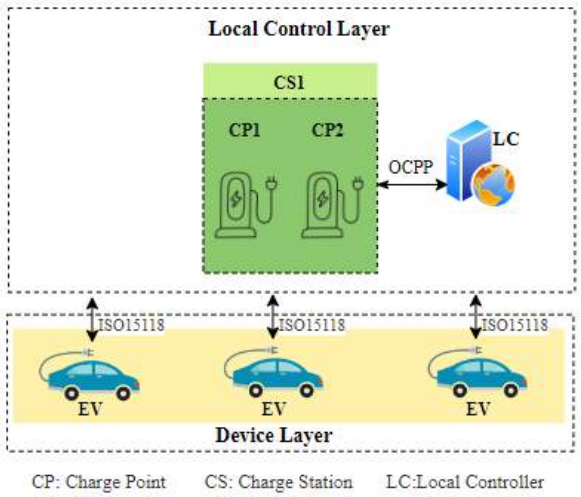


Fig. 5.2 EVCMS layered framework

The system presents a centralized platform designed to oversee EV charging operations and reservation tracking, and a distributed platform for user autonomy. It utilizes the OCPP protocol to establish seamless communication with CS, facilitating efficient management of the entire charging process. Supporting multiple charging points simultaneously, the system ensures a user-friendly interface that enhances the experience for EV users. This approach offers benefits in terms of load management and energy cost optimization. The framework's centralized control enables the aggregation of EV charging data and facilitates the implementation of optimization algorithms to create efficient charging schedules. Optimization techniques play a pivotal role in this framework. The algorithm developed considers factors such as charging pricing, CS availability, Peak ON/OFF time and user preferences to determine the optimal charging schedule for each EV. By dynamically adjusting charging rates and schedules, the framework aims to optimize charging plans and mitigate the impact of high charging loads on the grid, especially during peak demand periods.

As the adoption of EVs continued to rise, it became evident that scaling the EVCMS framework to accommodate a larger number of EVs posed challenges. This evolution from a

single-station model to a multiple-station network was not merely a response to the growing demand for EV charging but a strategic adaptation that sought to leverage the strengths of both centralized and distributed approaches. The single-station EVCMS framework may encounter several challenges that hinder its efficient operation. This framework may struggle to scale efficiently when charging demand increases in EV infrastructure. Balancing the load on a single station while accommodating user preferences can also be challenging.Thus, the system might encounter difficulties in managing increased user preferences and generating optimized plans. As the system aims to provide tailored plans, the optimization process could become intricate with increasing numbers of EV charging requests. To address these issues, the framework was extended to include multiple CSs, creating a more robust solution capable of handling a larger user base effectively.

5.2.2 H-EVCMS framework

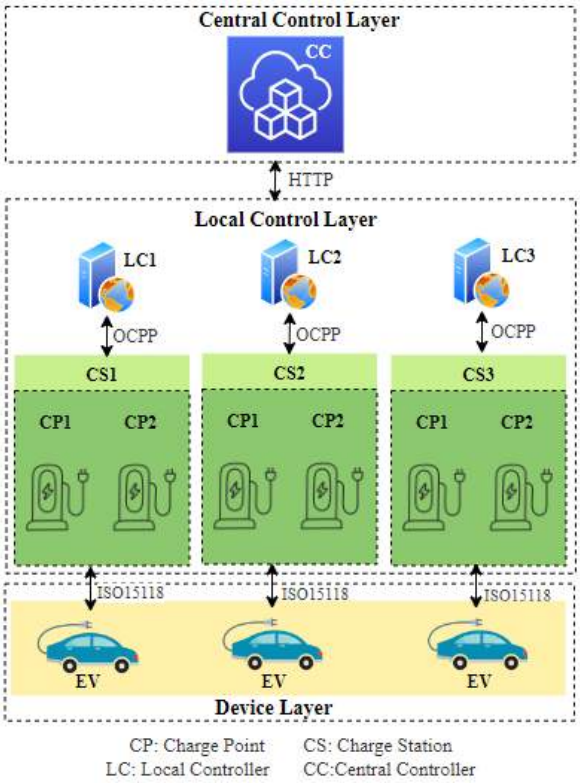


Fig. 5.3 H-EVCMS layered framework

The H-EVCMS framework, is designed to accommodate the needs of a multiple CS in order to address the challenges faced by EVCMS framework. (Fig.5.3) shows the multiple station H-EVCMS layered framework. This framework is comprised of 3 layers, the device

layer, local control layer and central control layer. The device layer is responsible for connecting EVs to charge point of CS using ISO15118 protocol. And the Local control layer is responsible to connect charge point of CS to Local controller using OCPP. Finally the central control layer is responsible for connecting different local controller to central controller to communicate data for controlled decision.Expanding the EVCMS framework to multiple CSs inherently enhances scalability. With a distributed network of stations, the framework can easily accommodate increased demand without overburdening any single station. This framework is a cloud ready EV charging management system that centrally oversee and coordinate multiple CS across a distributed network. Within this H-EVCMS framework, each CS is equipped with the necessary communication capabilities to interface with the server. This communication allows the server to gather real-time data from each CS, including availability and power consumption. By harnessing this data, the server dynamically chooses a CS with less weight and optimizes the charging schedule. This mechanism ensures efficient utilization of resources and load balancing across the charging network.

Moreover, the framework offers a user-friendly interface accessible through web application. EV owners can conveniently monitor the availability of CSs, make reservations, and receive real-time updates on their charging sessions. This enhances user convenience and provides a seamless charging experience. Thus, the enhanced system retained its user-preference-driven approach while incorporating an intelligent allocation mechanism that considered station availability across the network of charging points. This expansion allowed for better load distribution, reduced wait times, and optimized energy usage, all while maintaining individual user preferences. By combining user autonomy with centralized optimization, the enhanced system seeks to deliver an improved charging experience that aligns with both individual needs and the broader goal of efficient energy distribution.

5.3 System Methodology

This section details the structure and operations of the H-EVCMS framework, featuring three algorithms. The first determines the weight of CS for load balancing, the second checks for availability in CS, and the third generates optimized charging plan based on Time of use charge. All these algorithms improve the framework's efficiency and enhance the EV user experience.

5.3.1 CS Load Balancing

Algorithm 5.1 Determine Weights of CSs

Input : ChargStations, Bookings, DateTime,
Constant A=0.6 (Weight of 'TotalBookings')
Contant B=0.4 ('TotalPower')

Output : List of CSs sorted in ascending order according to their respective 'weight'

Initialize:
$stations = []$ **Start:**
for *ChargStationId* ***in*** *ChargStations* **do**
    $chargStationBookings = Find$(bookings for ChargstationID)
    $TotBooking =$**length**$(ChargStationBookings)$ $TotPwr = 0$
    for *booking* ***in*** *ChargStationBookings* **do**
        $TotPwr = TotPwr$ **+** $booking.expectPwr$
    $weight = A \times TotBooking$ **+** $B \times TotPwr$ **Add** current CS data and its respective weight to the *stations* list
Sort the list of *stations* in ascending order for each CS's weight
return *list of stations*

The (Algorithm 5.1) for determining the weights of CSs in the H-EVCMS framework is designed to facilitate effective load balancing. It takes the following inputs: *ChargingStation-Data*, a list of charging stations (CSs) with their respective connectors; *Bookings*, a list of user reservations for the requested date; and *DateTime*, the specific date and time for which scheduling is requested. It also uses two constants: $A = 0.6$ and $B = 0.4$, representing the weights assigned to the total number of bookings and total power consumption, respectively. These values were empirically selected to prioritize user demand (bookings) while still accounting for infrastructure load (power usage). While a balanced ratio such as 0.5:0.5 could have been used, giving a slightly higher weight to the number of bookings helps reduce wait times and improve user experience by favouring stations that are less congested. The algorithm initializes an empty list called *stations* to store each CS's data along with its computed weight. It then iterates through each CS in the input data, filtering bookings that match the corresponding CS ID. For each station, it calculates the *TotalBookings* and *TotalPower* using the relevant entries using below defined equation. The combined weight is calculated based on these two factors and their respective constants. If a CS has zero bookings and zero power usage, instead of assigning a weight of 0 (which would eliminate it

from consideration), a minimum default weight of 1 is assigned. This ensures fair scheduling opportunities for all stations and promotes the utilization of underused infrastructure.

$$\textbf{TotalBookings} = \sum_{\substack{\text{booking} \in \text{Bookings} \\ (\text{booking.ChargStationID} == \text{station.ID})}} \text{booking.ChargstationBookings} \tag{5.1}$$

$$\textbf{TotalPower} = \sum_{\substack{\text{booking} \in \text{Bookings} \\ (\text{booking.ChargStationID} == \text{station.ID})}} \text{booking.PowerConsumption} \tag{5.2}$$

The weight of the CS (Weight) is then determined using the specified constants:

$$\textbf{Weight} = A \times \text{TotalBookings} + B \times \text{TotalPower} \tag{5.3}$$

The station data along with their weights is added to the list of stations. The CS are then arranged in ascending order based on their weights:

$$\textbf{CSsorted} = \text{sort}(\text{CS}, \text{by } W) \tag{5.4}$$

The system selects a CS with the least weight for booking:

$$\textbf{CS}_{\text{selected}} = \arg\min(W) \tag{5.5}$$

This step ensures that stations with lower weights, indicating fewer bookings or lower power consumption, come first in the sorted list. The algorithm returns the sorted list of stations as its output. This prioritization is crucial for optimizing the allocation of charging resources, especially in situations where different stations may experience varying levels of demand and power requirements.

5.3.2 CS Allocation

The(Algorithm 5.2) for checking availability and selecting a CS along with a connector is designed to determine the feasibility of accommodating a new charging request at the user-specified date and time (DateTime). It also considers the desired charging duration and the pre-sorted list of CSs (Stations) based on their respective weights. The availability of the selected CS is determined:

$$\text{Availability}_{\text{CSi}} = \begin{cases} 1, & \text{if slot is available} \\ 0, & \text{otherwise} \end{cases} \tag{5.6}$$

Algorithm 5.2 Select charging station

Input : DateTime (requested date time), Duration, Stations,Bookings

Output : isAvailable, finalDateTime, selectdChargStation, selectdConnectorId

Initialize:
isAvailable =`false` *finalDateTime* =`null` *selectdChargStation* =`station[0]` *selectdConnectorId* =`station[0].Connector[0]` *startTime* =`get time from DateTime` *endTime* =`startTime + duration`

Start:
for *station* ***in*** *Stations* **do**
  for *connectorId* ***in*** *station* **do**
    isConflicting = *False connectorBookings*=*Find*(bookings for stationid AND connectorId)
    for *booking* ***in*** *ConnectorBookings* **do**
      Check if booking is conflicting **if** *isConflicting* == *True* **then**
        break
      continue *to check conflict in next booking*
    if *isConflicting* == *True* **then**
      continue *to check conflict in next connector*
    isAvailable = *true* *selectdChargingStation* = *station* *selectdConnectorId* = *connectorId* *finalDateTime* = *DateTime* **break**
  if *isAvailable*==*True* **then**
    break
  continue *to check availability in next station*
return *isAvailable, finalDateTime, selectdChargingStation, selectdConnectorId*

The algorithm initializes the availability status (isAvailable) as false and defaults the selected CS (selectedChargingStation) and connector ID (selectedConnectorId) to the first station and its first connector, respectively. It calculates the end time based on the requested start time and duration. It then iterates through each CS and its connectors to find a suitable combination that does not conflict with existing bookings. For each connector, it filters the bookings associated with that specific CS and connector combination. It checks for conflicts by comparing the requested charging time with the existing bookings for the same station and connector. If a conflict is found, the algorithm continues searching through the

next connector; otherwise, it marks the availability status as true and breaks out of the loop, signifying a successful finding of an available charging slot.

The algorithm returns the availability status, the final date and time (either the requested time or a newly assigned one based on availability), the selected CS, and the chosen connector ID. This approach ensures that the algorithm optimally selects the first available charging slot from the sorted list of CSs, taking into account existing bookings and their respective time slots. This scenario is best explained in (Fig.5.4) showing process of determining CS weight and assigning EV considering the weight and CS availability.Here the weight of each CS was calculated, then sorted each CS weight in ascending order. For instance W_{22}, that is the weight of CS_2 with CP_2 has the least weight. Even though CS_{22} has the less weight but its availability is shown in red indicating not available at the given time slot so algorithm chooses the next CS from the sorted list. Thus, CS_{11} that is CS_1 with CP_1 is selected as it has the next less weight and its available as shown by green.

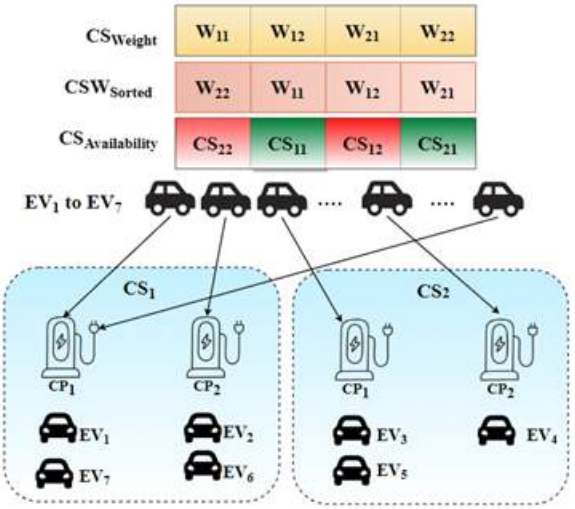


Fig. 5.4 Process of determining CS weight and assigning EV

5.3.3 Charging Price Optimization

The (Algorithm 5.3) for generating plans is designed to create effective charging plans for EVs by assessing peak periods and calculating cost multipliers based on availability.The Peak Status for a charging slot is determined:

$$\text{PeakStatus} = \begin{cases} 1, & \text{Peak On} \\ 0, & \text{Peak Off} \end{cases} \tag{5.7}$$

Algorithm 5.3 Generate Plan

Input : isAvailable, finalDateTime, duration(Dh), CostOfEnergy (Ec), Chargingrate(Crate), peakDemandCharge(Pdc), ServiceCharge(SCcs)

Output : List of Plans

Start:
if *((dT ≥ 8am AND dT ≤ 11am) OR (dT ≥ 4pm AND dT ≤ 10pm))* **then**
    isPeakOn=true
isPeakOn=false
if *isPeakOn == true* **then**
    if *isAvailable* **then**
        availabilityCostMultiplier(ACM) = 0.5
    availabilityCostMultiplier(ACM) = 1
if *isAvailable* **then**
    availabilityCostMultiplier(ACM) = 0
availabilityCostMultiplier(ACM) = 0.25
TimeOfUseCharge (ToU) = ACM × Pdc
ChargingPrice = (Ec × Dh × Crate) + ToU + SCcs
if *isPeakOn == false* **then**
    Power = Dh × Crate
    Plan 1: Based on 'duration' adjust power
    Plan 2: Based on 'power' adjust duration
Power = Dh × Crate
Plan 1: Based on 'duration' adjust power
Plan 2: Based on 'power' adjust duration
dateTime = get a time-slot available in the peakOff
Plan 3: Based on 'duration' adjust power in peakoff
Plan 4: Based on 'power' adjust duration in peakoff
Return: List of Plans

It considers the user's specified arrival time, duration, and power requirements to generate plans tailored to their needs. Plans may involve adjusting power or duration to meet constraints, and Eco plans are introduced by shifting time slots to peak off periods for cost optimization. The algorithm outputs a list of plans, accounting for user preferences, availability constraints, and cost considerations. The ACM used in this algorithm plays a significant role in pricing decisions by adjusting the ToU charges based on real-time peak status and

station availability. As shown in Table 4.1 in Chapter 4, ACM is derived by combining two binary conditions: whether the time falls in a peak window, and whether the charging station is available. This structured approach ensures that charging is incentivized during off-peak hours and at stations with higher availability, thereby supporting load balancing and cost efficiency. For these plans, the algorithm calculates the overall charging price using the formula below, which is derived and explained in detail in Chapter 4.

$$\textbf{EV Charging Price} = (Ec * Es) + ToU + SCcs \tag{5.8}$$

(Fig.5.5) illustrates the flowchart of the H-EVCMS framework algorithm. The flowchart visually guides the process of resource management and overall optimization process.The process begins by calculating the weight of each CS based on factors such as the total number of bookings and total power consumption estimation for a specified date. Subsequently, the CS are arranged in ascending order according to their calculated weights. The framework selects a CS with the least weight for booking. Upon selection, the system checks the availability of slots in the chosen CS. If a slot is available, the framework proceeds to examine its peak status and determines the Availability Cost Multiplier. The EV charging price is then calculated, and optimized plans are generated based on this information. However, if the selected CS does not have availability for the requested time, the framework systematically checks the availability with the next CS in the weighted list. This process continues until a CS with an available slot is found. If no available slot is found in the subsequent CSs, the framework suggests the next available slot in the initially selected CS. Once an available slot is secured, the subsequent steps of calculating the charging price and generating optimized plans remain consistent. This comprehensive approach ensures the efficient utilization of CSs, considering factors such as availability, peak status, and cost optimization in providing a user-friendly EV charging experience.

The complexity analysis of algorithm is to assess its computational efficiency in terms of time and space.In the initial steps, the algorithm computes TotalBookings and TotalPower for each CS by linearly scanning through all bookings (Nb). This process has a time complexity of O(K× (Nb)), where K represents the number of CSs. Following this, the algorithm calculates weights for each CS, considering both TotalBookings and TotalPower. The time complexity for this step is also O(K× (Nb)).The subsequent sorting of CSs based on their weights introduces a time complexity of O(K× log(K))in the average case. However, selecting the CS with the minimum weight is a constant time operation O(1). Additionally, operations such as checking availability, determining peak status, and calculating costs are all constant

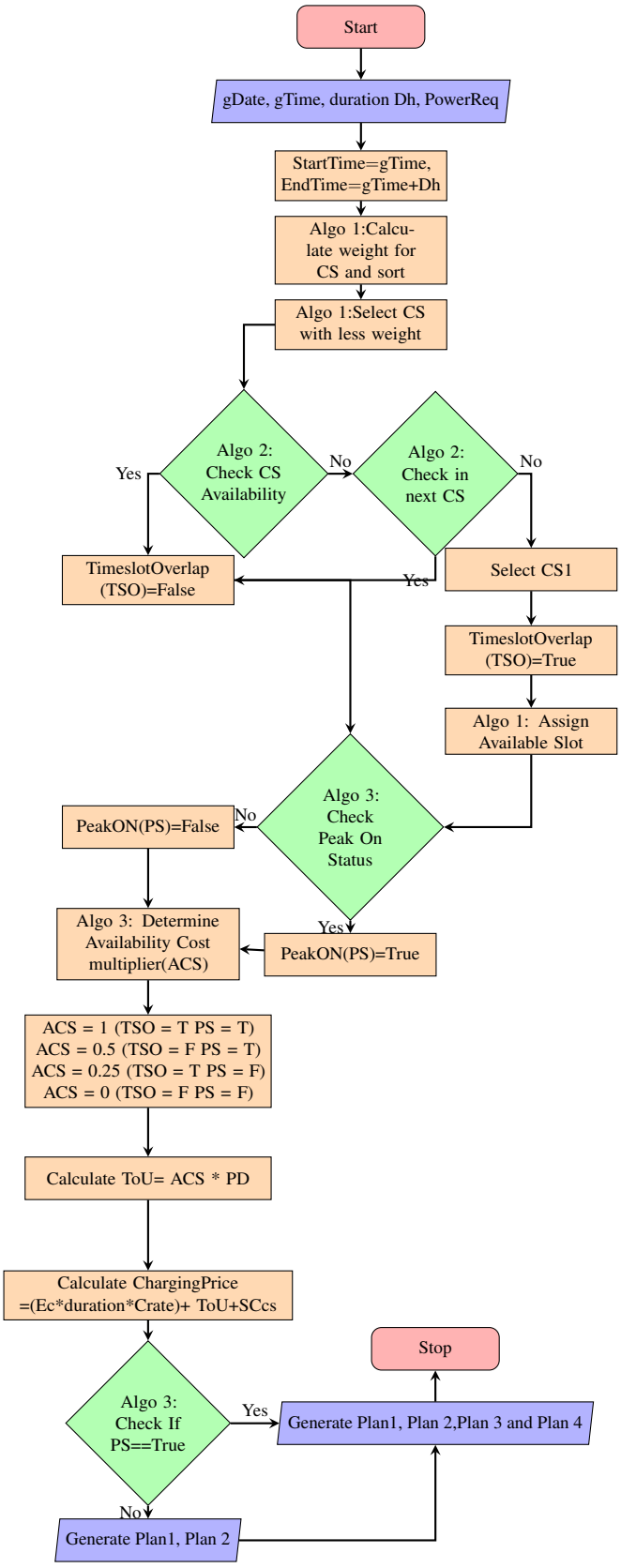


Fig. 5.5 H-EVCMS Framework Flowchart

time O(1).The overall time complexity of the algorithm can be expressed as:

$$O(K \times N_b) + O(K \times \log K) + O(1)$$

In terms of space complexity, the algorithm requires space proportional to the number of CSs O(K) for storing CS data and weights. The space complexity of the linear sorting algorithm is O(log(K))and other variables have constant space requirements O(1).The overall space complexity of the algorithm is:

$$O(K) + O(\log K) + O(1)$$

The algorithm demonstrates efficiency in managing multiple CSs while considering various factors, including booking metrics, weights, availability, peak status, and cost optimization.

5.4 System Security

Within H-EVCMS framework, OCPP chargebox simulator is integrated to provide communication between charge point and charge management system. The OCPP chargebox simulator was taken from the repository [149] to integrate with H-EVCMS framework. The OCPP protocol was defined, to use connect, authorize, start, stop, meter value and disconnect function of chargebox simulator. These function ensures encryption of communication channel to avoid unauthorized access. When an EV user start any communication with the chargebox, its always initiates with a handshake authenticating every user with correct reservation details. It also ensures the date and time of the reservation, making sure that user can connect only at the reserved time slot. Even though this encryption ensures authorization of communication channel in use, it does not authenticate each session.

The OCPP implementation should have robust session management to ensure that once a session is started with specific reservation details, it cannot be duplicated or reused by another user until the session is properly terminated. This could be explained by (Fig.5.6) showing compromised session management by OCPP chargebox simulator. In this scenario, the normal user is connected to chargebox simulator at the reserved time slot and at the same time malicious user is duplicating the same reservation details. This is seen in(Fig.5.6) that both users are using same booking ID at the same timestamp and their charging status is also same. In order to address this issue, reservation locking and concurrency check needs to be integrated in current OCPP chargebox simulator to have robust session management.

By implementing a reservation locking mechanism,each reservation will be associated with a single charging session. When a user initiates a charging session using reservation

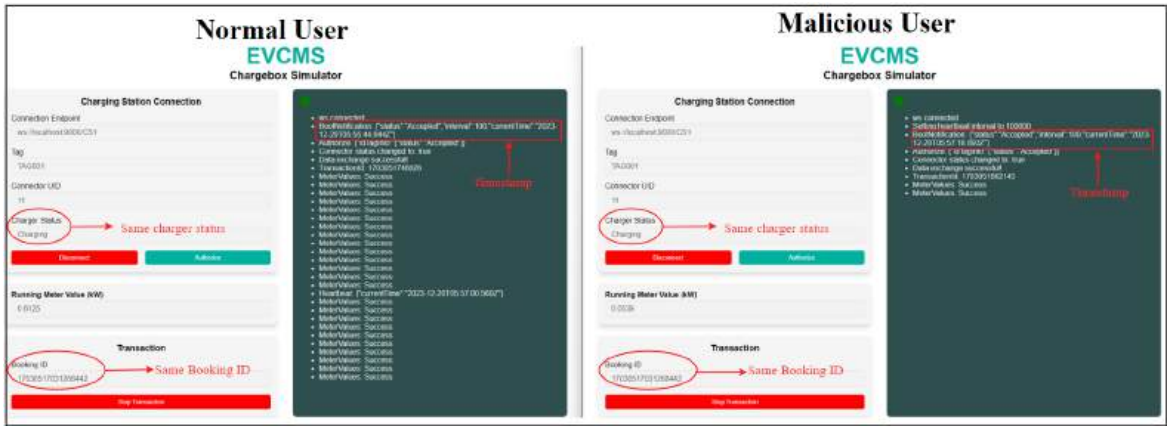


Fig. 5.6 Visualization of Compromised Session Management in the OCPP Chargebox Simulator

details, it will ensure that the reservation is marked as (in use) to prevent concurrent usage. So if any other user attempts to start a session with same reservation details, the system will check if these details are currently associated with an active session or not. If its in use of an active session then charger status of another session will be offline, not allowing concurrent access to same booking. Thus avoiding concurrent usage of same reservation details by multiple users. The charging session management algorithm as given in (Algorithm 5.4) uses a basic lock mechanism to set control access. When starting a session, it checks if the reservation details are in use of any existing transaction, preventing concurrent access. If any booking ID is in use then it updates the charger status as occupied, otherwise it allows the charging by updating the charger status as available. This enhancement made in OCPP chargebox simulator will secure the session, as shown in (Fig.5.7). In this scenario,the normal user is connected to chargebox simulator at the reserved time slot but now the malicious user is not allowed to start the session with the same reservation details as it was allowed before. If there is concurrent access to the same reservation details while it's already in use, the charging status will be displayed as occupied.

Algorithm 5.4 Charging Session Management

Input : *Transactions*, *bookingId*

Output : *Charger Status*

Start:
Initialize: *allowTransaction* = True **for** *transaction in transactions* **do**
    if *transaction.bookingId* == *bookingId* **then**
        allowTransaction = False **break**
if *allowTransaction* == *False* **then**
    Print("Charger status: Occupied")
else
    Print("Charger status: Available")

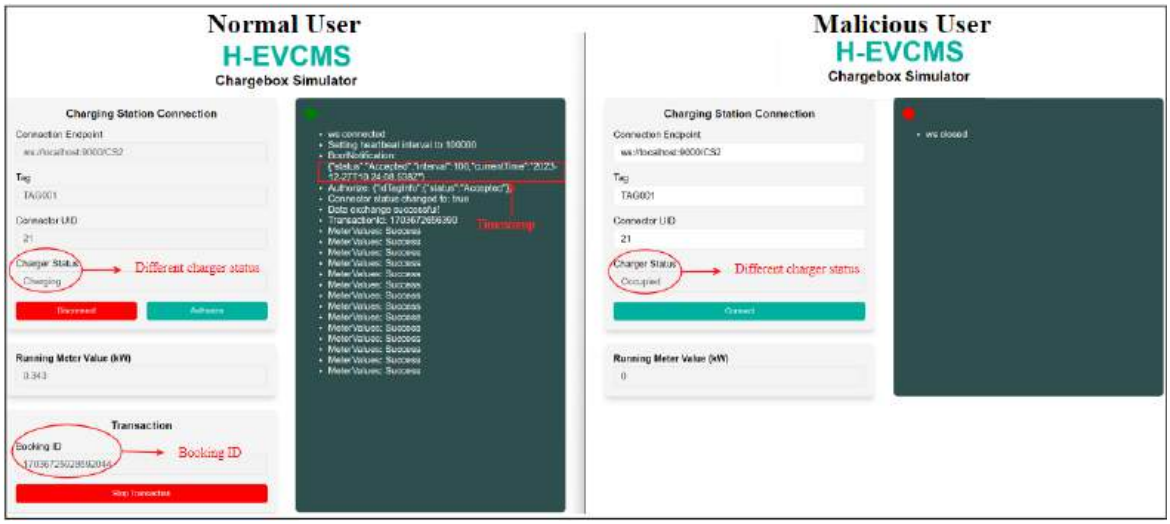


Fig. 5.7 Visualization of Secure Session Management in the OCPP Chargebox Simulator

The computational complexity of the proposed H-EVCMS framework security relies on several crucial operations. Firstly, there's the Authorization and valid reservation check. Before granting access to the charging infrastructure, the H-EVCMS system verifies the provided booking ID and authorizes the user. This involves checking the booking ID against the booking list for validity. The time complexity of this operation is O(n), where 'n' represents the number of booking IDs. As the number of booking IDs grows with the increasing number of users and charging sessions, the computational requirement for validating reservations also increases. However, the linear time complexity ensures that the validation process remains efficient even as the system scales to accommodate a larger user

base and more charging sessions. Secondly, there's the session Locking and Concurrency Check. To maintain data integrity and prevent conflicts, the H-EVCMS system performs concurrency checks and updates charger statuses by traversing the list of active sessions. The time complexity of this operation is also O(n), where 'n' represents the number of active sessions. As the number of active sessions increases due to a higher volume of EV charging activities, the computational requirement for concurrency checks and status updates also rises. However, the linear time complexity allows the system to handle concurrency efficiently, ensuring that charging sessions can proceed smoothly without conflicts or data inconsistencies.

Due to rapid growth in EV adoption, scalability is crucial for the H-EVCMS framework. The algorithms used exhibit linear time complexity, which means that their computational requirements increase proportionally with the size of the input data (i.e., the number of booking IDs and active sessions). This scalability ensures that the H-EVCMS system can accommodate a growing number of users and charging sessions without experiencing significant performance degradation. Additionally, the system's modular design allows for easy scalability by adding more computing resources or optimizing algorithms as needed to handle increasing workload demands.

The H-EVCMS system was developed based on the OCPP protocol. To improve the security of the system, user authorization was conducted before granting access to the charging infrastructure. Additionally, robust session management was implemented by leveraging the real-time access feature provided by OCPP, ensuring that the start and stop of charging transactions aligned with the reserved time slots. Furthermore, advanced features not supported by OCPP, such as session locking and concurrency check, were incorporated. Charging reservation authorization was also introduced, where each reservation was verified against the current date and time to grant access only if it matched the reservation details. Moreover, to enhance the overall security posture of the system, communication with the server was conducted via the HTTPS protocol.

Table 5.1 Comparison of H-EVCMS with current state of art

Security Feature	**H-EVCMS**	**[14]**	**[141]**	**[8]**
User Authorization	Yes	Yes	Yes	Yes
Robust Session Management	Yes	Yes	Yes	Yes
Reservation Authorization	Yes	No	No	Yes
Concurrency Control	Yes	No	No	No
HTTPs Communication	Yes	No	No	No

A comparative analysis was conducted with similar works by researchers utilizing the OCPP protocol, as depicted in the (Table 5.1). The results demonstrated that the H-EVCMS system outperformed other models in terms of security features and overall performance.

5.5 Performance Evaluation

The following case study evaluates the practical implementation of the H-EVCMS framework in optimizing EV charging operations in a metropolitan area. The aim is to compare the efficiency and resource utilization between two different EV charging scenarios.

5.5.1 Case Study: Implementation

This section provides an overview of H-EVCMS framework implementation and assessment of its spread measure using two different EV charging scenarios. The H-EVCMS framework introduces a significant enhancement, allowing seamless integration of multiple CSs. In (Fig.5.8), the user interface of H-EVCMS is depicted from the server side, presenting a comprehensive view of daily bookings and power consumption for each CS. This graphical representation effectively illustrates the distribution of bookings and power consumption, providing CS management users with enhanced insights for better decision-making.To assess the result of H-EVCMS framework two different EV charging Scenario is taken into consideration.Scenario 1, which involves a single CS with two charge points, and Scenario 2, where multiple CSs are available, each equipped with two charge points. Our objective is to compare the efficiency and resource utilization of these two scenarios.

In scenario 1 configuration, all user bookings are concentrated on a single CS with two charge points. This design may result in congestion and heightened demand on the single station alone. The bookings matrix B1 and power consumption matrix P1 represent the allocations specific to CS 1 in scenario 1 as follows:

$$B_1 = \begin{bmatrix} B_{\text{CS1, CP1}} & B_{\text{CS1, CP2}} \end{bmatrix}$$

$$P_1 = \begin{bmatrix} P_{\text{CS1, CP1}} & P_{\text{CS1, CP2}} \end{bmatrix}$$

In scenario 2 configuration, bookings are distributed across multiple CSs, and a weighted calculation incorporates both the total number of bookings and power consumption. The booking matrix B2 and power consumption matrix P2 illustrate diverse allocations across CSs 1 and 2 in scenario 2 as follows:

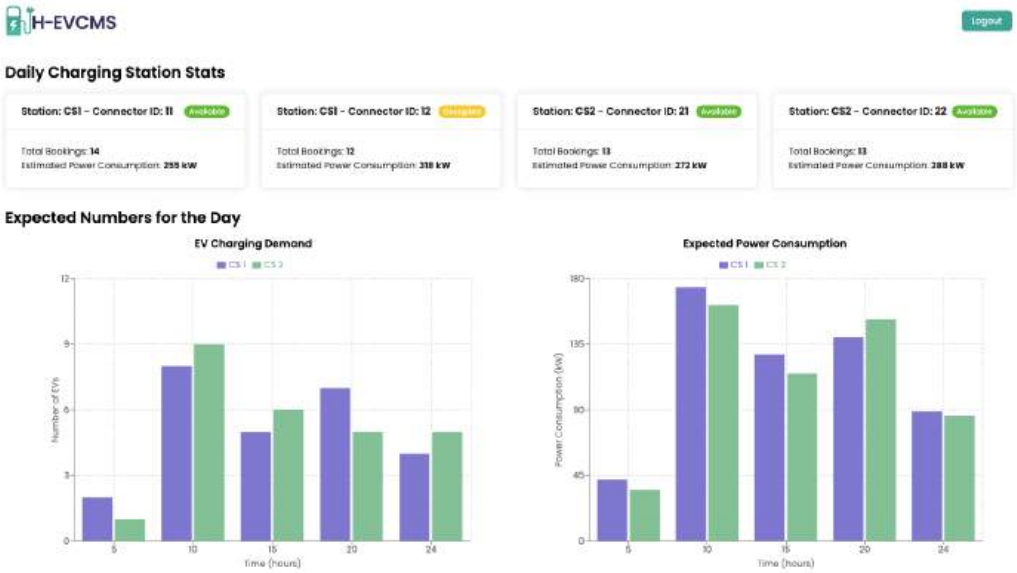


Fig. 5.8 Server side user interface of H-EVCMS

$$B_2 = \begin{bmatrix} B_{\text{CS1, CP1}} & B_{\text{CS2, CP1}} \\ B_{\text{CS1, CP2}} & B_{\text{CS2, CP2}} \end{bmatrix}$$

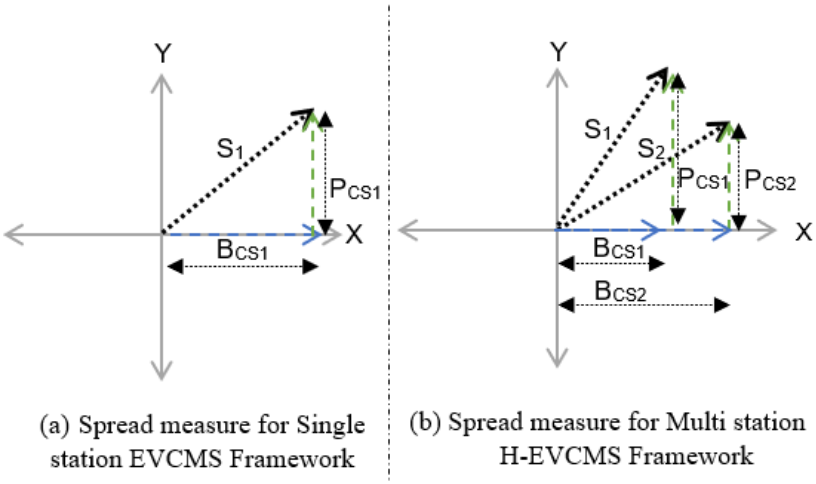
$$P_2 = \begin{bmatrix} P_{\text{CS1, CP1}} & P_{\text{CS2, CP1}} \\ P_{\text{CS1, CP2}} & P_{\text{CS2, CP2}} \end{bmatrix}$$

To gauge the performance, we introduce a spread measure S, treated as a vector with booking and power consumption as its components. This measure captures the balance and distribution of these factors across CSs. The vector S allows for a comprehensive assessment of the relationship between bookings and power consumption, providing insights into their distribution across different CSs in scenario 1 and scenario 2.

$$\vec{S} = \begin{bmatrix} B_{\text{CS}} \\ P_{\text{CS}} \end{bmatrix}$$

(Fig.5.9) illustrates the spread measure vector representation for H-EVCMS scenarios. In (Fig.5.9a), the spread measure vector represents the EVCMS scenario with 1 CS and (Fig.5.9b) showcases the spread measure vector for the H-EVCMS scenario with 2 CSs.

The vectors visually depict the magnitudes and directions of the spread measures, providing insights into the allocation and utilization of charging resources in each scenario. For better understanding two scenarios are further explained with data points in EVCMS and H-EVCMS framework.



(a) Spread measure for Single station EVCMS Framework

(b) Spread measure for Multi station H-EVCMS Framework

Fig. 5.9 Spread measure vector representation

Scenario 1 (EVCMS framework)

$$B_1 = \begin{bmatrix} 338 & 162 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 7300 & 3577 \end{bmatrix}$$

$$\|\mathbf{S_1}\| = \sqrt{(B_{\text{CS}})^2 + (P_{\text{CS}})^2}$$

$$\|\mathbf{S_1}\| = \sqrt{(B_{\text{CS1, CP1}} + B_{\text{CS1, CP2}})^2 + (P_{\text{CS1, CP1}} + P_{\text{CS1, CP2})^2}}$$

$$\|\mathbf{S_1}\| = \sqrt{(338 + 162)^2 + (7300 + 3577)^2}$$

$$\|\mathbf{S_1}\| = \sqrt{500^2 + 10877^2} \approx 10,888$$

Scenario 2 (H-EVCMS framework)

$$B_2 = \begin{bmatrix} 126 & 125 \\ 115 & 134 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 2635 & 2542 \\ 2500 & 3200 \end{bmatrix}$$

$$\|\mathbf{S_2}\| = \sqrt{(B_{\text{CS}})^2 + (P_{\text{CS}})^2}$$

$$\|\mathbf{S_2}\| = \sqrt{(B_{\text{CS1}})^2 + (P_{\text{CS1}})^2 + (B_{\text{CS2}})^2 + (P_{\text{CS2}})^2}$$

$$\|\mathbf{S_2}\| = \sqrt{(B_{\text{CS1, CP1}} + B_{\text{CS1, CP2}})^2 + (B_{\text{CS2, CP1}} + B_{\text{CS2, CP2}})^2} + \sqrt{(P_{\text{CS1, CP1}} + P_{\text{CS1, CP2}})^2 + (P_{\text{CS2, CP1}} + P_{\text{CS2, CP2}})^2}$$

$$\|\mathbf{S_2}\| = \sqrt{(126+115)^2 + (125+134)^2} + \sqrt{(2635+2500)^2 + (2542+3200)^2}$$

$$\|\mathbf{S_2}\| = \sqrt{241^2 + +259^2 + 5135^2 + 5742^2} \approx 7,711$$

The comparison of spread measures indicates that Scenario 2, with multiple CSs, achieves a more balanced distribution of bookings and power consumption across stations, demonstrating improved efficiency and resource utilization compared to the concentrated nature of Scenario 1. This outcome suggests that a hybrid approach, as implemented in Scenario 2, may lead to a more optimal allocation of charging resources, reducing congestion and enhancing overall system performance.The practical implementation of the H-EVCMS framework in a metropolitan area highlights its effectiveness in optimizing EV charging operations. By distributing load across multiple CSs, shifting bookings to off-peak periods, and balancing loads, the framework supports sustainable transportation trends through improved efficiency, reduced operational costs, and better resource utilization, contributing to a more sustainable and scalable EV charging infrastructure.

However, scalability can present challenges in large-scale EV charging deployments, potentially leading to performance bottlenecks and degraded user experiences. To mitigate these concerns, the H-EVCMS framework integrates load balancing algorithms and resource management techniques to efficiently distribute workloads across multiple CSs, ensuring optimal performance as the system scales to meet increasing demand. Moreover, infrastructure variability, characterized by differences in CS types, power capacities, and communication protocols, may hinder interoperability and system integration. Nevertheless, the H-EVCMS framework tackles this challenge by primarily leveraging the OCPP for communication, ensuring compatibility with a diverse range of CSs. This standardized protocol promotes seamless integration and interoperability, enabling the framework to interact effectively with various infrastructure components while maintaining reliability and efficiency. Despite these advancements, regulatory compliance remains an important consideration for H-EVCMS implementation. Future endeavours will concentrate on aligning with regulatory requirements through comprehensive assessment and collaborative efforts, ensuring adherence to relevant standards and regulations.

5.5.2 Result Analysis

In this section, the performance of the EVCMS and H-EVCMS systems is evaluated using booking data generated using a custom script and the implemented systems in a controlled environment. The custom script generated the booking time, duration, and power requirements where as the CS allocation and cost determination in (Table 5.2) and (Table 5.3) is done by the algorithms implemented in EVCMS and H-EVCMS systems respectively. This data is designed to simulate real-world booking scenarios.

Table 5.2 CS Data generated by EVCMS System

BID	Dh	Pow	PS	AS	AC	OC	CS
1	120	35.2	0	1	17.8	17.8	CS2
2	60	17.6	1	0	12	9.75	CS1
3	30	8.8	0	0	5.35	5.35	CS1
4	120	35.2	1	0	20.8	18.55	CS1
5	30	8.8	1	0	7.6	5.35	CS1
6	105	30.8	1	1	17.1	15.6	CS2
:							
496	30	8.8	1	1	6.1	4.6	CS1
497	75	22	1	0	14.2	11.95	CS2
498	120	35.2	1	0	20.8	18.55	CS1
499	75	22	0	1	11.2	11.2	CS1
500	75	22	0	0	11.95	11.95	CS2

(Table 5.2) presents a detailed overview of CS (CS) data generated by the EVCMS system. This comprehensive breakdown includes essential parameters such as Booking ID (BID), charging duration (Dh), power consumption (Pow), peak charging status (PS), CS availability status (AS), actual charging cost (AC), optimized cost after applying optimization (OC), and the corresponding CS.

(Table 5.3) displays a detailed overview of CS data generated by the H-EVCMS system. All parameters in this table are identical to those in (Table 5.2). The (Table 5.3) encapsulates CS data while considering both CS connected to a central server. This connectivity influences the availability status of CS, subsequently impacting actual cost (AC) and optimized cost (OC) calculations.

To generate these dataset, each booking underwent a random time slot assignment. Subsequently, these time slots were classified as either Peak on (occurring between 8-10 AM and 4-8 PM) or Peak off, thereby determining the PS values. AS values were derived based on the availability in the CS during the designated time slots. Additionally, both Dh and Pow for each charging session were randomly assigned. Furthermore, the AC was computed based

Table 5.3 CS Data generated by H-EVCMS System

TID	Dh	Pow	PS	AS	AC	OC	CS
1	120	35.2	0	1	17.8	17.8	CS2
2	60	17.6	1	1	10.5	9	CS1
3	30	8.8	0	0	5.35	5.35	CS2
4	120	35.2	1	1	19.3	17.8	CS1
5	30	8.8	1	1	6.1	4.6	CS1
6	105	30.8	1	1	17.1	15.6	CS1
:							
496	30	8.8	1	1	6.1	4.6	CS2
497	75	22	1	1	12.7	11.2	CS1
498	120	35.2	1	0	20.8	18.55	CS2
499	75	22	0	1	11.2	11.2	CS2
500	75	22	0	1	11.2	11.2	CS1

on the assigned duration and power consumption. To illustrate optimized cost scenarios, a subset of bookings was strategically shifted to Peak off slots. For calculating the cost, EV charging price formula given in equation(12) was used and following assumptions were made:

- **Ec**: It is the energy cost, set at 50 cents per kWh.
- **Pmax**: It is the maximum power output, set to 22 kW per day.
- **Ceff**: It is the charging efficiency, assumed to be 80% (0.8).
- **PDC**: It is the peak demand charges, set at 300 cents.
- **SCcs**: It is the service charge for the CS, set at 20 cents.

The above assumed charging efficiency, energy cost, and service charges are constant and reflective of typical values similar to those observed in real-world scenarios.A fixed percentage of users prefer specific time slots, introducing a bias to demonstrate system capabilities.The biased ratio in the data splitting is intentionally introduced to demonstrate the effectiveness of the framework under controlled conditions. This allows us to clearly show the system's potential performance improvements and validate its robustness. In real-world applications, similar patterns and results are expected, making the study's findings relevant and applicable.

The main goal of the H-EVCMS model was to handle more CSs efficiently as compared to EVCMS model. (Fig.5.10) illustrates the comparison of availability between the EVCMS

and H-EVCMS approaches. The x-axis corresponds to the time of day in hours, while the y-axis represents the number of available slots. EVCMS has uneven availability, with CS1 having more slots than CS2.In contrast, H-EVCMS, with centralized control provides better availability that is evenly spread between both CS1 and CS2. The evaluation shows that the EVCMS model has fewer available slots compared to H-EVCMS.

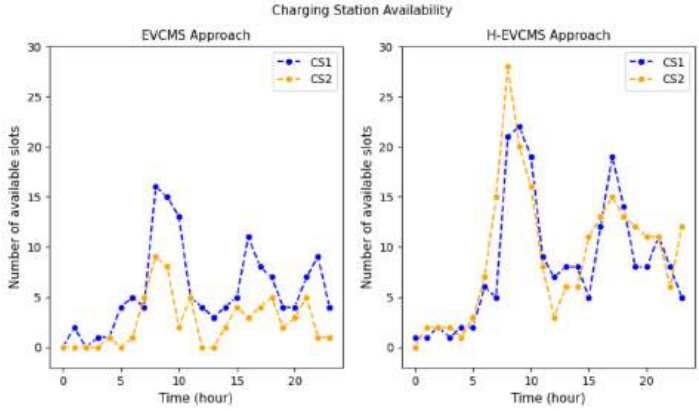


Fig. 5.10 CS availability among EVCMS and H-EVCMS approach

In (Fig.5.11), the comparison of EV charging demand between the EVCMS and H-EVCMS approaches is depicted. The x-axis denotes the time of day in hours, while the y-axis indicates the count of EVs. The evaluation reveals a higher demand for EV charging in CS1 than CS2 in the EVCMS approach. Conversely, in H-EVCMS, the demand for EV charging is evenly distributed between CS1 and CS2. This highlights the superior performance of H-EVCMS over the EVCMS approach in managing EV charging demand.

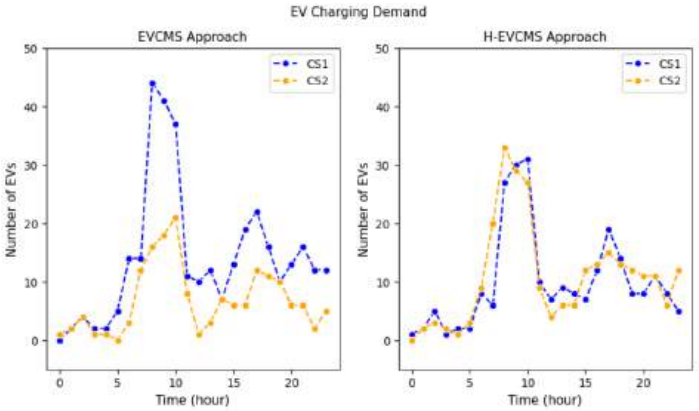


Fig. 5.11 EV charging demand among EVCMS and H-EVCMS approach

To provide a detailed view of EV charging demand during peak periods (Peak On/Off), (Fig. 5.12) presents a comparison between the EVCMS and H-EVCMS approaches during

peak periods. The x-axis represents the time of day in hours, and the y-axis illustrates the count of EV charging instances during peak periods. In the EVCMS approach, CS1 exhibits higher demand during the peak On period, while CS2 maintains a balanced demand across both periods. This results in an uneven load distribution between the two CSs. In contrast, H-EVCMS achieves a more balanced EV charging demand in both peak periods, showcasing superior resource utilization compared to the EVCMS approach.

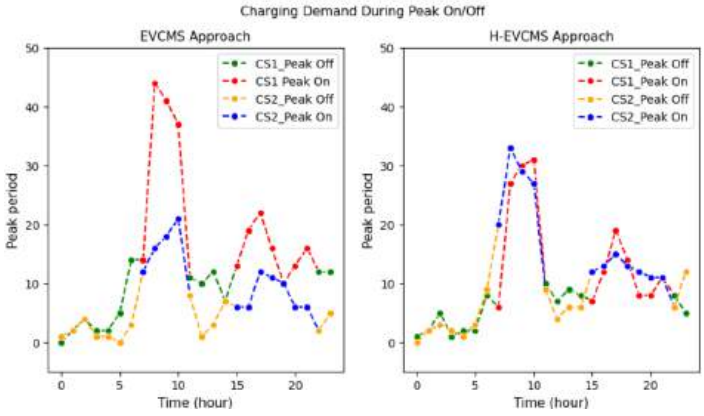


Fig. 5.12 EV Charging demand during Peak On/Off period among EVCMS and H-EVCMS approach

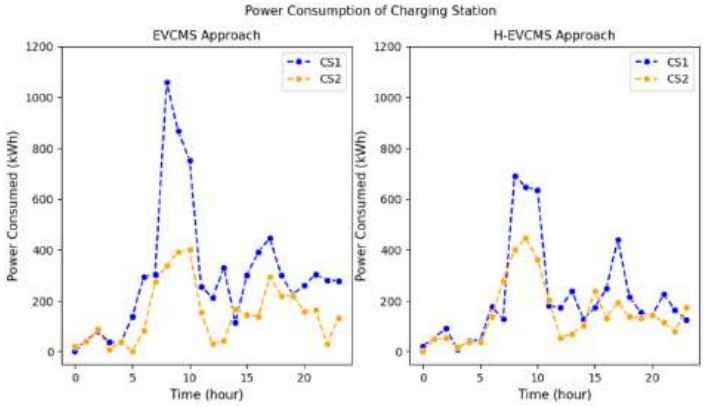


Fig. 5.13 Power consumption of CS among EVCMS and H-EVCMS approach

(Fig.5.13) displays the power consumption of CS in both the EVCMS and H-EVCMS approaches. The x-axis denotes the time of day in hours, and the y-axis represents the power consumed by each CS in kilo-Watt-hours (kWh). In the EVCMS approach, where charging demand is higher in CS1 than CS2, the power consumption mirrors this pattern, with CS1 using more power. In contrast, H-EVCMS, benefiting from a more balanced EV charging demand, showcases an equitable distribution of power consumption. This indicates

that H-EVCMS avoids putting excessive strain on one CS by ensuring a more even power consumption compared to EVCMS.

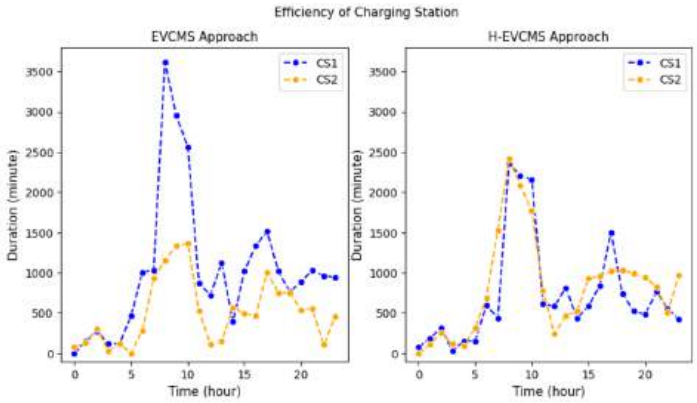


Fig. 5.14 Efficiency of CS among EVCMS and H-EVCMS approach

(Fig.5.14) showcases the charging efficiency of CS in both the EVCMS and H-EVCMS approaches. This efficiency is assessed by measuring the total duration required to charge all EVs within each CS. The x-axis represents the time of day in hours, while the y-axis depicts the duration in minutes taken by each CS to charge its EVs. In the EVCMS approach, CS1 exhibits a longer duration for charging each EV compared to CS2. Conversely, in the H-EVCMS approach, each CS demonstrates a shorter time to charge all its EVs, attributed to a more balanced distribution of the charging load. This underscores the enhanced efficiency of H-EVCMS in minimizing the time needed to charge all EVs within a specific slot.

H-EVCMS demonstrated superior scalability and efficiency in managing multiple CSs. It consistently outperformed EVCMS in providing balanced availability, evenly distributing EV charging demand, optimizing resource utilization during peak periods, maintaining even power consumption across CSs, and achieving enhanced charging efficiency. These findings highlight the effectiveness of H-EVCMS in addressing the challenges associated with CS management shown in EVCMS model, making it a more robust and efficient solution for handling increasing demands in the EV charging infrastructure.

5.5.3 Comparative Analysis

When assessing the effectiveness of the H-EVCMS compared to existing systems, careful consideration was given to the technology adopted, along with the factors considered for optimization, charging cost reduction, and load balancing within the EV charging infrastructure. The H-EVCMS was evaluated against current state-of-the-art systems referenced as [150], [151], and [152], to assess its advancements in the field.

In previous research works, the authors respectively focused on the development of a stochastic model based on the dynamic impact of load congestion [150], an EV charging scheduling algorithm [151], and a prioritization ranking algorithm coupled with a blockchain-based incentive system [152]. These contributions have been pivotal in advancing the optimization of EV charging operations. The stochastic model in [150] aimed to optimize EV user behaviours based on road congestion and battery usage, while the EV charging scheduling algorithm in [151] focused on optimizing EV charging behaviours based on grid load. Additionally, the prioritization ranking algorithm and blockchain-based incentive system in [152] optimized EV charging behaviours based on renewable energy utilization.

Building upon these developments, the H-EVCMS framework was designed for efficient charging operations in multi-CS environments. Notably, while the H-EVCMS framework emphasizes optimizing EV charging prices and schedules based on the charging infrastructure itself, the methodologies discussed in [150], [151], and [152] also promote optimization based on EV user behaviours and charging patterns. In the context of charging cost reduction, strategies such as improving charging infrastructure utilization and scheduling charging during peak, off-peak, or idle timings were explored in [150], [151], and [152]. Building on these concepts, the H-EVCMS framework reduces charging costs by incentivize charging during off-peak hours and optimizing charging infrastructure utilization during peak and off-peak periods. Leveraging insights from [150], where load balancing is based on vehicle availability and charging time, from [151], where it relies on regional load profiles of the grid, and from [152], where it considers transmission loads of distributing grids, the H-EVCMS framework incorporates a nuanced approach to load balancing by calculating the weight of each individual CS.

H-EVCMS is designed to efficiently allocate and schedule charging sessions for EVs. Traditional EVCMS methods often rely on static power capacity constraints, which can limit their effectiveness in balancing the load across CS and responding to real-time grid conditions. The integration of advanced monitoring and analysis techniques can significantly enhance the performance and reliability of these systems. So H-EVCMS in the context of DTR and Time-by-time analysis was also assessed.

Dynamic Thermal Rating (DTR) allows for the real-time monitoring of transmission line conditions. DTR systems, highlighted in references [153],[154],[155], have demonstrated a remarkable increase in line capacity by 30-50%, presenting opportunities to enhance line ratings and improve power grid reliability. H-EVCMS allocates EVs to CS based on their weights, aiming for balanced load distribution. However, H-EVCMS operates within certain power capacity constraints.With DTR integration, H-EVCMS gains the capability to dynamically adjust its charging strategies in response to real-time line conditions. Specifically,

during off-peak hours when transmission lines are underutilized, DTR can dynamically increase line capacity. This enhancement allows H-EVCMS to allocate more EVs to CS simultaneously without risking grid overload. By leveraging DTR, H-EVCMS can optimize EV charging schedules with greater flexibility, maximizing charging efficiency while ensuring grid stability.

Time-by-time analysis, as showcased in references [156], [157], and [158], is crucial for continuously optimizing resource utilization by monitoring data at frequent intervals to effectively adapt to changing conditions. While H-EVCMS already accounts for user preferences, peak/off-peak periods, and load distribution across CS, integrating time-by-time analysis enhances its ability to dynamically respond to real-time grid conditions. In the event of unexpected fluctuations in electricity demand, this integration enables H-EVCMS to promptly adjust charging schedules, optimizing grid utilization and ensuring stability.

Although DTR and time-by-time analysis are not currently integrated into the H-EVCMS framework, their potential benefits have been assessed. This evaluation highlights how incorporating these techniques could enhance system performance. DTR could enable H-EVCMS to adjust charging strategies based on real-time transmission line conditions, allowing for greater load flexibility and improved grid reliability. Similarly, time-by-time analysis would support dynamic adaptation to fluctuating demand by enabling fine-grained monitoring and adjustment of charging schedules. Together, these techniques suggest future enhancements that could significantly improve the responsiveness, efficiency, and stability of H-EVCMS when integrated.

5.6 Chapter Summary

The H-EVCMS framework is an enhancement to the existing EVCMS framework. The H-EVCMS framework is a cloud ready charging management system for managing multiple CS. This proposed framework enables remote charging management through the implementation of the OCPP communication protocol. Notably, the OCPP has been enhanced within this framework to improve the security aspects of session management. An additional objective of the H-EVCMS framework achieved is to minimize charging costs and evenly distribute the charging demand across multiple CS. This is achieved by assigning weights to each CS and selecting the one with the least weight, thereby optimizing resource utilization. Th comparative analysis demonstrated that H-EVCMS framework outperforms EVCMS, highlighting better resource utilization of CS. This outcome underscores the framework's ability to enhance the robustness and optimization of EV infrastructure, showcasing improved resource allocation and overall efficiency in managing CS.

The future scope of the H-EVCMS framework involves a thorough examination of potential cyber threats to EV charging infrastructure. The focus will be on examining cyber threats that target the communication links between charge point and the server used by EV user to connect. Despite the use of the OCPP for secure communication channels, vulnerabilities still exist. Our goal is to further enhance the OCPP version to improve the security of the communication link from charge points to the server. This initiative aims to fortify the existing measures and address potential cyber threats, ensuring a more resilient and secure framework for charge point-to-server interactions.

Chapter 6

Slow DoS attack on H-EVCMS

This chapter analyses the network traffic behaviour of various slow DoS attacks on the EVCMS framework using HTTPS requests. In this chapter, **Section 6.1** provides background on the need to consider DoS attacks in the context of EV charging, explaining the potential risks and impacts of such attacks on the EVCMS framework. **Section 6.2** outlines the experimental methodology, detailing the different types of slow DoS attacks that can target the EVCMS. **Section 6.3** presents the performance evaluation, demonstrating the effective detection of slow DoS attacks on the EVCMS framework through the analysis of network-based parameters, specifically window size and delta time. **Section 6.4** proposes a mitigation strategy designed to prevent slow DoS attacks on the EVCMS framework. Finally, **Section 6.5** provides a chapter summary, assessing the effectiveness and efficiency of the experiment performed on the EVCMS framework to improve its security.

6.1 Background

The widespread adoption of EVs has emphasized the importance of effectively managing charging services. The EVCMS framework plays a crucial role in efficiently overseeing EV charging infrastructure, facilitating optimal resource allocation, grid balancing, cyber security and ensuring seamless user experiences [60]. The system's benefits in enhancing charging infrastructure are evident, but it is essential to acknowledge and address potential vulnerabilities that could make it susceptible to cyber-attacks [28]. Despite various cyber threats, this paper focuses on vulnerabilities related to DoS in EVCMS. Authentication flaws can be exploited by malicious actors aiming to overwhelm the EVCMS framework with unauthorized requests, potentially causing service disruptions [102]. Communication channel vulnerabilities may allow attackers to flood the EVCMS with malicious traffic, rendering it unresponsive to legitimate user requests [159]. Additionally, challenges in

resource allocation, if not addressed, could be manipulated to exhaust EVCMS resources, resulting in a DoS scenario [90].

EVCMS integrated with OCPP is generally regarded as a secure option, but it is crucial to acknowledge that an EVCMS framework can still be vulnerable to various types of attacks[10]. Despite its safety, there is a risk of charging services being disrupted through DoS attacks, which can be initiated directly on the server using HTTPs requests[160]. However, it's worth emphasizing that EVCMS with OCPP remains a more secure choice in comparison to other communication channels that might lack adequate security measures[160]. A typical DoS attack scenario on EVCMS is shown in (Fig.6.1).

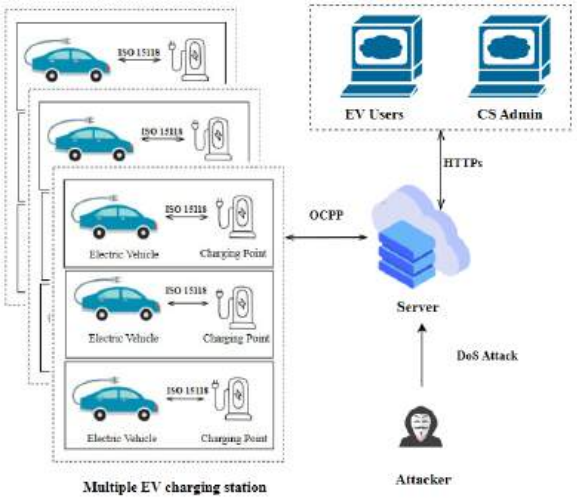


Fig. 6.1 DoS attack scenario on EVCMS

A DoS attack on an EV charging process typically involves an attacker flooding the charging system with an overwhelming amount of traffic or requests by attacking the server through HTTPs request function[161]. A new variant of DoS attacks are termed 'slow DoS attacks'. Unlike traditional DoS attacks that flood systems with lots of traffic, slow DoS attacks quietly drain resources by creating many legitimate connections[162]. This method can make EVCMS unreachable for real users without using significant resources. HTTP servers are a major target for slow DoS attacks. Attackers use lots of legitimate connections from one place to overwhelm the server. Surprisingly, just one slow DoS attack can disable an entire web server, no matter how strong it is. Detecting slow DoS attacks is difficult because they don't result in large traffic spikes and use real connections, blending in with normal traffic.To mitigate these risks effectively, a proactive and comprehensive approach to cyber security is imperative, including the adoption of industry-wide standards[163]. Given that slow DoS attacks represent a critical cyber security threat capable of impacting EVs and the associated charging infrastructure, it is necessary to conduct a thorough analysis of

these various attacks and assess their potential impact. This chapter focuses on analysing the unusual traffic patterns generated by various slow DoS attacks targeting EVCMS.

6.2 Experimental Methodology

This section thoroughly examines Slow DoS attack, focusing on four different types and the stages involved in each. It aims to clarify the exchange of requests and responses between the attacker and the server, which ultimately leads to an attack. By breaking down these processes, the section aims to provide a clear understanding of how these attacks operates.

6.2.1 SlowLoris Attack

SlowLoris is a sophisticated method used to disrupt websites by overwhelming their available connections as shown in (Fig.6.2), making them different from traditional flooding attacks. SlowLoris begins by trying to connect to a website but never completes the process. It sends partial data to initiate a connection and then stops, leaving the connection half-open.To keep the disruption going, SlowLoris sends small bits of data now and then. This prevents the website from closing the connection due to inactivity but doesn't complete the request.SlowLoris fills up the website's capacity to handle connections by keeping its half-open connections active. Surprisingly, this doesn't require much internet speed, making it accessible even with a slow connection.Because all connection slots are taken up, legitimate users can't access the website. They try to connect, but the website is too busy dealing with the fake half-open connections, making it seem like the site is down.

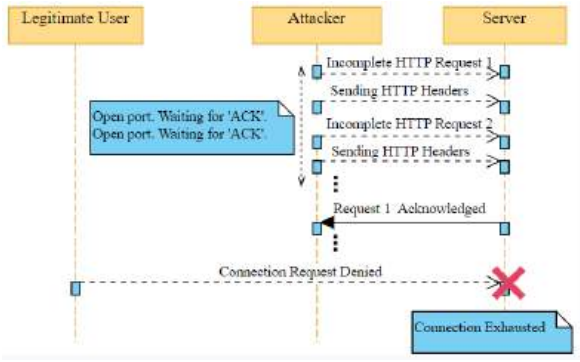


Fig. 6.2 Implementation of SlowLoris Attack

SlowLoris is effective because it quietly disrupts a website's ability to serve legitimate visitors without needing much power or being easily detected.The attack consumes minimal bandwidth, allowing even users with basic internet connections to execute it. SlowLoris

presents a challenge for websites to detect, as it doesn't generate a sudden surge in traffic but rather comprises numerous unresolved connections.

6.2.2 Rudy Attack

The RUDY (Are You Dead Yet?) attack focuses primarily on exploiting POST requests rather than GET requests as shown in (Fig.6.3), specifically targeting web forms commonly submitted through POST. Unlike GET requests, which reveal data in the URL, POST requests allow for a substantial amount of data to be included in the request body. The attacker initiates a POST request to submit data to a form on the target server. This request's headers are transmitted normally, indicating the forthcoming content body. Following the headers, the attacker deliberately sends the actual data at a slow pace, either byte by byte or in small increments. These intervals are meticulously timed to prevent the connection from timing out. This slow transmission is particularly effective with POST requests due to the potential length and complexity of the request body. Most servers are designed to await the complete reception of data before closing the connection. By exploiting this characteristic with POST requests, RUDY effectively ties up a connection slot for an extended duration. This method allows the RUDY attack to keep connections with the target server for a long time. This could make the server run out of resources and make it hard for real users to access the website.

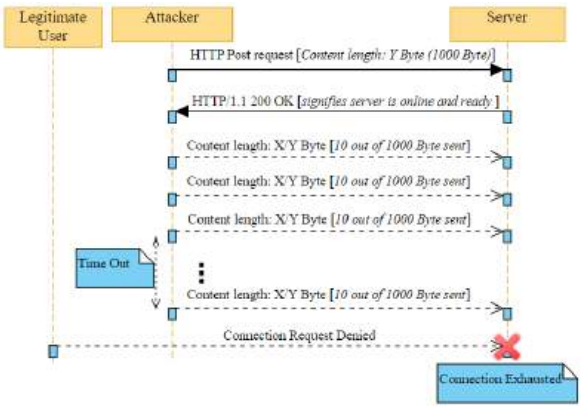


Fig. 6.3 Implementation of Rudy Attack

6.2.3 Slow Read Attack

Slow Read disrupts the flow of data between the website and the attacker by targeting the TCP protocol as shown in (Fig.6.4).Instead of HTTPs GET or POST, in this attack TCP is used. Initially, the attacker establishes a connection with the website in a standard manner,

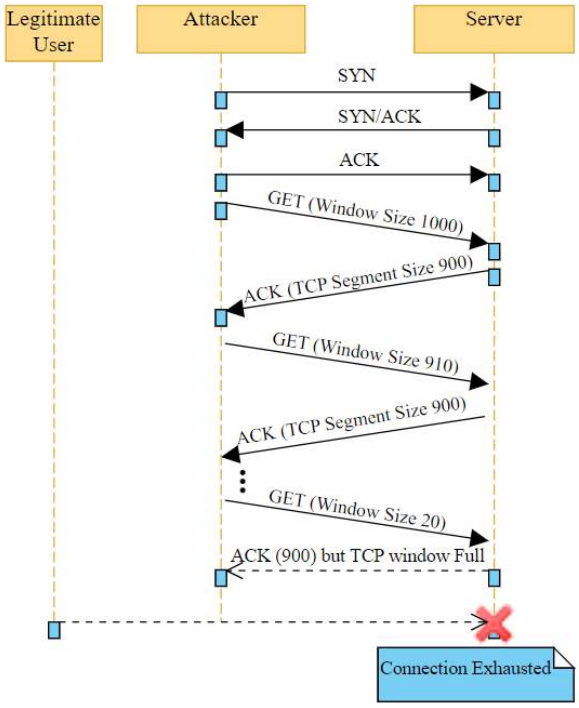


Fig. 6.4 Implementation of Slow Read Attack

similar to initiating a phone call. Once connected, the attacker manipulates the speed of data transmission, deceiving the website into believing it can only handle small amounts of data at a time, even though it's capable of processing more. The attack then slows down the data flow by forcing the website to send information in tiny increments, prolonging each connection. This compels the website to allocate more resources to maintain these connections, diverting attention away from legitimate users. As a result, the website's resources become tied up with these slow connections, impairing its ability to serve new visitors efficiently. This can lead to significant slowdowns or even complete unavailability of the website for other users.

6.2.4 Range Attack

The vulnerability in Apache's handling of the Range Header, leading to a DoS scenario, is commonly referred to as the "Range" or "Apache Killer" attack. This exploit allows attackers to exhaust server resources by flooding it with numerous connection requests, resulting in significant slowdowns or service unavailability as shown in (Fig.6.5).When a range request is initiated, Apache generates multipart responses for each specified range. This process consumes memory, making it possible for attackers to exploit this vulnerability by flooding the server with numerous connection requests, ultimately leading to resource exhaustion and a DoS scenario.

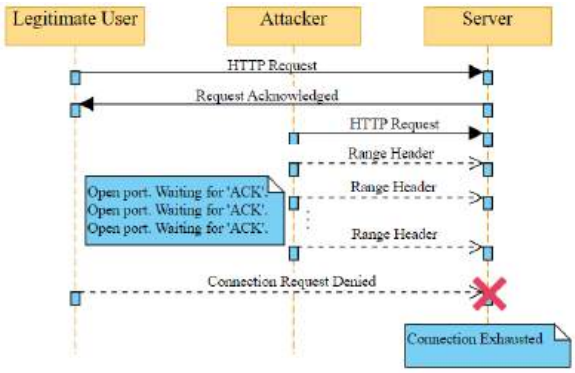


Fig. 6.5 Implementation of Range Attack

If the Apache version is below 2.2.20, upgrading to a version above that threshold will mitigate the risk of this DoS attack. It's worth noting that if one is already utilizing the latest version of the Apache server, such as version 2.4.59, they are not vulnerable to this specific attack, as the necessary security patches and enhancements have been implemented to address this issue.

6.3 Performance Evaluation

The following section provides an overview of the simulation environment utilized, detailing the setup used for conducting experiments. Additionally, it presents the results obtained from these simulations, along with a comprehensive analysis of the data. This section examines into the process of observing and analysing the network traffic behaviour associated with various types of HTTPs-based slow DoS attacks.

6.3.1 Simulation Environment

The experimental setup comprises a Virtual Machine Manager, specifically Windows Hyper-V, hosting two virtual machines: Kali Linux and Ubuntu Server. Within the Kali Linux environment, the slowhttptest tool is installed, utilized for conducting DoS attack. On the Ubuntu Server, an Apache Server hosts the EVCMS web application, along with Tshark, a command-line tool for packet capture and analysis from the Wireshark suite. To simulate and analyse the impact of various types of DoS attacks on EVCMS website, slowhttptest tool was utilized. This tool facilitated conducting SlowLoris, R-U-Dead-Yet (RUDY), and Slow Read attacks on an Apache server hosting the website.

The experimental process begins with a ping test to determine the availability of the target machine on the network. Subsequently, packet capture is initiated using Tshark on the Ubuntu

Server. Once the presence of the target is confirmed, DoS attacks are sequentially executed. To conduct all these attacks, the slowhttptest tool was configured with 1000 connections, an interval of 10 seconds between follow-up headers, a connection rate of 200 per second, the target URL as (https://evcms/index.html), and a timeout of 3 seconds for each connection, while also generating statistics.

a) SlowLoris attack The attack was performed using the following command:

```
slowhttptest -c 1000 -H -g -o SlowLoris_stats -i 10 -r 200 -t GET -u https://
    evcms/index.html -x 24 -p 3
```

The -H flag enabled SlowLoris,and the -o SlowLoris_stats specified the output file prefix for statistics. The -x 24 parameters set the maximum length of follow-up headers.

b) RUDY attack The attack was performed using the following command:

```
slowhttptest -c 1000 -R -g -o rudy_stats -i 10 -r 200 -t POST -u https://evcms/
    index.html -p 3
```

The -R flag enabled RUDY, and the -o rudy_stats specified the output file prefix for statistics. The HTTPs method was set to POST (-t POST) because RUDY typically used POST requests to send headers slowly.

c) Slow Read attack The attack was performed using the following command:

```
slowhttptest -c 1000 -B -g -o slowread_stats -i 10 -r 200 -t GET -u http://evcms
    /index.html -p 3
```

The -B flag enabled Slow Read,and the -o slowread_stats specified the output file prefix for statistics.

Upon completion of each DoS attack, the Tshark instance is halted, and the resulting pcap file is saved for further analysis. The final stage involves analysing the captured packets to determine findings appropriate to the ability and impact of the conducted DoS attacks. Subsequently, offline analysis of the captured data was conducted to gain insights into the distinct characteristics and behaviour of the different types of slow DoS attack traffic.

6.3.2 Result Analysis

The experiment involved collecting attack traffic data from the test bed to identify network parameters useful in detecting slow DoS attacks, particularly SlowLoris, RUDY, and slow read. It's worth noting that the Range DoS attack was excluded due to mitigation in the latest

Apache version. Examining the results involves analysing packet traffic patterns using key metrics such as HTTPs requests and responses, TCP streams, packets sent, bytes transferred, and average duration in seconds. Further detailed analysis is conducted by examining packet traffic patterns, coupled with an in-depth analysis of window size and intervals between packet arrivals (delta time). These metrics provide insights into network behaviour, helping to detect abnormal traffic patterns indicative of slow DoS attacks. This is accomplished by comparing these observations with typical patterns observed in normal traffic. Mitigation strategies were then implemented, after which the attacks were performed again on the system, and the results were analysed.

Table 6.1 Analysis of Packet Traffic Pattern

Metric	Normal	SlowLoris	RUDY	Slow Read
HTTP Requests	800	1,151	1,114	1,035
HTTP Responses	800	1,039	1,021	1,035
TCP Streams	1,500	2,086	2,051	2,076
Total Packets sent	10,000	26,484	19,677	11,793
Total Bytes Transferred	800,000	2,864,807	2,504,890	1,458,963
Average Duration (s)	15.00	69.69	34.67	90.82

The (Table.6.1) above compares various metrics collected during normal and attack scenarios, specifically focusing on SlowLoris, RUDY, and Slow Read attacks. These metrics help to identify distinguishing characteristics of each attack type by comparing them with a normal scenario.

a) **HTTPs requests and response:** HTTPs requests are the number of requests sent from clients to the server, while HTTPs responses are the replies sent from the server back to the clients. Monitoring the number of HTTPs requests and responses is crucial for detecting abnormal traffic patterns. An unusually high number of requests or a significant discrepancy between requests and responses can indicate the presence of a DoS attack. In SlowLoris, the number of HTTPs requests (1,151) and responses (1,039) is significantly higher than normal, indicating many connections being initiated but not necessarily completed. This discrepancy can signal the SlowLoris attack's tactic of maintaining numerous half-open connections. In rudy, the HTTPs requests (1,114) and responses (1,021) also show an increase, reflecting the attack's method of sending data slowly within HTTPs POST requests. This behaviour results in more requests being sent over time to keep the connections open. In Slow read, the HTTPs requests (1,035) and responses (1,035) are the same, not suggesting any abnormal

pattern in terms of request-response mismatch. However, the attack maintains open connections while data is read slowly, leading to prolonged connection durations.

b) **TCP streams:** It refer to the number of concurrent TCP connections established between clients and the server. A high number of TCP streams can indicate multiple open connections, which is typical in certain types of DoS attacks such as SlowLoris, where connections are kept open for extended periods. In SlowLoris, the number of TCP streams (2,086) is significantly higher than normal, consistent with the attack's strategy of maintaining many open connections. This high number of streams can overwhelm the server's capacity to manage connections. In RUDY, the TCP streams (2,051) are also elevated, reflecting its method of opening multiple connections to send data slowly. This results in a large number of concurrent connections being sustained. In Slow Read, the TCP streams (2,076) are elevated as well, indicating many open connections where data is read very slowly to congest the network. This can lead to resource exhaustion on the server due to the high number of maintained connections.

c) **Total packet sent:** This metric represents the count of all packets transmitted within a connection. High packet counts, especially with low data rates, can signal an attack where many small packets are sent to overwhelm the server, typical in SlowLoris and Slow Read attacks. In SlowLoris, the total packets sent (26,484) is extremely high, indicating frequent transmission of partial headers to keep connections open. This high packet count with minimal data transfer is characteristic of SlowLoris, designed to exhaust server resources. In RUDY, the total packets sent (19,677) is also high but lower than SlowLoris, consistent with its balanced approach of maintaining active connections with frequent requests. This reflects the attack's method of sending small amounts of data slowly over time. In Slow Read, the total packets sent (11,793) is the lowest among the attacks, reflecting its strategy of minimizing data transfer while maintaining open connections. This lower packet count can still lead to significant impact due to the prolonged duration of the connections.

d) **Total Bytes Transferred:** This metric represents the sum of the lengths of all packets transmitted in the connection. It is useful for understanding the volume of data exchanged. In DDoS attacks like RUDY, the byte count per request might be low, but frequent requests can lead to high total transfer. In SlowLoris, the total bytes transferred is 2,864,807. This indicates more frequent packet sending or larger packet sizes compared to Slow Read. In RUDY, the total bytes transferred is 2,504,890. This is also higher than Slow Read, indicating more frequent packet sending or larger packet sizes. In Slow Read, the total bytes transferred is 1,458,963. This is the least data

transfer, aligning with its strategy to minimize data flow while keeping connections open.

e) **Average Duration per Connection:**This is the average time from the first to the last packet in each TCP connection. It is important because it helps identify long-lasting connections, which are typical in SlowLoris attacks where connections are kept open by sending partial headers or keep-alive headers at regular intervals. In SlowLoris, the average duration is 69.69 seconds. This shows a relatively long duration, aligning with its strategy to maintain open connections with minimal data transfer. In RUDY, the average duration is 34.67 seconds. This is shorter compared to the others, possibly due to its mechanism of slowly sending data within HTTPs POST requests. In Slow Read, the average duration is 90.82 seconds. This is the longest average duration, typical for its strategy to read data slowly and keep connections open.

f) **Window size:** The average TCP window size represents the average of the TCP window sizes advertised by the receiver, which dictates how much data the sender can send before receiving an acknowledgment. Anomalously small window sizes might be indicative of certain types of attacks aimed at slowing down the connection.(Table.6.2) represents the average TCP window sizes observed in both client-to-server (C2S) and server-to-client (S2C) communication channels under normal and attack conditions, including SlowLoris, RUDY, and Slow Read attacks. The "Total Average" column provides an overall average of the window sizes for each scenario.

Table 6.2 Analysis of TCP Window Size

Metric	Average TCP Window Size (in Bytes)		
	Client to Server	Server to Client	Total
Normal	8000	8000	8000
SlowLoris	14759.81	6291.75	13,000.67
Rudy	15906.11	8229.63	15,081.56
Slow Read	849.80	12892.43	6,538.97

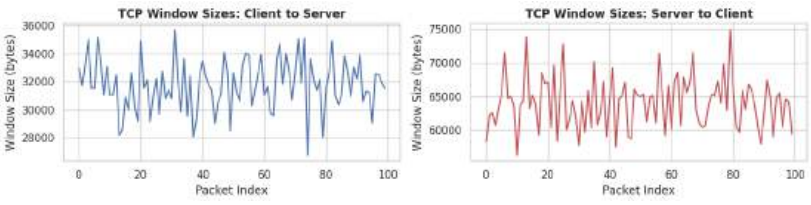


Fig. 6.6 TCP window size in normal scenario

In normal Scenario as shown in (Fig.6.6),the average TCP window size is consistent at 8,000 bytes for both C2S and S2C communication, indicating a balanced and expected flow control in a non-attack scenario.

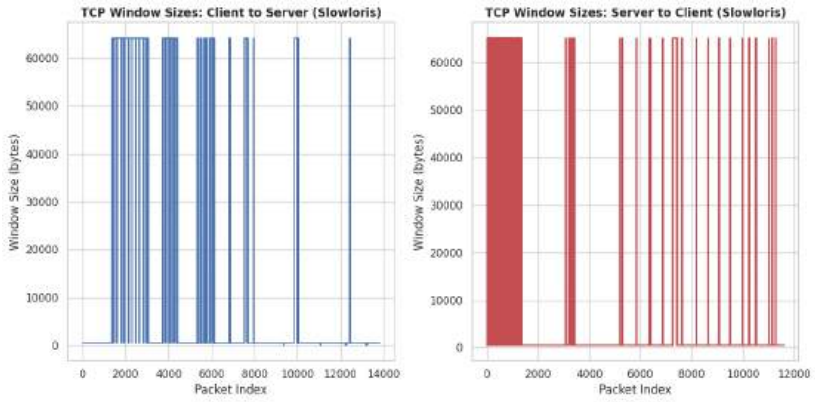


Fig. 6.7 TCP window size in SlowLoris attack

In the SlowLoris scenario as shown in (Fig.6.7), the C2S window size is significantly higher (14,759.81 bytes) compared to the S2C window size (6,291.75 bytes), resulting in a total average of 13,000.67 bytes. This imbalance suggests that the attacker maintains a high buffer for incoming data while potentially throttling outgoing responses to keep connections open.

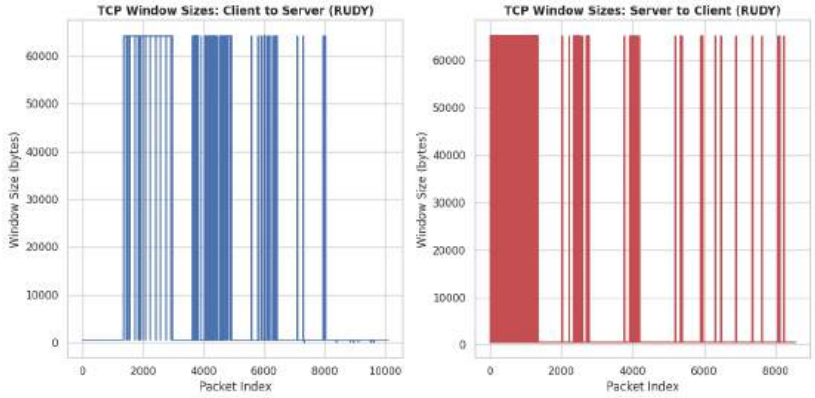


Fig. 6.8 TCP window size in rudy attack

For the RUDY attack as shown in (Fig.6.8), both C2S (15,906.11 bytes) and S2C (8,229.63 bytes) window sizes are elevated, with a total average of 15,081.56 bytes. This indicates that the attack involves a high capacity for receiving data from the client, coupled with substantial server responses, aligning with the attack's method of sending data slowly over multiple connections.

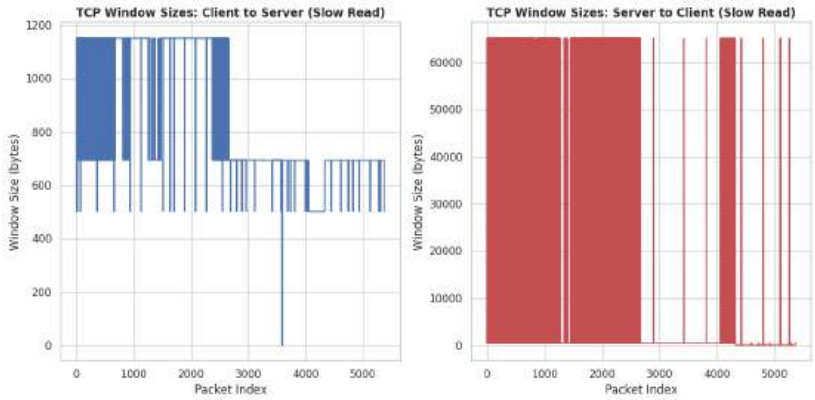


Fig. 6.9 TCP window size in slow read attack

In the case of the Slow Read attack as shown in (Fig.6.9), the C2S window size is exceptionally low (849.80 bytes), whereas the S2C window size is relatively high (12,892.43 bytes), resulting in a total average of 6,538.97 bytes. This disparity is characteristic of the Slow Read attack, where the client deliberately limits its buffer to slow down the data read process from the server, thereby congesting the server resources.

g) Delta Time: This metric measures the average time interval between packets in a TCP stream. Very high delta times can indicate a SlowLoris or Slow Read attack, where the attacker intentionally delays sending or acknowledging packets to keep connections open without transferring significant data. The (Table.6.3) presents the average delta time between packets for both client-to-server (C2S) and server-to-client (S2C) communication channels under normal and attack scenarios, specifically focusing on SlowLoris, RUDY, and Slow Read attacks.

Table 6.3 Analysis of Average Delta time between packets

Metric	Average Delta time between packets (in sec)		
	Client to Server	Server to Client	Total
Normal	0.50	0.50	1.00
SlowLoris	5.4120	0.4812	5.9581
Rudy	3.5675	0.2521	4.0339
Slow Read	16.2818	1.2052	19.4031

In Normal scenario as shown in (Fig.6.10), the average delta time between packets is consistent at 1.00 seconds overall, with both C2S and S2C communication averaging

0.50 seconds. This regular interval indicates normal traffic patterns with efficient data transmission and acknowledgment.

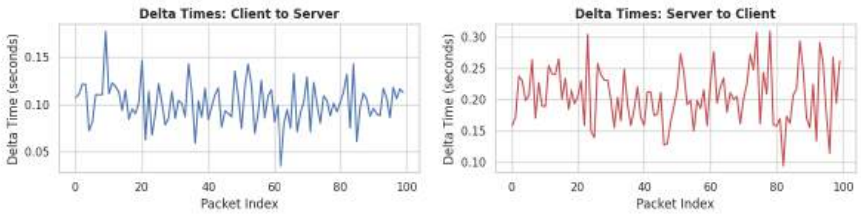


Fig. 6.10 Average Delta time in normal scenario

In SlowLoris attack as shown in (Fig.6.11), The total average delta time (5.9581 seconds) is significantly higher than normal. The C2S delta time (5.4120 seconds) is particularly elevated, while the S2C delta time (0.4812 seconds) remains relatively low. This pattern suggests that the attacker delays sending packets from the client to the server to keep connections open, which is a hallmark of the SlowLoris attack.

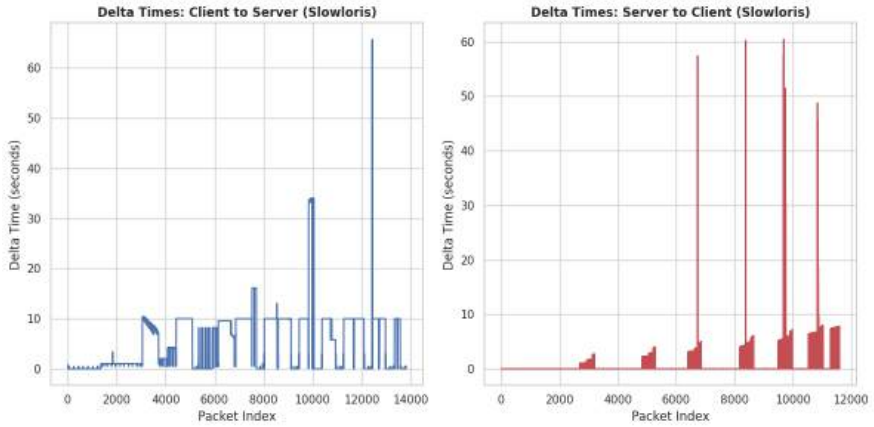


Fig. 6.11 Average Delta time in SlowLoris attack

In RUDY attack as shown in (Fig.6.12), The total average delta time (4.0339 seconds) is also elevated. The C2S delta time (3.5675 seconds) is higher than the normal scenario, while the S2C delta time (0.2521 seconds) is quite low. This indicates that the RUDY attack involves moderate delays in sending data from the client, maintaining active connections without overwhelming the server with immediate responses.

In Slow Read attack as shown in (Fig.6.13), the total average delta time (19.4031 seconds) is the highest among all scenarios, indicating very long intervals between packet transmissions. The C2S delta time (16.2818 seconds) is exceptionally high, while the S2C delta time (1.2052 seconds) is higher than normal but not as pronounced.

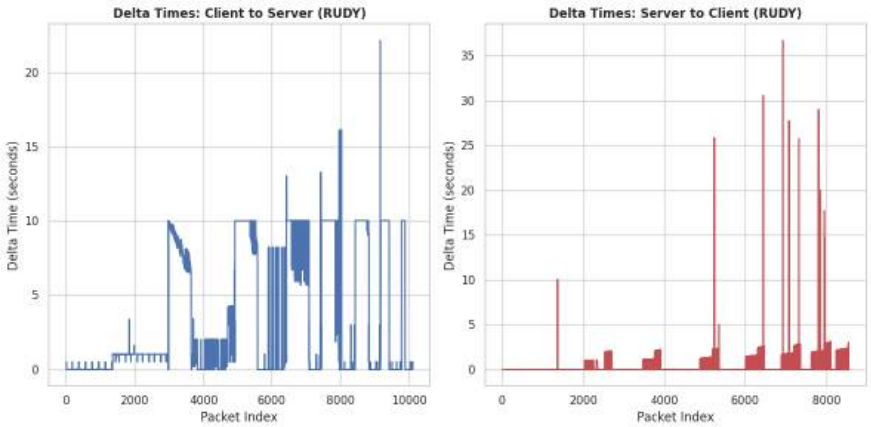


Fig. 6.12 Average Delta time in Rudy attack

This extreme delay in packet intervals is characteristic of the Slow Read attack, where the attacker deliberately reads data very slowly to prolong the connection and exhaust server resources.

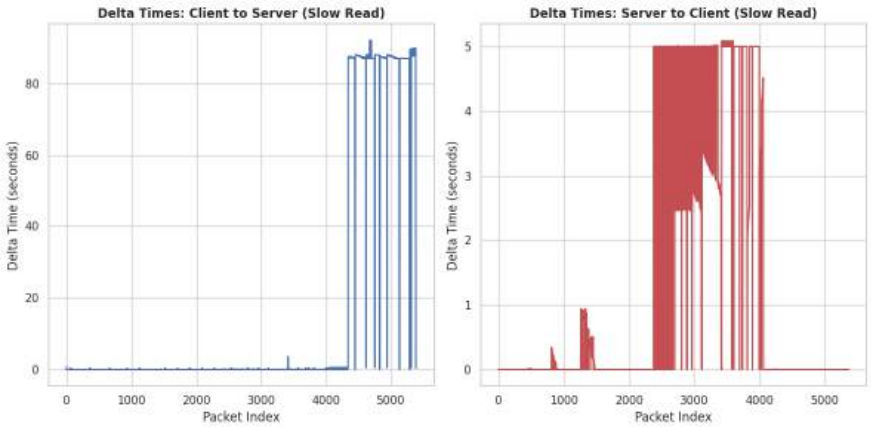


Fig. 6.13 Average Delta time in slow read attack

6.4 Mitigation:Detect and Stop Slow DoS attacks

To effectively detect and mitigate DoS attacks, a structured approach involving the configuration of Suricata and iptables was employed. To tailor Suricata to specific requirements, custom detection rules were created. These rules were designed to identify SlowLoris, RUDY, and Slow Read attacks by examining HTTPs request patterns. The rules were crafted to trigger alerts when patterns indicative of these DoS attacks were detected, and they included instructions to block the offending IP addresses dynamically.

To detect and prevent SlowLoris, RUDY, and Slow Read attacks, the following custom rules have been created:

a) **SlowLoris Detection and Prevention:** This rule monitors HTTPs traffic directed towards the home network. It identifies established connections where the traffic is flowing to the server and includes a "GET" request in the HTTP header. If it detects 100 such requests from the same source within 10 seconds, it triggers an alert indicating a possible SlowLoris attack. This activity is classified as an attempted denial-of-service attack and assigns it a unique identifier (sid:1000001). Additionally, the offending source IP address is automatically blocked for 5 minutes.

```
alert https any any -> any any (
msg:"Possible SlowLoris Attack";
flow:established,to_server;
content:"GET"; http_header;
threshold:type limit, track by_src, count 100, seconds 10;
classtype:attempted-dos;
sid:1000001;
fwsam:src,5 minutes;)
```

b) **RUDY Detection and Prevention:** Similar to the SlowLoris rule, this rule also monitors HTTPs traffic towards the home network. However, it specifically looks for "POST" requests in the HTTPs header. If it detects 100 such requests from the same source within 10 seconds, it triggers an alert for a possible RUDY attack. This rule is classified under a unique identifier (sid:1000002) and blocks the source IP for 5 minutes to prevent further malicious activity.

```
alert https any any -> any any (
msg:"Possible RUDY Attack";
flow:established,to_server;
content:"POST"; http_header;
threshold:type limit, track by_src, count 100, seconds 10;
classtype:attempted-dos;
sid:1000002;
fwsam:src,5 minutes;)
```

c) **Slow Read Detection and Prevention:** This rule focuses on detecting Slow Read attacks by monitoring established connections with "GET" requests in the HTTPs header. It uses the same threshold as the previous rules, triggering an alert if 100 such requests are detected from the same source within 10 seconds. This rule is assigned a unique identifier (sid:1000003) and similarly blocks the offending IP for 5 minutes.

```
alert https any any -> any any (
msg:"Possible Slow Read Attack";
flow:established,to_server;
```

```
content:"GET"; https_header;
threshold:type limit, track by_src, count 100, seconds 10;
classtype:attempted-dos;
sid:1000003;
fwsam:src,5 minutes;)
```

In addition to creating detection rules, a script was developed to block IP addresses flagged by Suricata. This script dynamically adds offending IPs to the iptables block list, effectively preventing further malicious traffic from those addresses. The (Algorithm 6.1) runs DoS attacks from a Kali machine, monitors Suricata logs for detections of SlowLoris, RUDY, and Slow Read attacks, and verifies blocked IP addresses in iptables rules. It returns logs of detected attacks and blocked IP addresses for further analysis.

Algorithm 6.1 Detecting and Preventing DoS Attacks

Input : Kali machine for performing DoS attacks, Suricata logs, iptables rules

Output : Logs of detected attacks and blocked IPs

1 **Run DoS Attacks:**
Perform DoS attacks (SlowLoris, RUDY, Slow Read) from the Kali machine.

2 **Monitor Suricata Logs:**
foreach *entry in Suricata logs* **do**

3 **if** *DoS attack detected* **then**

4 Log entry: "DoS attack detected from Malicious IP addresses"

5 **Verify Blocked IP Addresses:**
foreach *blocked IP address in iptables rules* **do**

6 **if** *IP address matches detected DoS attack* **then**

7 Log entry: "Malicious IP addresses blocked due to detected DoS attack."

8 **Return Logs:**
Return logs of detected attacks and blocked IP addresses for analysis.

A series of tests were conducted to verify the effectiveness of the setup. DoS attacks were performed from a Kali machine, and Suricata logs were monitored. The iptables rules were checked to confirm that the offending IP addresses were being dynamically added to the block list.

In (Fig.6.14), evidence of one of the conducted test is presented, demonstrating the effectiveness of the system. Suricata logs successfully detected an attack, and iptables effectively added the blocked IP address. This method has proven to be highly effective in mitigating slow DoS attacks.

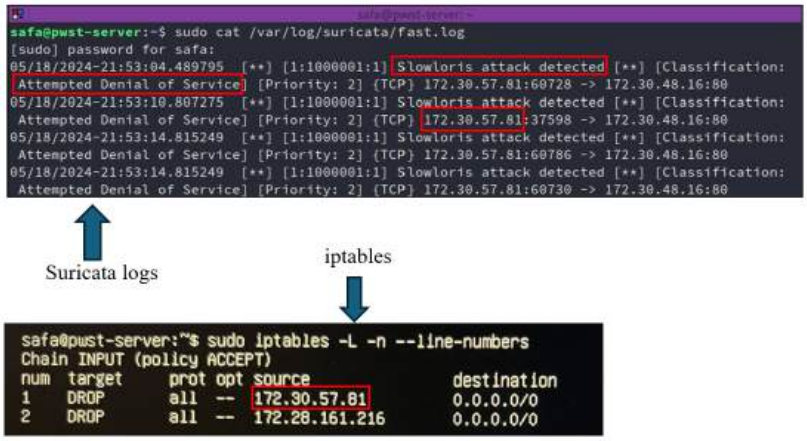


Fig. 6.14 Evidence of DoS attack detected and prevented

6.5 Chapter Summary

The rapid integration of EVs into transportation systems highlights the need for secure EV charging infrastructure. This study addresses vulnerabilities in EVCMS framework to DoS attacks, specifically focusing on slow HTTPs-based DoS attacks. These attacks, including SlowLoris, RUDY, and Slow Read, exploit HTTPs request handling by maintaining prolonged connections and sending data slowly to exhaust server resources. The study underscored the importance of robust detection and mitigation strategies. The combination of custom Suricata rules and iptables blocking mechanisms was highly effective in identifying and mitigating such attacks, ensuring the resilience and security of EV charging services. Detailed analysis of packet traffic patterns, including window sizes and delta times, provided valuable insights into the behaviour of different attack types, aiding in the development of precise detection mechanisms. Among the attacks analysed, the Slow Read attack was particularly severe. It had the longest average duration per connection and the lowest total bytes transferred, meaning it kept connections open for a long time while sending very little data. The Slow Read attack also showed a very high delay between packets, indicating it intentionally slowed down the data reading process to exhaust server resources. This prolonged duration and delay made the Slow Read attack highly disruptive, as it tied up server resources for extended periods, preventing legitimate traffic and leading to a more significant denial of service compared to SlowLoris and RUDY.

Future research should focus on developing anomaly detection model using advance machine learning techniques to improve the detection of network security threats. Machine learning can analyse large amounts of network data to spot subtle signs of slow DoS attacks, making detection more accurate and reducing false alarms. Additionally, creating automated

systems that adjust security measures based on the threat level could further protect the EVCMS framework. These systems could automatically change firewall rules, manage server resources, or redirect traffic to stop attacks without needing human intervention. By exploring these research areas, the security and reliability of EV charging infrastructure can be greatly enhanced, supporting the growth and adoption of EVs worldwide.

Chapter 7

MitM attack on H-EVCMS

This chapter analyses the MitM risks associated with intercepting OCPP client-server communication. In this chapter, **Section 7.1** provides background on the need to consider MitM attacks in the context of EV charging, explaining the potential risks and impacts of such attacks on the EVCMS framework. **Section 7.2** outlines the experimental methodology, analysing the OCPP model and the impact of MitM attacks on OCPP communications. **Section 7.3** presents the performance evaluation, providing a description of the experimental setup used to implement the MitM attack on the OCPP communication protocol used in EVCS, along with an analysis of the traffic intercepted during the MitM attack and the results. Finally, **Section 7.4** provides a chapter summary, assessing the effectiveness and efficiency of the experiment performed on the EVCMS framework to improve its security.

7.1 Background

The adoption of EVs and EVCS has surged due to their environmental benefits, such as reduced GHG emissions and decreased reliance on fossil fuels [3]. EVs also offer economic advantages, like lower operating costs and reduced maintenance compared to ICE vehicles [126]. EVCS are crucial for supporting the widespread use of EVs by providing the necessary infrastructure for charging, enabling longer travel distances and enhancing convenience for EV owners [143]. Despite these benefits, EVCS are vulnerable to various security risks, especially concerning the communication protocol used to manage interactions between CP and ChgSrv [28]. OCPP is a widely adopted standard designed to ensure interoperability between charging stations from different manufacturers and back-end systems [9]. However, OCPP 1.6 and the newer OCPP 2.0.1 have significant security flaws that can be exploited by malicious actors [10]. These vulnerabilities include weak authentication mechanisms and

insufficient data integrity checks, which can lead to unauthorized access and manipulation of the communication channel [104].

OCPP's main versions, 1.6 and 2.0.1, support communication via either SOAP (Simple Object Access Protocol) or JSON (JavaScript Object Notation) [9]. While version 2.0.1 includes enhancements, OCPP 1.6 remains widely utilized and is particularly vulnerable to security risks. OCPP functions by transmitting data bidirectionally over HTTPS or Web Sockets, depending on whether SOAP or JSON is used. Regardless of the format, the data is encrypted with TLS 1.2. However, TLS 1.2 has its own vulnerabilities which can be exploited by attackers[164]. Researchers have identified several vulnerabilities in OCPP 1.6, including the potential for MitM attack, which can compromise the security and reliability of EVCS operations [104],[165], [13]. These attacks exploit the protocol's inadequate handling of data transmitted over Web Socket connections. In cases where encrypted data is intercepted, attackers face significant challenges in deciphering the information, but the interception itself still poses risks such as potential disruptions in service [104].

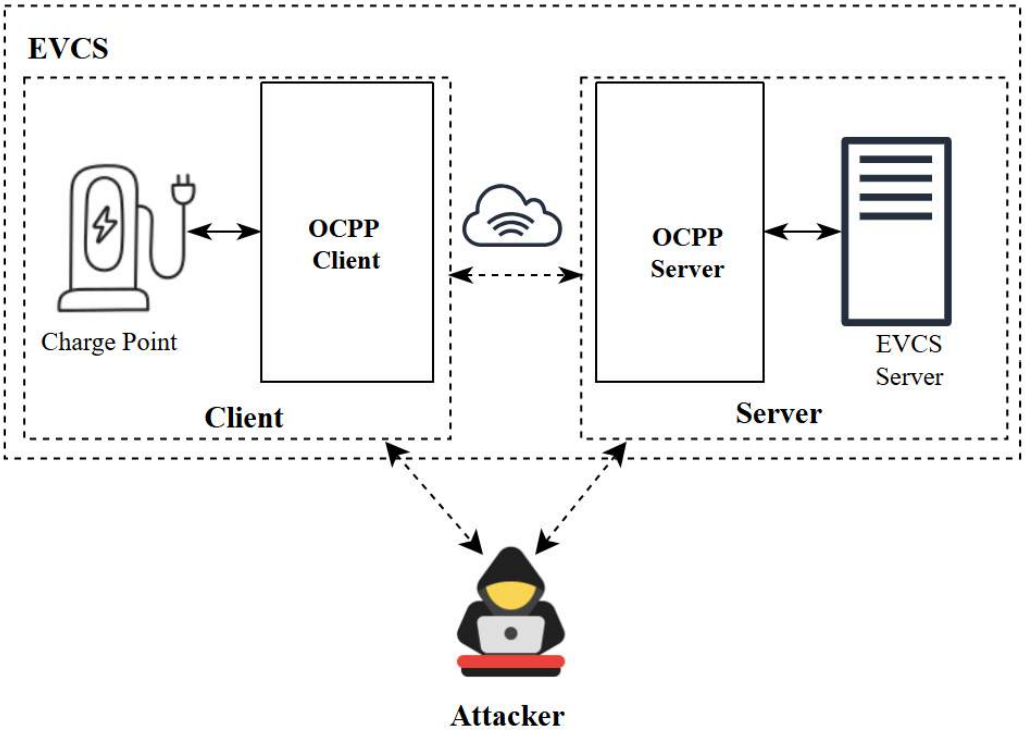


Fig. 7.1 MitM attack in EVCS using OCPP

The consequences of MitM attacks on OCPP-based systems are severe. Attackers can intercept and manipulate the data exchanged between the CP and ChgSrv, leading to fraudulent charging sessions, denial of service, and even broader impacts on the power grid's stability [104]. Such attacks not only threaten the confidentiality, integrity, and availability

of the charging services but also pose significant financial and reputations risks to the stakeholders involved, including EV owners, charging station operators, and energy providers [10],[25]. In light of these security challenges, this paper analyses the risks associated with MitM in EVCS using the OCPP protocol as shown in (Fig.7.1), to demonstrate the potential communication vulnerabilities that exist between OCPP Client and OCPP server. Here OCPP is utilized to send data through web sockets utilizing JSON. Our findings underscore the necessity for enhanced security measures, including prevention of MitM and robust encryption protocols, to safeguard against these threats and ensure the secure operation of EVCS.

7.2 Experimental Methodology

This section analyse OCPP model and the impact of MitM attacks on OCPP communications. It first explains the key communication messages exchanged between the client (CP) and server (ChgSrv). Then, it explores how exploiting these messages in a MitM attack exposes sensitive information, detailing how attackers can misuse each intercepted message.

7.2.1 OCPP Attack Model

This paper aims to protect EVCS from MitM attacks targeting the OCPP. In these attacks, a potential attacker intercepts and possibly alters the communication between the CP and the ChgSrv without either party's knowledge. The attacker does this by tapping into the communication link between the CP and the ChgSrv, which may use wireless technology. Regardless, OCPP relies on bidirectional data transmission over this link, using HTTPs or Web sockets with either SOAP or JSON formats. In our experiment, we have utilized OCPP using Web sockets with JSON. However, it is crucial to note that, in OCPP version 1.6, the information that is sent between client and server, or vice versa, is currently transmitted in encrypted form using TLS 1.2. The type of encryption used ensures that even the "change cipher spec" message is encrypted. This basic level of encryption provided by OCPP protects the data from being intercepted, modified, or disrupted by attackers during transmission. However, if a MitM attack is successfully executed, the attacker can intercept the encrypted messages. While these messages are encrypted, sophisticated attacks could potentially decrypt them. Therefore, it is essential to implement measures to prevent MitM attacks in the first place.

To illustrate the information that we aim to protect, we must first examine the communication between the OCPP Client (Charge Point) and the OCPP Server (EVCS Charge Server)

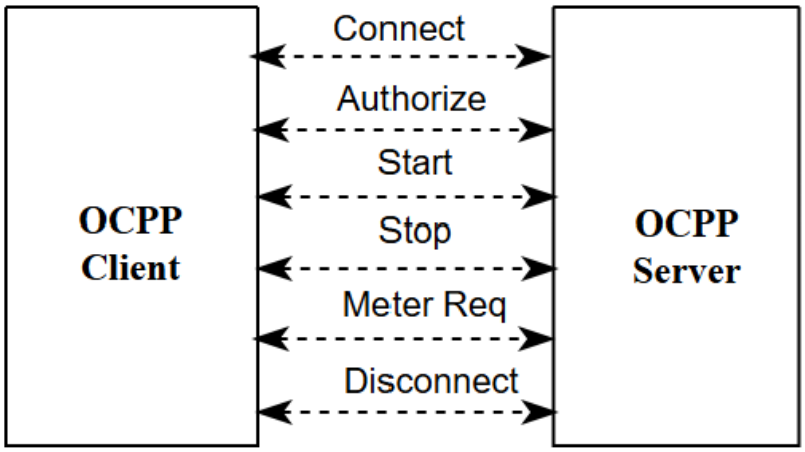


Fig. 7.2 OCPP Client and Server Communication in EVCS

in an EVCS as defined by the OCPP specification [139]. The communication between the client and server involves various types of messages as shown in (Fig.7.2), to manage the charging process and ensure proper functioning.

Initially, the connection process is initiated by the CP to establish communication with the ChgSrv. This begins with a Boot Notification, where the CP sends a message to the ChgSrv detailing the CP's status, vendor, model, firmware version, and readiness to operate. The ChgSrv responds with a confirmation to accept or reject the connection based on the provided information. Before any charging session can commence, the user must be authorized. The CP sends an Authorize Request to the ChgSrv, including the user's identification information, such as user ID. The ChgSrv processes this request and responds with an Authorize Response indicating whether the user is permitted to start a charging session. Upon authorization, the Start Transaction process begins. The CP sends a Start Transaction Request to the ChgSrv, detailing the user's identifier, the charging point, and the session start timestamp. The ChgSrv acknowledges this request with a Start Transaction Response, providing a unique transaction ID to track the session.

To end the charging session, a Stop Transaction Request is sent by the CP to the ChgSrv, including the transaction ID, final meter reading, and session end timestamp. The ChgSrv confirms the transaction's end with a Stop Transaction Response. During the charging session, the CP periodically sends Meter Value Requests to the ChgSrv, reporting the energy consumed with the current meter reading and timestamp. The ChgSrv processes and stores this information for future reference or billing. Finally, to close the communication link, a Disconnect Notification is sent by the CP to the ChgSrv, indicating the intention to

terminate the connection. The ChgSrv acknowledges this notification and safely closes the communication session. Understanding these messages is crucial for ensuring their security, as each involves the transmission of potentially sensitive data that must be protected from interception, alteration, and disruption.

7.2.2 MitM Attack Analysis on OCPP

In a MitM attack, an attacker intercepts and possibly alters the communication between the CP and the ChgSrv. Below, we analyse the data transferred in each message and discuss how an attacker could maliciously use this information if intercepted.

- **Connect (Boot Notification)**
  As shown below, the actual data transformed from CP to ChgSrv in Connect request.
  Vulnerable Data: Vendor, model, firmware version, current status of the CP.
  Malicious Use: An attacker could gather detailed information about the hardware and software used by the CP, which can be used for targeted attacks. By understanding the specifics, the attacker might exploit known vulnerabilities in the firmware or hardware.

```
"chargePointVendor":"AVT-Company",
"chargePointModel":"AVT-Express",
"chargePointSerialNum":"avt.001.13.1",
"chargeBoxSerialNum":"avt.001.13.1.01",
"firmwareVersion":"0.9.87",
"iccid": "",
"imsi": "",
"meterType": "AVT NQC-ACDC",
"meterSerialNumber": "avt.001.13.1.01"
```

- **Connect(Boot Notification Response)**
  Vulnerable Data: Confirmation of CP's connection status.
  Malicious Use: If an attacker intercepts and blocks this message, the CP might not be able to establish a connection with the ChgSrv, rendering the charging station non-functional. Alternatively, the attacker could send a false response, misleading the CP about its connection status.

```
"status": "Accepted",
"interval": 100,
"currentTime":"2023-09-10T10:36:18.475Z"
```

- **Authorize Request**
  Vulnerable Data: User identification information (user ID).
  Malicious Use: An attacker could capture the user's identification details and use them to impersonate the user, potentially gaining unauthorized access to charging services or user accounts.

```
"idTag":"TAG001"
```

- **Authorize Response**
  Vulnerable Data: Authorization status.
  Malicious Use: Intercepting this message allows an attacker to alter the authorization status. They could deny service to legitimate users by sending a false rejection or allow unauthorized users by sending a false acceptance.

```
"status":"Accepted"
```

- **Start Transaction Request**
  Vulnerable Data: User identifier, charging point ID, timestamp, transaction details.
  Malicious Use: By capturing this information, an attacker could start unauthorized charging sessions. They might also manipulate transaction details, such as altering timestamps or the amount of energy to be delivered, leading to incorrect billing or fraudulent use of services.

```
"connectorId":1,
"idTag":"TAG001",
"timestamp":"2023-09-10T10:38:57.882Z",
"meterStart":0,
"reservationId":16943302990312306
```

- **Start Transaction Response**
  Vulnerable Data: Unique transaction ID, acknowledgment.
  Malicious Use: An attacker could intercept and modify this message to provide a different transaction ID, causing confusion in tracking the session. This might also be used to interrupt or hijack ongoing transactions.

```
"transactionId": 1,
"idTagInfo":{
"status": "Accepted"}
```

- **Meter Value Request**
  Vulnerable Data: Current meter reading, timestamp. **Malicious Use:** An attacker could alter meter readings, causing inaccurate billing or energy tracking. They might also flood the ChgSrv with false meter readings, potentially disrupting the system's ability to manage energy usage.

```
"connectorId": 1,
"transactionId": 1,
"meterValue": [{
"timestamp": "2023-09-10T10:38:58.896Z",
"sampledValue":[ {
"value": "0.0049"}
]}]
```

- **Meter Value Response**
  Vulnerable Data: Acknowledgment of meter value.
  Malicious Use: By blocking or altering this message, an attacker could disrupt the communication flow, causing the CP to resend meter values unnecessarily, which could lead to network congestion or data inconsistencies.

```
"status":"Accepted"
```

- **Stop Transaction Request**
  Vulnerable Data: Transaction ID, final meter reading, timestamp.
  Malicious Use: An attacker could manipulate this message to prematurely stop a transaction or alter the final meter reading, resulting in incorrect billing. They could also block this message to prevent the session from ending, causing prolonged charging without proper authorization.

```
"transactionId": 1,
"idTag": "TAG001",
"timestamp":"2023-09-10T10:42:55.021Z",
"meterStop": 1
```

- **Stop Transaction Response**
  Vulnerable Data: Acknowledgment of transaction termination.
  Malicious Use: By intercepting and altering this message, an attacker could make the CP believe the transaction has ended when it has not, or vice versa. This could lead to incorrect logging of sessions and potential disputes between users and operators.

```
"idTagInfo": {
"status": "Accepted"}
```

- **Disconnect Notification**
  Vulnerable Data: Notification of intent to disconnect.
  Malicious Use: An attacker could block or alter this message to keep the CP connected longer than necessary or prematurely disconnect it. This could be used to deny service to legitimate users or disrupt the normal operation of the charging station.

```
"idTagInfo": {
"status": "Disconnect"}
```

- **Disconnect Response**
  Vulnerable Data: Acknowledgment of disconnection.
  Malicious Use: By intercepting and altering this message, an attacker could disrupt the proper disconnection process, potentially causing errors in the system's operation or leading to data loss.

```
"status": "Accepted"
```

Table 7.1 OCPP Messages and Their Vulnerability to Attack Vectors

OCPP Message	**Interception**	**Modification**	**Impersonation**	**Denial of Service**
Boot Notification	✓	✓	✓	✓
Boot Notification Response	✓	✓	✗	✓
Authorize Request	✓	✓	✓	✗
Authorize Response	✓	✓	✗	✓
Start Transaction Request	✓	✓	✓	✗
Start Transaction Response	✓	✓	✗	✓
Meter Value Request	✓	✓	✗	✗
Meter Value Response	✓	✓	✗	✓
Stop Transaction Request	✓	✓	✗	✓
Stop Transaction Response	✓	✓	✗	✓
Disconnect Notification	✓	✓	✗	✓
Disconnect Response	✓	✓	✗	✓

To understand the potential impact of a MitM attack on the communication between the CP and ChgSrv, we analyzed the different types of messages exchanged. Each message type is assessed for its vulnerability to various attack vectors, such as interception, modification, impersonation, and denial of service. The following (Table.7.1) summarizes these vulnerabilities.

7.3 Performance Evaluation

This section provides a description of the experimental setup used to implement the MitM attack on the OCPP communication protocol used in EVCS. Additionally, it includes an analysis of the traffic intercepted during the MitM attack in the result analysis section.

7.3.1 Simulation Environment

For this proof-of-concept, an Ubuntu VM and a Kali Linux VM are used. The Ubuntu VM hosts the ChargeBox simulator, including the OCPP server and OCPP client (OCPP charger). The Kali Linux VM conducts the MitM attack, utilizing its pre-installed penetration testing tools. First, network discovery is performed to identify the IP addresses of the Ubuntu VM and the target device (OCPP client) using nmap on Kali Linux with the following command:

```
nmap -sn 172.24.32.0/24
```

After identifying the IP,the default gateway is located. Thus Mitm is conducted using ARP spoofing command as follows:

```
sudo arpspoof -i eth0 -t UbuntuIP gatewayIP
sudo arpspoof -i eth0 -t gatewayIP UbuntuIP
```

In this above command UbuntuIP used is (172.24.46.82) and gatewayIP used is (172.24.32.1). To facilitate packet forwarding between the Ubuntu VM and the gateway, IP forwarding is enabled in kali machine using the below command as follows:

```
echo 1|sudo tee /proc/sys/net/ipv4/ipforward
```

This command will help to forward the communication of packets to kali machine. Then finally traffic between the OCPP client and server is captured using tcp dump. After capturing the necessary traffic, IP forwarding is disabled. This setup allows for the successful

performance of a MitM attack on OCPP communication, with the captured pcap file ready for analysis in the result analysis section.

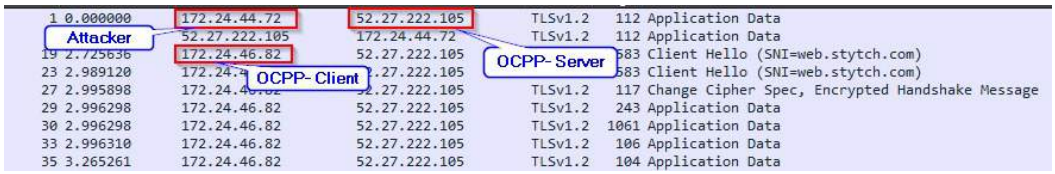


Fig. 7.3 Network traffic capture using tcp dump

After performing the MitM attack and capturing the network traffic using tcp dump as shown in (Fig. 7.3), it was observed that the traffic displayed includes communication between the OCPP client and server. Importantly, due to the MitM attack, the attacker's IP address is interjected into this communication flow. This presence indicates that the attacker successfully intercepted between the OCPP client and server, demonstrating the vulnerability of the communication to such attacks.

7.3.2 Result Analysis

While analysing the network traffic between the OCPP client and server, it was observed that TLS 1.2 is utilized to encrypt their communication. This ensures that most of the data exchanged remains confidential and integral during transmission. Although the communication packets such as the Client Hello and Server Hello, contains critical information such as session IDs and cipher suite.

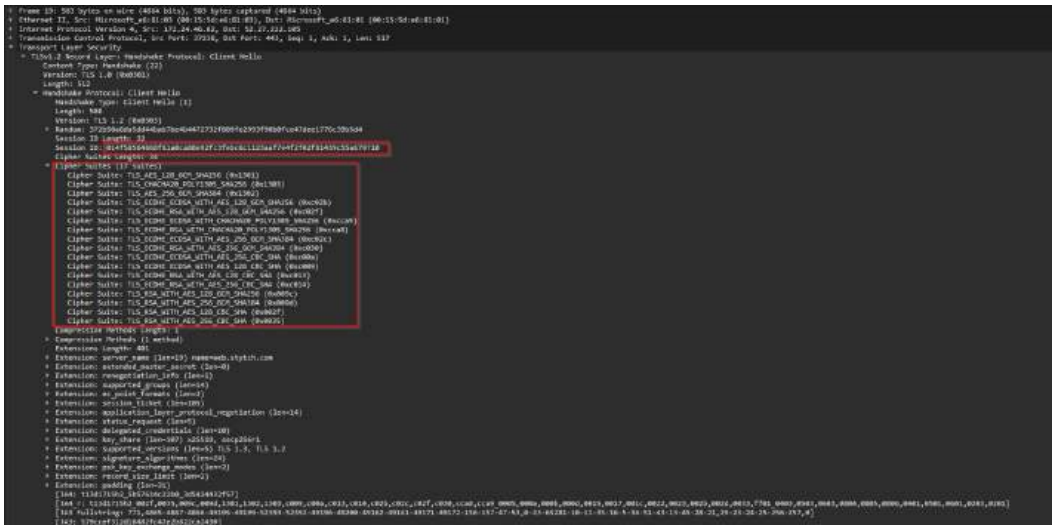


Fig. 7.4 Client Hello message analysis

(Fig.7.4) shows the Client Hello message, where the OCPP client initiates communication with the server. This message typically includes details like the chosen cipher suites, supported TLS versions, and a session ID if a session is being resumed. (Fig.7.5) depicts the Server Hello message, where the server responds to the client's hello. This message confirms the selected cipher suite from the client's list and includes its own session ID and additional security parameters.

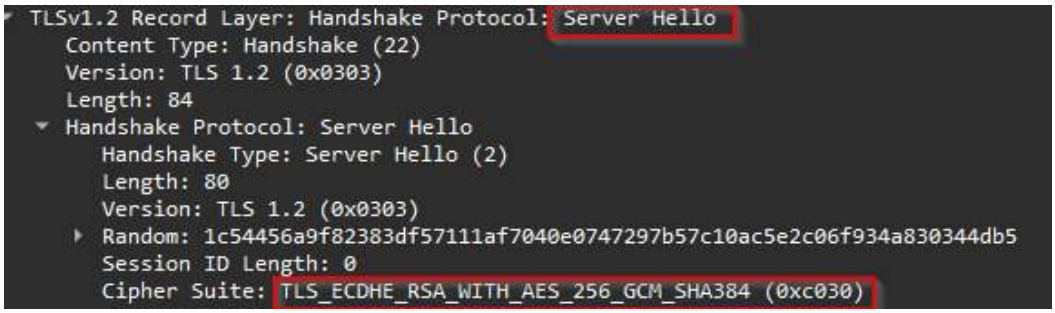


Fig. 7.5 Server Hello message analysis

These messages are integral parts of the TLS handshake process, establishing a secure connection between the OCPP client and server. The presence of session IDs and cipher suite information in these messages highlights key aspects of the security configuration used for encryption and authentication during communication.Session IDs in TLS are used to resume previous sessions, which can potentially expose session-related information if intercepted. An attacker intercepting a session ID could use it to hijack or impersonate the session.The cipher suite specifies the encryption algorithms and cryptographic parameters used to secure the communication. If intercepted, an attacker could analyse the cipher suite to identify potential vulnerabilities or weaknesses. This could lead to decryption of intercepted data if the cipher suite used is compromised. Thus, interception of session IDs and cipher suite details during the TLS handshake exposes the communication to various risks, including session hijacking, decryption of intercepted data, and potential compromise of the entire communication channel. Also, protocols used at the application layer are visible, as depicted in (Fig.7.6). Attackers can exploit vulnerabilities in these protocols.

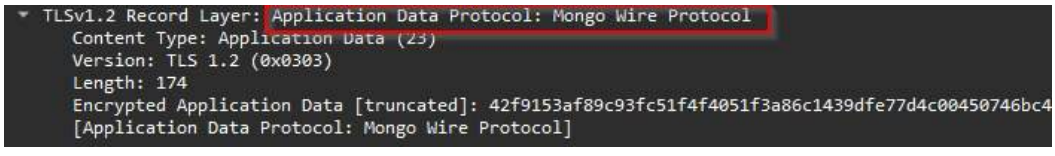


Fig. 7.6 Application data protocol analysis

Additionally,OCPP client and server communication is encrypted using TLS 1.2. However, TLS 1.2, while widely used, is known to have vulnerabilities that can compromise its

security [164]. TLS 1.2 lacks support for modern cryptographic algorithms and security enhancements found in later TLS versions, such as TLS 1.3 [164]. These vulnerabilities make TLS 1.2 potentially susceptible to interception, decryption, and manipulation of encrypted data by skilled attackers. As a result, relying solely on TLS 1.2 for securing sensitive communications, such as those in OCPP for EVCS, poses risks to data confidentiality and overall system integrity.

To mitigate these risks and enhance security, it is imperative for OCPP implementations to upgrade to the latest TLS version available. The latest TLS versions will address known vulnerabilities, introduce stronger encryption algorithms, and provide better resistance against MitM attack. Also measures can be taken at network level to safeguard network from MitM attack using VPN and Endpoint detection and response Tools. These proactive approach ensures that OCPP systems benefit from improved security measures, safeguarding against potential exploits and ensuring secure and reliable operation in modern cybersecurity landscapes.

7.4 Chapter Summary

The study highlights significant vulnerabilities in the OCPP 1.6, particularly concerning its susceptibility to MitM attacks. Despite using TLS 1.2 for encryption, which has known vulnerabilities, the protocol leaves critical data like session ID, cipher suite and application level protocol information exposed to interception. This exposes EVCS to risks such as data manipulation and service disruptions. Moving forward, adopting newer, more secure versions of TLS and implementing robust encryption practices are essential to mitigate these risks and ensure the secure operation of EVCS infrastructure. In the future, conducting further assessments of vulnerabilities in OCPP and implementing corresponding mitigations will be essential to enhance the security of EVCS, thereby ensuring their resilience in the evolving landscape of electric vehicle technologies.

Chapter 8

Cyber defense in OCPP

This chapter aims to provide a comprehensive approach to enhancing the security of OCPP by identifying potential vulnerabilities, assessing risks, and proposing actionable mitigation strategies.In this chapter **Section 8.1** provides background on the OCPP communication architecture and the associated risks. **Section 8.2** outlines the proposed security framework, explaining the OCPP transaction process and introducing the IAAM framework. **Section 8.3** identifies OCPP threats using the STRIDE model. **Section 8.4** analyses OCPP threats using an attack tree, including an attack workflow to better understand the flow of each attack. **Section 8.5** evaluates the assessed threats with the DREAD model to calculate the impact, likelihood, and risk of each identified attack. **Section 8.6** proposes mitigation strategies through enhanced communication processes for OCPP. **Section 8.7** presents a result analysis, including experimental setup along with server logs and data-centric evaluations of the improved OCPP. **Section 8.8** compares the state-of-the-art analysis with current research work. Finally, **Section 8.9** concludes the study and suggests future research directions.

8.1 Background

The rise of EVCS globally is closely linked to the increasing adoption of EVs [125]. Facilitating this growth is OCPP, a communication standard designed to enhance interactions between EVCS and central management system[9]. (Fig.8.1) illustrates how OCPP facilitates seamless communication between the OCPP client and the OCPP server. This protocol supports essential functionalities such as remote operations, reservation handling, and smart charging capabilities, crucial for efficient EV charging [9]. OCPP accommodates both SOAP/XML and Web Sockets/JSON protocols, ensuring robust and flexible communication [166]. SOAP/XML provides compatibility with older systems, ensuring clear and structured messaging, while Web Sockets/JSON enables faster, real-time interactions by maintaining

open connections and minimizing delays. Despite the availability of OCPP 2.1 with advanced features and improved security measures, OCPP 1.6 remains widely adopted due to its ease of implementation, user-friendly interface, and extensive existing infrastructure [11].

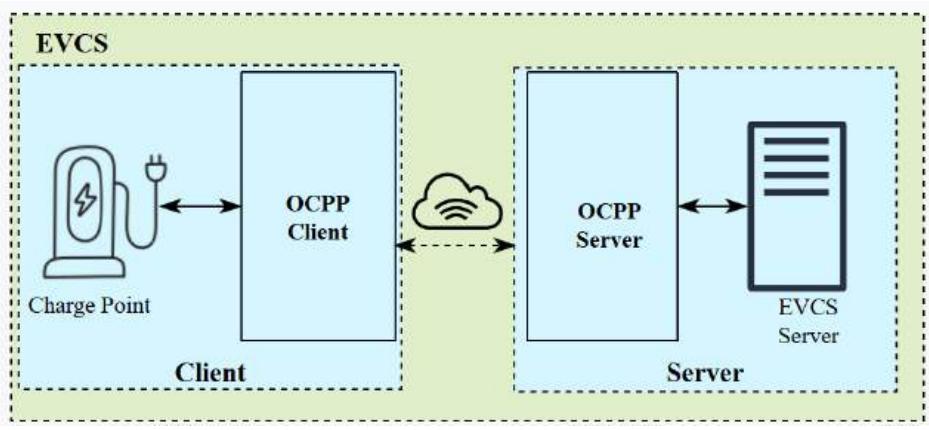


Fig. 8.1 OCPP Communication Architecture

However, the extensive use of OCPP has also brought to light several critical security vulnerabilities [10]. Practical exploits involving unencrypted Web Socket communication within OCPP have demonstrated how attackers could terminate charging sessions, execute remote code, and deploy malicious firmware on EVSE [166]. Saiflow have identified improper handling of multiple connections from a single CP [108], potentially exposing charging infrastructure to DoS attacks. The investigation carried by them also uncovered inadequate authentication mechanisms within OCPP [108], leaving systems vulnerable to unauthorized access and potential exploitation by malicious actors. Flaws in the management of reservation IDs have been found [13], highlighting risks such as fraudulent transactions and misuse of charging stations within OCPP networks. Additionally, there is no authentication or multi-factor authentication done before the CP is connected to the ChgSrv [21], making it possible for attackers to hijack connections using stolen user credentials and reservation IDs. Additionally, flaws in the implementation of Transport Layer Security (TLS) expose OCPP to Man-in-the-Middle (MitM) attacks, enabling attackers to intercept and modify data during transmission [104]. Thus, insecure backend communication, firmware theft, and unauthorized access to EV data further compromise the confidentiality and integrity of charging transactions [21], [19]. These tangible threats underscore the need for immediate security enhancements.

Mitigating these risks is crucial to ensuring the secure and reliable functioning of EV charging infrastructure. As electric mobility becomes increasingly essential in the global efforts to reduce carbon emissions and combat climate change [3], the security of charging networks is critical. Failing to address vulnerabilities could result in operational disruptions,

financial losses, and a decline in user trust in EV technologies [28]. This paper seeks to provide an in-depth analysis of the security risks identified in OCPP and suggest effective improvements and mitigation strategies to strengthen the protocol against potential threats. By enhancing security within OCPP, the goal is to foster the continued growth and adoption of electric mobility solutions while protecting against cyber threats.

8.2 Proposed Security Framework

This section begins with an overview of the OCPP transaction flow followed by proposed framework design to evaluate the vulnerabilities identified in the OCPP transaction process.

8.2.1 OCPP Transaction

The OCPP facilitates communication between CP and ChgSrv in the EV charging infrastructure. This protocol enables essential information exchange for managing and monitoring charging sessions across diverse charging stations. Messages in OCPP follow a request-response model as shown in (Fig.8.2), where CP initiate requests to ChgSrv for operations like starting or stopping charging sessions, retrieving status information, and reporting diagnostics. These messages ensure effective management and monitoring of EV charging sessions. Following are the key message request and response which form the core of the OCPP protocol:

1. **Connect Request and Response:** The Connect request is initiated by the CP to establish a communication link with the ChgSrv. This message ensures that the CP is recognized and can communicate with the ChgSrv. The corresponding response from the ChgSrv confirms the establishment of the connection and provides any necessary configuration parameters.

2. **Authorize Request and Response:** Before a charging session can begin, the CP sends an Authorize request to the ChgSrv to verify the credentials of the user. The ChgSrv processes this request to ensure that the user is permitted to use the charging service. The Authorize response from the ChgSrv includes the authorization status, indicating whether the charging session can proceed.

3. **Start Transaction Request and Response:** Once authorization is successful, the CP sends a Start Transaction request to the ChgSrv to initiate the charging session. This request includes details such as the user ID and the connector used. The ChgSrv

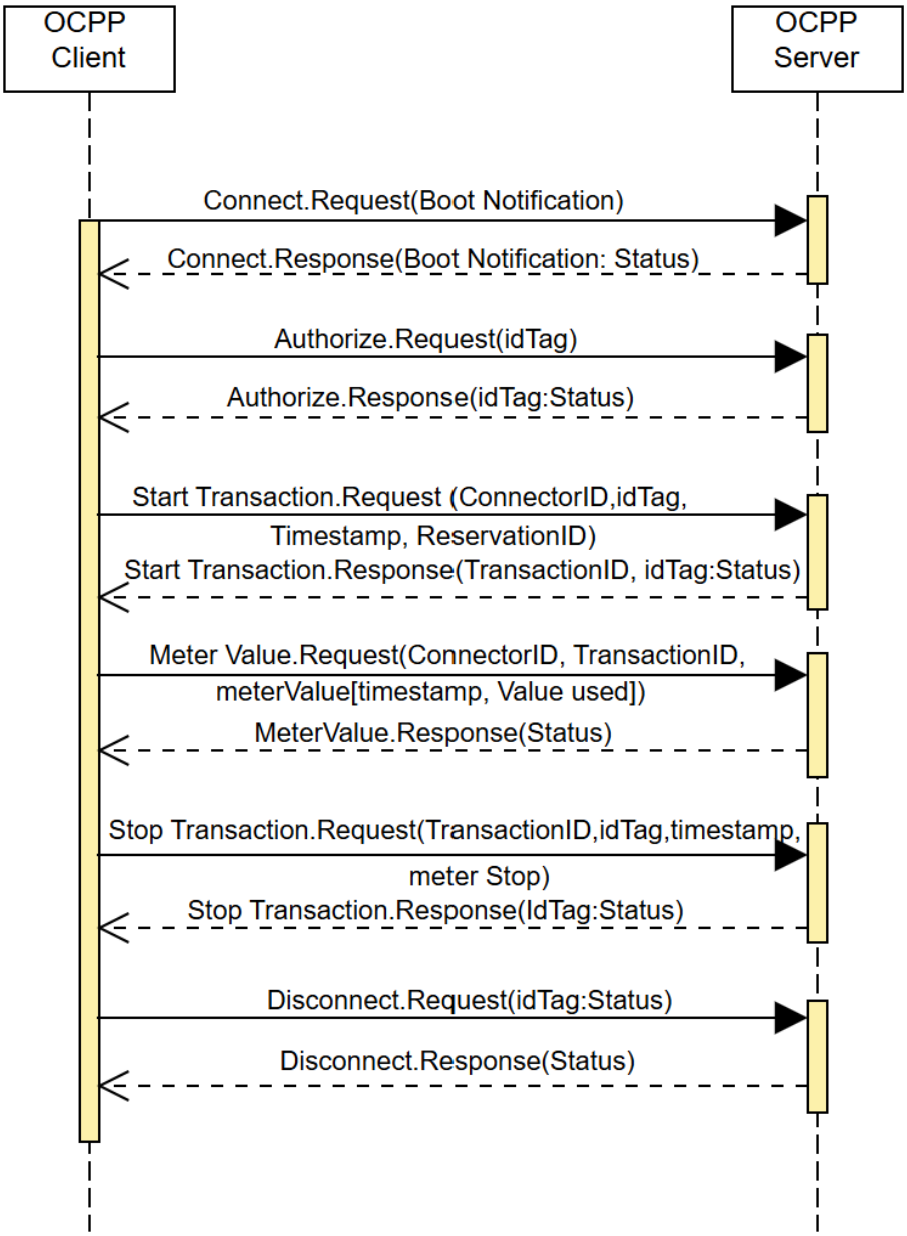


Fig. 8.2 Transaction message flow in OCPP

responds with a Start Transaction response, which includes a unique transaction ID and the status of the request, confirming that the charging session has started.

4. **Stop Transaction Request and Response:** To end a charging session, the CP sends a Stop Transaction request to the ChgSrv. This request contains the transaction ID, user ID, timestamp and meter reading. The ChgSrv processes this request and sends back a Stop Transaction response, confirming the end of the charging session and providing final transaction details.

5. **Meter Values Request and Response:** During an ongoing charging session, the CP periodically sends Meter Values requests to the ChgSrv to report the current meter readings. This allows the ChgSrv to monitor the energy consumption in real-time. The ChgSrv responds with a Meter Values response, acknowledging the receipt of the meter data.

6. **Disconnect Request and Response:** If the communication between the CP and ChgSrv needs to be terminated, a Disconnect request is sent by the CP. This message informs the ChgSrv that the CP will no longer be available for communication. The ChgSrv responds with a Disconnect response, confirming that the disconnection process has been acknowledged and completed.

8.2.2 IAAM Framework

In this section, we propose the Identify, Analyse, Assess, Mitigate (IAAM) framework as shown in (Fig.8.3) to enhance the security of OCPP. This framework introduces a structured approach within the EV charging ecosystem, ensuring that the OCPP protocol is more resilient against potential threats.

1. **Identify:** The first phase involves the systematic identification of security vulnerabilities within the OCPP protocol. Utilizing the STRIDE framework, potential threats are classified. This step ensures that no threat category is overlooked during the security evaluation process.

2. **Analyse:** Once vulnerabilities are identified, the next step is to analyse them using an attack tree model. This analysis maps out potential attack pathways, highlighting how adversaries could exploit weaknesses in OCPP. The attack tree structure facilitates a clear visualization of each vulnerability root cause and possible escalation paths.

3. **Assess:** After analysing vulnerabilities, the framework assesses the severity and risk of each threat using the DREAD model. This phase evaluates the potential damage,

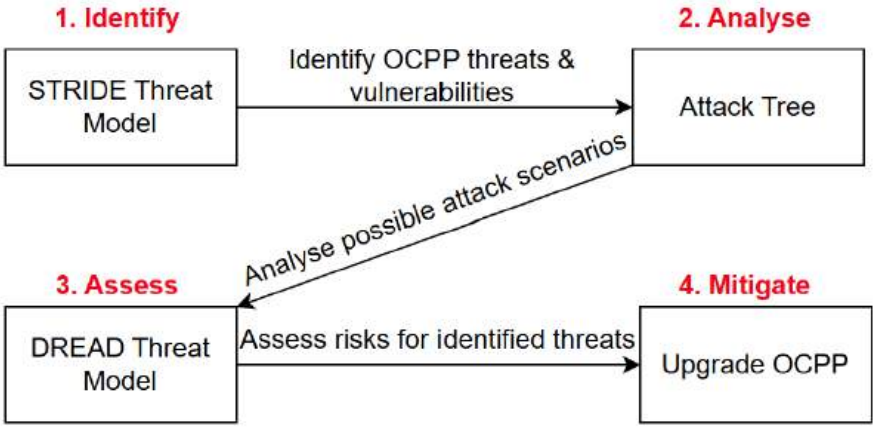


Fig. 8.3 IAAM Framework

reproducibility, exploit ability, affected users, and discoverability of each vulnerability, resulting in a prioritized list of risks to address.

4. **Mitigate:** The final step is to propose mitigation strategies aimed at upgrading the OCPP protocol. The proposed mitigation strategies are designed to address both immediate vulnerabilities and long-term system resilience, ensuring a secure EV charging ecosystem.

The IAAM model ensures a comprehensive security approach for OCPP, offering a continuous cycle of vulnerability management and system improvement.

8.3 Identify Threat: STRIDE

The STRIDE model, developed by Microsoft, is a widely used threat classification framework designed to identify security risks across six key categories[167] as given below:

1. **Spoofing:** Spoofing refers to an attacker's ability to impersonate a legitimate entity
2. **Tampering:**Tampering involves the unauthorized alteration of data during transmission.
3. **Repudiation:** Repudiation occurs when an action cannot be traced to its origin, making it difficult to determine who initiated certain operations.
4. **Information Disclosure:**Information disclosure involves the exposure of sensitive data to unauthorized parties.

5. **Denial of Service:** Denial of Service attacks aim to disrupt the normal operation of a system by overwhelming resources or making services unavailable.

6. **Elevation of Privilege:** Elevation of Privilege refers to a situation where an attacker gains higher-level access than they are authorized for.

Each category corresponds to a specific type of security threat, allowing for a comprehensive assessment of potential vulnerabilities within EVCS[112]. In this analysis, the STRIDE model has been used to categorize and group vulnerabilities discovered by various authors in their respective works on EVs, EVCS, and communication protocols like OCPP. The research highlights vulnerabilities identified by these authors, mapping them to the relevant STRIDE categories as shown in (Table 8.1). By using this structured approach, the goal is to provide a comprehensive understanding of the threats affecting EV charging infrastructure and to propose strategies for mitigating these security risks.

Koscher et al.[168] provided foundational insights into the security vulnerabilities of Electronic Control Units (ECUs) in vehicles. Their research highlighted the risks of spoofing, where attackers could impersonate legitimate vehicle components, and tampering, where malicious alterations to system data could occur. Additionally, they identified the potential for DoS attacks, which could disrupt vehicle operations by overwhelming ECUs. The study also discussed the risks of elevation of privilege, where attackers could gain unauthorized access to higher-level system controls, and repudiation, where actions taken by attackers could not be traced or attributed back to them. These findings set the stage for subsequent studies on improving the security of ECUs in automotive systems. Rouf et al.[169] built on this knowledge by examining the Tire Pressure Monitoring System and identifying how attackers could exploit authentication weaknesses to inject spoofed messages,leading to tampering and data manipulation. They also identified the risk of repudiation, where attackers could deny their involvement, and information disclosure, where sensitive data could be accessed. Additionally, they pointed out the potential for DoS attacks, which could disrupt the Tire Pressure Monitoring System by overwhelming it with malicious messages. Their work emphasized the significant threats to automotive systems and the need for enhanced security. Similarly, Checkoway et al.[170] expanded on the vulnerabilities presented by Koscher et al.,addressing spoofing, tampering, and Information Disclosure risks through insecure interfaces and unprotected firmware updates. They emphasized the need for robust security in third-party components, highlighting the potential for Repudiation, Privilege Escalation, and DoS attacks. Their emphasis on the need for robust security in third-party components echoed the concerns raised in Rouf et al.'s work. Woo et al.[20] analysed Controller Area Network (CAN) vulnerabilities, highlighting how malicious applications could impersonate legitimate

Table 8.1 Analysis of Vulnerabilities in EV Charging System

Research	Threat actor			Threat class					
	EV	EVCS	OCPP	Spoofing	Tampering	Repudiation	Info Disclosure	Denial of Service	Elevation of Privilege
Koscher et.al[168]	✓	-	-	✓	✓	✓	-	✓	✓
Rouf et al.[169]	✓	-	-	✓	✓	✓	✓	✓	-
Checkoway, S., et al.[170]	✓	-	-	✓	✓	✓	✓	✓	✓
Woo et al.[20]	✓	-	-	✓	✓	✓	✓	✓	-
Jafarnejad, S. et.al.[171]	✓	-	-	✓	✓	✓	-	✓	-
Garcia et al.[172]	✓	-	-	✓	✓	✓	-	✓	-
Mazloom, S. et.al.[173]	✓	-	-	✓	✓	-	✓	-	✓
Karthik, T. et.al.[22]	✓	-	-	✓	✓	-	✓	✓	-
Currie, R. et al.[23]	✓	-	-	✓	✓	✓	-	✓	-
Luo, Q. et.al.[174]	✓	-	-	✓	✓	-	✓	-	✓
Jouvray, C. et al.[175]	-	✓	-	✓	✓	✓	✓	✓	✓
Schneider[6]	-	✓	-	✓	✓	-	✓	-	-
Circontrol[176]	-	✓	-	✓	-	-	✓	-	-
P. van Aubel et al.[177]	-	✓	-	✓	✓	-	✓	-	-
Antoun, J., et.al[178]	-	✓	-	✓	✓	✓	✓	✓	✓
Acharya et al.[179]	-	✓	-	✓	✓	-	-	-	-
Girdhar, M. et al.[111]	-	✓	-	✓	✓	✓	-	-	-
Carryl, C. et.al[24]	✓	✓	-	✓	-	-	✓	✓	-
Baker, R. et.al.[78]	✓	✓	-	✓	-	-	✓	✓	-
Sayed, M.A. et.al[4]	✓	✓	-	✓	✓	-	-	-	✓
Rubio, J.E. et al.[104]	-	-	✓	✓	✓	-	✓	-	-
Alcaraz, C. et al.[13]	-	-	✓	✓	✓	-	✓	✓	-
Garofalaki, Z. et.al[10]	-	-	✓	✓	✓	-	✓	✓	✓
Gebauer, L. et al.[19]	-	-	✓	✓	✓	-	✓	✓	✓
Johnson et al.[166]	-	-	✓	✓	✓	✓	✓	✓	✓
Sarieddine, K. et al.[21]	-	-	✓	-	✓	✓	✓	✓	✓

ones to execute spoofing attacks. Their research also covered tampering and privilege escalation risks, demonstrating how attackers could manipulate vehicle communication. Repudiation and DoS threats were identified, along with the possibility of Information Disclosure through compromised systems. Their findings, focus on vehicular network intrusion detection, emphasize the systemic risks EVs face as they adopt increasingly complex communication protocols.

Jafarnejad et al.[171] analysed vulnerabilities in the Sevcon Gen4 controller, identifying risks such as Spoofing, where attackers could impersonate legitimate controllers, and Tampering, where malicious actors could alter the data transmitted by the controller. The study also highlighted Repudiation, where attackers could deny their malicious actions, and DoS, which could disrupt the controller's functionality, leaving the system inoperable. Similarly, Garcia et al.[172] focused on the security of Remote Keyless Entry systems, demonstrating how weak authentication mechanisms could lead to Spoofing through remote cloning, Tampering with communication between the key fob and the vehicle, and Repudiation, where attackers could deny their involvement in malicious activities. The study also pointed out DoS attacks that could disable keyless entry systems. Mazloom et al.[173] analysed vulnerabilities in the MirrorLink protocol for In-Vehicle Infotainment systems. They identified Spoofing, where attackers could impersonate devices, Tampering, where data could be altered during communication, Information Disclosure, where sensitive data could be leaked, and Elevation of Privilege, where attackers could gain unauthorized control over the infotainment system. Karthik et al.[22] discussed the risks in automotive software update systems, emphasizing Spoofing through disguised malicious updates, Tampering with update files, Information Disclosure regarding sensitive data, and DoS, which could hinder the update process, rendering ECUs inoperable. Currie et al.[23] examined vulnerabilities in the CAN bus, revealing how Spoofing could be used to impersonate CAN devices and Tampering with data could occur during transmission. They also highlighted the issue of Repudiation, where the origin of malicious messages could be denied, and DoS attacks that could target vehicle functions. Lastly, Luo et al.[174] focused on wireless telematics systems in connected vehicles, showing that Spoofing could involve malicious actors impersonating vehicle systems, Tampering could occur through data alterations, Information Disclosure could lead to sensitive vehicle data exposure, and Elevation of Privilege could allow attackers to gain unauthorized control over the vehicle's telematics system through malware.

Jouvray et al.[175] were among the first to identify vulnerabilities within the ISO/IEC 15118 and Power Line Communication protocols, which are critical for EV charging. Their research emphasized Spoofing, specifically through contract reuse, where attackers could impersonate legitimate users or entities, and Tampering, where they could modify charging

configurations. They also highlighted Information Disclosure, where sensitive data could be exposed due to weak security measures, laying the groundwork for understanding the security challenges in EVCS. Additionally, Jouvray et al. pointed out Repudiation risks, where attackers could deny their actions, and Elevation of Privilege, which could enable unauthorized access to higher system privileges, underscoring the need for secure architecture in EVCS systems. Schneider Electric's analysis[6] expanded on these concerns by examining their EVLink Parking EVCS. They identified Spoofing risks due to hard-coded credentials, which attackers could exploit to gain unauthorized access to the system. Their study also revealed Tampering vulnerabilities through code injection, and Information Disclosure via SQL injection attacks, which could expose sensitive data through software vulnerabilities. Circontrol's CirCarLife report[176] reinforced these findings, specifically addressing Spoofing through an authentication bypass (CVE-2018-17918) and Information Disclosure through another vulnerability (CVE-2018-17922) that exposed sensitive credentials in plaintext.

P. van Aubel et al.[177] proposed a cryptographic solution to address vulnerabilities in EV charging systems, focusing on Spoofing, Tampering, and Information Disclosure. Their approach involved implementing digital signatures, encryption, and authentication trees to mitigate these risks. They emphasized the importance of using proper recipient and signer identifiers to prevent attackers from impersonating legitimate entities, altering data, or exposing sensitive information. Antoun et al.[178], using the STRIDE framework, analysed public EV charging vulnerabilities and reinforced earlier findings. They identified Spoofing where attackers impersonated legitimate entities to steal information, Tampering through unauthorized changes to messages, and Repudiation risks where compromised systems could deny payment actions. They also pointed out Information Disclosure vulnerabilities that allowed attackers to access sensitive data, alongside DoS attacks, which could overwhelm network services, and Elevation of Privilege scenarios where attackers gained unauthorized control over infrastructure. Acharya et al.[179] focused on vulnerabilities in EV charging infrastructure, highlighting Spoofing of charging commands and Tampering with EVCS servers via malware installation. Their research underscored the risk of these attacks leading to significant disruptions to the electric grid, showing the interconnectedness of EV charging systems and grid infrastructure. Finally, Girdhar et al.[111] focused on extreme fast charging systems, identifying similar Spoofing, Tampering, and Repudiation threats, emphasizing the critical need for robust security measures in the evolving EV charging infrastructure.

Carryl et al.[24] examined grid networks in the context of EV charging, revealing vulnerabilities related to Spoofing, Information Disclosure, and DoS. They demonstrated how malicious actors could impersonate legitimate entities to manipulate data and overload the grid. Their study highlighted that insufficient security measures for transmitted data could

lead to Information Disclosure, exposing sensitive data to unauthorized access. Additionally, they identified DoS risks, where fake requests could overwhelm the system, compromising the grid's stability. Baker et al.[78] reinforced these findings by identifying similar vulnerabilities, particularly Spoofing and Information Disclosure, within EV charging infrastructure. Their research emphasized how weak encryption and unprotected communication channels could facilitate unauthorized access, allowing attackers to manipulate charging processes. Like Carryl et al., they raised concerns about DoS vulnerabilities, where attackers could flood the network with fake requests, threatening the operational integrity of EVCS. Sayed et al.[4] expanded on these vulnerabilities by focusing on Spoofing, Tampering, and Elevation of Privilege risks within EV systems. Their research demonstrated how attackers could manipulate charging requests and grid operations, highlighting the interconnected risks between EVs and EVCS. By emphasizing these vulnerabilities, Sayed et al. showcased the potential for serious disruptions in charging infrastructure and overall grid management.

Rubio et al.[104] took a proactive approach in mitigating OCPP risks by focusing on cryptographic solutions aimed at addressing Spoofing, Tampering, and Information Disclosure vulnerabilities. Their exploration of secret-sharing schemes enhanced the security of communications between CPs and CSs, effectively reducing risks associated with unauthorized access and data manipulation. Alcaraz et al.[13] examined vulnerabilities specific to OCPP, revealing how attackers could intercept and alter communications between CPs and CSs. Their findings highlighted Spoofing and Tampering risks, alongside significant concerns regarding Information Disclosure and DoS attacks, which could disrupt the functionality of charging stations. Garofalaki et al.[10] applied the STRIDE framework to analyse security risks within EV charging infrastructure, identifying similar Spoofing vulnerabilities stemming from weak authentication practices, particularly in older OCPP versions. Their study also uncovered Tampering risks, particularly through Address Resolution Protocol (ARP) spoofing, and highlighted Information Disclosure due to poor encryption practices. They echoed the concerns raised by Alcaraz et al. regarding DoS and Elevation of Privilege vulnerabilities, underscoring the systemic issues faced by EV charging systems.

Gebauer et al.[19] further mapped OCPP vulnerabilities to the STRIDE framework, emphasizing the critical risks of Spoofing by malicious OCPP servers and Tampering with data. Their work highlighted significant Information Disclosure risks posed by malicious code infiltrating the system, alongside persistent concerns regarding DoS attacks and Elevation of Privilege scenarios, indicating a pressing need for enhanced security measures across the protocol. Johnson et al.[166] built upon these findings by identifying specific vulnerabilities in OCPP, mapping them to the STRIDE framework. Their research revealed Spoofing risks stemming from weak authentication mechanisms, Tampering threats evident

during MitM attacks, and Repudiation issues due to inadequate logging. They also pointed out the critical risks of Information Disclosure due to unencrypted communications and the potential for DoS attacks that could overwhelm Central System Management Systems with fake requests. Additionally, they noted the Elevation of Privilege risk associated with the Log4Shell vulnerability. Finally, Sarieddine et al.[21] investigated backend vulnerabilities within OCPP, identifying improper authentication as a means for attackers to spoof legitimate connections. Their analysis also revealed Tampering through phantom EVCS hijacking connections, Repudiation issues stemming from untraceable actions, and Information Disclosure risks due to session fixation. They highlighted the dangers of unrestricted authentication attempts leading to potential DoS attacks, alongside Elevation of Privilege risks related to single-factor authentication vulnerabilities.

While previous studies have analysed vulnerabilities in EVs, EVCS, and OCPP, this work specifically focuses on OCPP. The approach is distinct in that it aims to enhance the security of older versions of OCPP, the most widely adopted in existing EVCS infrastructure. Unlike newer OCPP versions that introduce security improvements but require hardware upgrades due to a lack of backward compatibility, this work strengthens the security of older versions of OCPP without necessitating costly infrastructure changes. This allows charge stations to mitigate vulnerabilities while continuing to operate with their current hardware, bridging the gap between security advancements and real-world deployment challenges.

Table 8.2 Threats Mapped to STRIDE Categories

STRIDE Category	**Threat Name**	**Threat Actor**
Spoofing	Th1: Cloning Attack	EV / EVCS
	Th2: Identity Theft	OCPP
	Th3: Replay Attack	OCPP
Tampering	Th4: Data Tampering	OCPP
	Th5: Firmware Injection	EV
Repudiation	Th6: Payment Fraud	OCPP
Information Disclosure	Th7: Data Breach	OCPP
Denial of Service	Th8: DoS	OCPP
Elevation of Privilege	Th9: Privilege Escalation	OCPP

Based on the vulnerabilities identified by the researchers, threats have been classified and categorized using the STRIDE framework, with a primary focus on the OCPP as a threat actor, as presented in (Table 8.2). Understanding these following threats is crucial for developing effective security measures to mitigate vulnerabilities in OCPP:

Th1: Cloning Attack: Attackers clone CP or ChgSrv, enabling unauthorized access to EVs or EVCS, potentially leading to theft or control of the EV.

Th2: Identity Theft: This threat can be executed by reusing or stealing credentials used in OCPP communications, allowing unauthorized users to impersonate legitimate entities within the OCPP ecosystem.

Th3: Replay Attack: Attackers can capture OCPP messages (e.g. Start Transaction requests) and resend them to initiate unauthorized actions, leveraging the inherent vulnerabilities in OCPP.

Th4: Data Tampering: Data tampering can occur within the OCPP communications, where attackers modify messages exchanged between EVs and EVCS, leading to erroneous transactions or operations.

Th5: Firmware Injection: While this primarily targets the EV itself, vulnerabilities in OCPP could be exploited to inject unauthorized firmware during the update process via the communication channel.

Th6: Payment Fraud: This threat arises from compromised EVCS or systems utilizing OCPP, leading to discrepancies in the payment amounts displayed for a single transaction. This can result in the denial of legitimate payment transactions, causing financial losses for both consumers and service providers.

Th7: Data Breach: Unencrypted communications or inadequate security measures in OCPP can lead to data breaches, exposing sensitive user or vehicle information. Although TLS 1.1 is used but its considered less secure.

Th8: DoS: A DoS attack can target the OCPP interface by flooding it with connection requests, disrupting normal service operations at the EVCS level.

Th9: Privilege Escalation: In OCPP, the absence of user authentication during the BootNotification allows attackers to bypass security measures and gain unauthorized access to higher privilege levels, potentially leading to misuse of system resources.

8.4 Analyse Threat

8.4.1 Attack Tree

EVCS have become critical components in the infrastructure supporting the transition to electric mobility. These systems rely on the OCPP for backend communications, facilitating interactions between CP and ChgSrv. The (Table 8.3) provides an analysis of identified attacks mapped to STRIDE threats. This table helps to categories the attacks map to STRIDE Categories as shown in (Table 8.4).

Table 8.3 OCPP Attacks mapped to STRIDE Threats

OCPP Attack	**STRIDE Threats**
A1: Data Sniffing	Th7
A2: CP Cloning	Th1, Th2, Th4, Th7
A3: Duplicate Connection	Th2, Th3, Th4 , Th7
A4: Duplicate Booking	Th2, Th3 , Th4, Th7
A5: Conflicting Meter Values	Th2, Th3,Th6, Th7
A6: Flooding Requests	Th8
A7: Injection Attack	Th2, Th4, Th5, Th7, Th9

The attacks underscore the potential vulnerabilities that can be exploited through various threat vectors within the OCPP ecosystem. Each identified attack is mapped to corresponding STRIDE threat categories, illustrating the specific security vulnerabilities present. Additionally, (Fig.8.4) presents an attack tree that demonstrates the flow of threats leading to each of the attcks identified in OCPP, further clarifying how these vulnerabilities can be exploited in real-world scenarios.

Table 8.4 OCPP Attack mapped to STRIDE Categories

OCPP Attack	S	T	R	I	D	E
A1: Data Sniffing				Y		
A2: CP Cloning	Y	Y		Y		
A3: Duplicate Connection	Y	Y		Y		
A4: Duplicate Booking	Y	Y		Y		
A5: Conflicting Meter Values	Y		Y	Y		
A6: Flooding Requests					Y	
A7: Injection Attack		Y		Y		Y

A1: Data Sniffing begins with Th7: Data Breach, where attackers employ packet sniffing techniques to intercept sensitive information transmitted over OCPP communication channels. This can lead to unauthorized access to user details, charging reservation information, or CP details.

A2: CP Cloning starts with Th7: Data Breach, allowing attackers unauthorized access, which leads to Th1: Cloning Attack. Here, attackers clone a CP, facilitating unauthorized access to EVCS. This further escalates to Th2: Identity Theft, enabling attackers to impersonate legitimate CPs by reusing stolen CP credentials. Additionally, this could lead to Th4: Data Tampering, where the attacker may manipulate communication data.

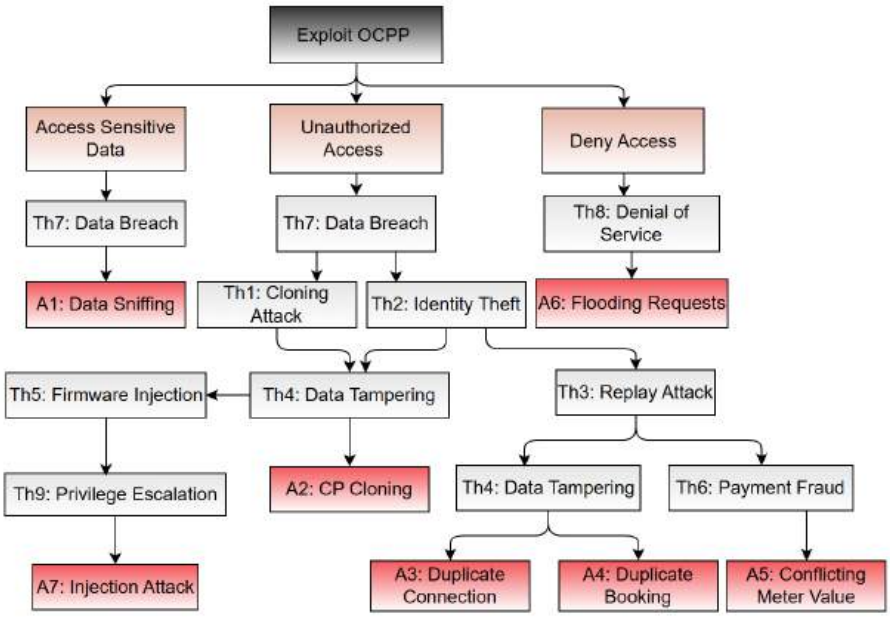


Fig. 8.4 OCPP Attack Tree

A3: Duplicate Connections begin with Th7: Data Breach, leading to unauthorized access. This situation escalates to Th2: Identity Theft and Th3: Replay Attack, allowing multiple unauthorized connections for the same CP. Furthermore, it may also involve Th4: Data Tampering, enabling the impersonation of valid entities by capturing and re-sending legitimate OCPP messages.

A4: Duplicate Booking starts with Th7: Data Breach, leading to unauthorized access, which allows for Th2: Identity Theft. Attackers exploit valid credentials to create unauthorized connections, progressing into Th3: Replay Attack, where reused booking IDs facilitate multiple reservations for the same CP. This scenario also entails Th4: Data Tampering, where attackers may alter the transaction data associated with the booking.

A5: Conflicting Meter Values begin with Th7: Data Breach, allowing unauthorized access. This can escalate to Th2: Identity Theft, enabling unauthorized connections, and further lead to Th3: Replay Attack, facilitating the exploitation of booking IDs. Additionally, Th6: Payment Fraud may occur due to inconsistencies in meter values, causing financial discrepancies for both consumers and service providers.

A6: Flooding Requests starts with the intention to disrupt services. This attack begins with unauthorized access i.e. Th7This can culminate in Th8: DoS, where attackers flood the EVCS with connection requests, disrupting normal operations and denying legitimate access.

A7: The Injection Attack begins with Th7: Data Breach, where attackers gain unauthorized access to the system. This leads to Th2: Identity Theft, enabling attackers to

impersonate legitimate entities and establish unauthorized connections within the EVCS. Following this, Th4: Data Tampering occurs, where attackers manipulate the data exchanged between EVs and the charging station, potentially altering transaction details. This further escalates to Th5: Firmware Injection, allowing attackers to modify the system through unauthorized firmware updates, compromising its functionality. Lastly, Th9: Privilege Escalation occurs, granting attackers unauthorized access to privileged actions within the EVCS, ultimately resulting in a successful Injection Attack (A7).

Table 8.5 Analysis of Attack Vulnerabilities in OCPP

Attack	Vulnerability
	Weak TLS Encryption
A1: Data Sniffing	Attackers utilize weak TLS encryption to intercept sensitive data transmitted over OCPP communication channels, compromising user details and charging reservation data.
	Weak Authentication
A2: CP Cloning	Attackers can clone a Charge Point (CP) to gain unauthorized access to EVs or EVCS, leading to potential theft or manipulation.
A3: Duplicate Connections	Attackers exploit valid credentials to create multiple unauthorized connections for the same CP, compromising the integrity of the system.
A4: Duplicate Booking	Attackers can reuse a booking ID, leading to unauthorized reservations and potentially allowing access to sensitive user information.
	Weak Session Management
A5: Conflicting Meter Values	Inconsistent meter values due to unauthorized access can mislead consumers and service providers, causing financial discrepancies.
	Weak Message Handling
A6: Flooding Requests	Attackers can send an overwhelming number of connection requests, disrupting normal operations and potentially leading to a DoS.
	Firmware Manipulation
A7: Injection Attack	Attackers can inject unauthorized firmware updates into the system, leading to data tampering and potential manipulation of communication between EVs and EVCS.

These attacks arise from identified vulnerabilities, as outlined in the (Table 8.5). The vulnerabilities in OCPP represent critical weaknesses that can be exploited by attackers to disrupt EV charging systems and compromise user data. These vulnerabilities include:

1. **Weak TLS Encryption:** Previous version of OCPP utilizes TLS 1.2, which has known vulnerabilities that make it susceptible to downgrade attacks and data sniffing. Attackers can exploit these weaknesses to intercept sensitive information, such as CP identifiers and reservation data, during OCPP communication, potentially allowing them to impersonate legitimate entities or conduct further attacks.

2. **Weak Authentication:** The OCPP protocol does not enforce user authentication before a CP and ChgSrv establish a connection using the BootNotification message. This lack of authentication means that anyone can attempt to connect to a ChgSrv, increasing the risk of unauthorized access and potential malicious activities.

3. **Weak Session Management:** OCPP lacks sufficient mechanisms to prevent duplicate connections using the same credentials or identifiers. This vulnerability allows attackers to establish multiple unauthorized sessions, which can compromise the integrity of the charging process and lead to confusion or resource misallocation.

4. **Weak Message Handling:** Because users are not authenticated before connecting to the ChgSrv, attackers can initiate multiple Connect.Request messages without any verification. This ability to flood the system with connection requests can overwhelm the ChgSrv and degrade its performance, leading to DoS conditions.

5. **Firmware Manipulation:** The vulnerabilities present in the firmware update process of OCPP enable attackers to inject unauthorized firmware. This can result in data tampering or manipulation of the system's operations, allowing attackers to control communications between EVs and ChgSrv, further jeopardizing the overall security and functionality of the EV ecosystem.

These vulnerabilities highlight the urgent need for improved security measures within OCPP to protect against potential attacks and ensure the integrity and reliability of EVCS.

8.4.2 Attack Workflow

This section outlines the attack workflow as illustrated in (Fig.8.5) related to multiple vulnerabilities present in the OCPP protocol. Each of these vulnerabilities can be exploited by a malicious client (referred to as Phantom CP) acting alongside a legitimate client (referred

to as Legitimate CP). In this setup, two Chargebox simulators are integrated into the system using OCPP to communicate with a central backend. The simulators represent the key operational components of an Electric Vehicle Charging Management System (EVCMS). Chargebox 1 acts as a legitimate client, while Chargebox 2 serves as a phantom client, designed to exploit protocol vulnerabilities.

In this scenario, it is assumed that due to weak TLS encryption, A1: Data Sniffing has taken place. The attacker has intercepted sensitive information from the CP, such as CP identifiers and credentials, during OCPP communication. With this information, the attacker proceeds to carry out A2: CP Cloning, where a phantom CP is created to impersonate the legitimate CP.

Attack workflow for A3: Duplicate Connections- In this scenario, an attacker gains access to the same Charge Point Identifier (CP ID) that is already being used by a legitimate charging station (Chargebox 1). This could happen through various means, such as intercepting the CP ID via unsecured network traffic or through social engineering. The attacker then uses this CP ID to establish a second, parallel connection to the backend system using a phantom chargebox (Chargebox 2). The OCPP protocol does not have sufficient mechanisms to verify or block duplicate connections with the same CP ID. As a result, both the legitimate chargebox and the phantom chargebox maintain active sessions with the server. This allows the attacker to either monitor the communication or potentially interfere with the legitimate chargebox's operations.

Attack workflow for A4: Duplicate Booking- In this case, the attacker manages to obtain the reservation ID that was issued to a legitimate user for a specific charging session. This could be done by eavesdropping on communication between the chargebox and the backend system or by gaining unauthorized access to the system. Armed with the reservation ID, the attacker can either use the same CP ID or a different one (depending on the attacker's objective). By submitting this reservation ID in the system, the phantom chargebox bypasses normal authorization procedures. The system is unable to distinguish between the legitimate user and the attacker, thereby allowing both to initiate charging sessions, potentially leading to improper resource allocation or session control.

Attack workflow for A5: Conflict in Meter Values- After the attacker uses a duplicate reservation ID, a potential side effect is the reporting of conflicting meter values from multiple chargeboxes for the same transaction or reservation. For instance, while the legitimate chargebox (Chargebox 1) may be reporting valid meter readings during an ongoing session, the phantom chargebox (Chargebox 2) might report different, conflicting values for the same transaction. These inconsistencies could arise because both chargeboxes are treated as valid by the backend due to the duplicate reservation ID. This misalignment of meter readings

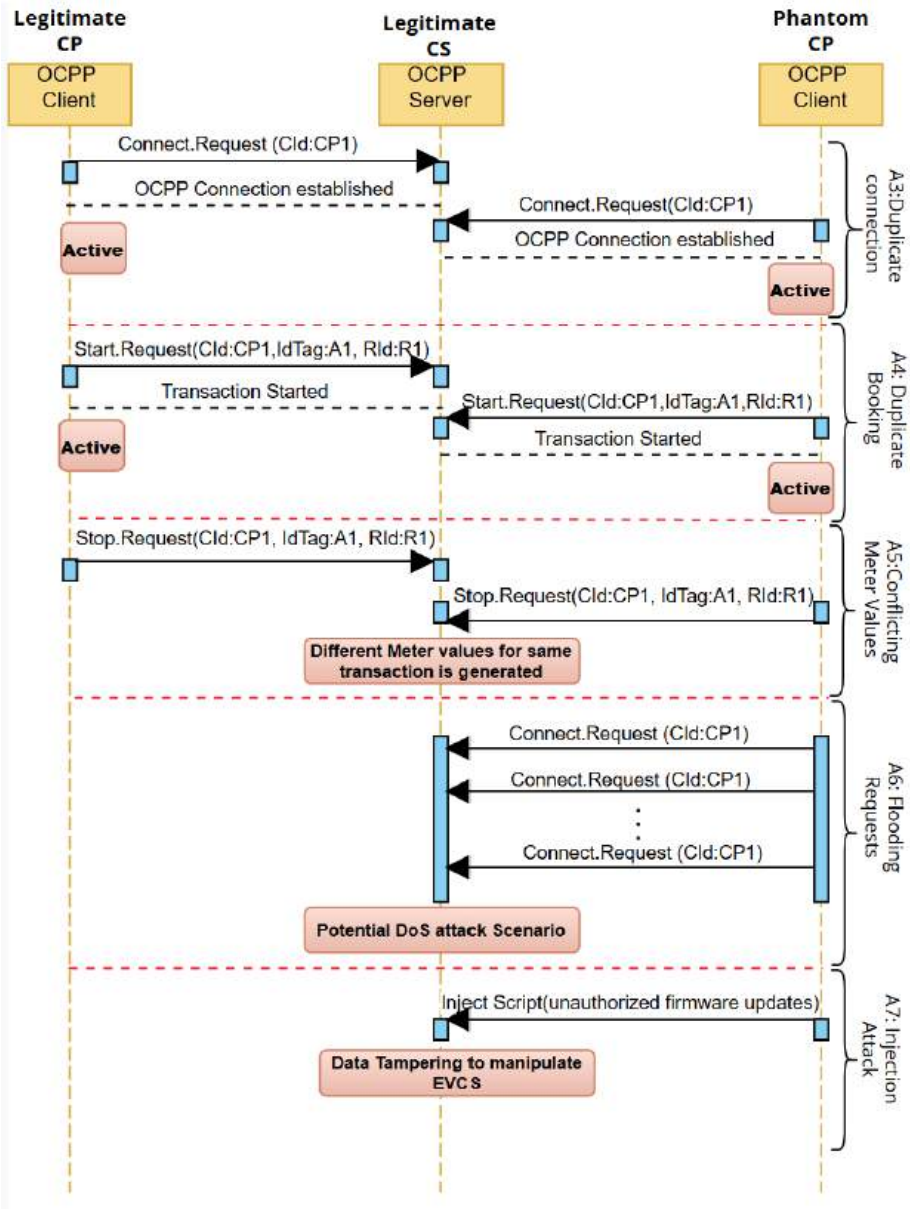


Fig. 8.5 Attack workflow of OCPP vulnerabilities

could introduce errors in energy consumption data, leading to confusion or irregularities in session reporting and billing.

Attack workflow for A6: Flooding Requests- In this type of attack, an attacker (either using a phantom chargebox or by compromising a legitimate one) floods the OCPP backend with an excessive number of requests or messages. For instance, an attacker could send an unlimited number of Connect.Request or BootNotification messages without waiting for the server to respond. This flood of requests could exhaust the backends resources, overwhelming it with message processing tasks. Although this doesn't directly harm individual users, the increased message volume could lead to a degraded performance of the backend system, ultimately affecting the ability of legitimate users to manage their charging sessions.

Attack workflow for A7: Injection Attack-In this scenario, the attacker (using a phantom CP or by compromising a legitimate CP) injects malicious scripts into the OCPP communication. These scripts exploit vulnerabilities in the firmware update process to gain unauthorized access to firmware controls. Once access is granted, the attacker can manipulate the firmware to tamper with data exchanged between the EV and EVCS. This could lead to unauthorized control over the EV charging process, manipulation of energy consumption data, or even disruption of the entire EVCS operation. The tampered firmware may also create persistent vulnerabilities that allow the attacker to maintain control over the system.

8.5 Assess Threat:DREAD

DREAD is a security risk assessment model used to evaluate and prioritize threats by categorizing them into five distinct factors [113], [180] as shown below:

1. **Damage Potential:** This refers to the extent of harm caused if a vulnerability is exploited.For instance, an attacker could manipulate charging data, leading to incorrect billing or financial loss for users and charging station operators. Additionally, there could be risks to the physical infrastructure, such as damage to the charging equipment or EVs, and even potential safety hazards. Evaluating damage potential is crucial to understanding the impact on users, service providers, and the overall charging ecosystem.

2. **Reproducibility:** Reproducibility measures how easy it is to repeat an attack.If an attack, such as spoofing or data tampering, can be consistently performed using publicly available tools or simple techniques, it poses a higher risk. For example, if a vulnerability allows an attacker to repeatedly inject false data into the charging

system, they can exploit it multiple times to defraud users or disrupt service. Assessing reproducibility helps prioritize vulnerabilities that could lead to frequent attacks.

3. **Exploit ability:** This factor assesses how simple it is to exploit the vulnerability.If an attacker can gain unauthorized access to the charging station's management interface using basic hacking techniques or readily available exploits, the exploit ability score will be high. Factors such as the need for specialized knowledge or access to proprietary tools can lower the exploit ability score. Understanding exploit ability helps identify vulnerabilities that require immediate attention, especially those that could be exploited with minimal effort.

4. **Affected Users:** This refers to the number of individuals or entities impacted by the exploitation of a vulnerability. A breach that compromises user data or charging records could affect a large number of EV owners and operators. For example, if a vulnerability allows an attacker to manipulate charging session data, it could result in incorrect billing for numerous users, eroding trust in the charging infrastructure. Evaluating the scale of potential user impact helps in prioritizing vulnerabilities that could affect many people.

5. **Discoverability:** This factor evaluates how easily attackers can find the vulnerability. This includes the visibility of the system's interfaces, the documentation available, and the overall security posture of the charging stations. If vulnerabilities are well-known or widely discussed in the cybersecurity community, attackers may be more likely to exploit them. For instance, if information about a specific vulnerability in a widely used charging protocol is publicly available, the likelihood of it being discovered and exploited increases. Understanding discoverability helps in designing better security measures to protect against potential attacks.

The (Table 8.6) summarizes how each attack maps to the DREAD categories, providing insights into their potential damage, reproducibility, exploit-ability, affected users, and discoverability in relation to OCPP.Each category is scored on a scale of 1 to 10, where the scores reflect the severity of the threat posed by the attack.To standardize the assessment, scores are interpreted as follows given by [113], [180]:

- **High:** Scores ranging from 8 to 10 indicate a significant threat, suggesting that the attack has the potential for severe damage, is easily reproducible, highly exploitable, affects a large number of users, and is easily discoverable.

- **Medium:** Scores from 5 to 7 suggest a moderate threat level. These attacks may have a limited impact or may be more difficult to reproduce, exploit, or discover.

Table 8.6 Assessing OCPP Attacks with DREAD

OCPP Attack	Damage Potential	Reproducibility	Exploit ability	Affected Users	Discoverability
A1: Data Sniffing	Sensitive information such as CP identifiers and credentials are exposed, leading to impersonation and unauthorized access.	The vulnerability is easily exploitable by attackers with access to the communication network.	Requires network interception capabilities, but weak encryption makes it feasible for attackers with moderate expertise.	Multiple users (EV drivers, operators) are at risk if credentials are compromised.	Without strong encryption, discovering this vulnerability is relatively easy through network sniffing tools.
A2: CP Cloning	Unauthorized access to EVCS can result in financial losses and unauthorized charging sessions.	Once a CP ID is intercepted, the attack is easily reproducible.	Exploitation is straightforward as it only requires cloning the CP ID.	Affects multiple users, including EV drivers and service operators.	The cloning method is easily discoverable through monitoring of charging sessions.
A3: Duplicate Connection	Allows multiple unauthorized sessions for the same Charge Point, disrupting service integrity.	Highly reproducible once an attacker intercepts a CP ID.	High exploit ability due to lack of duplicate connection verification in OCPP.	Affects users trying to access services from the same Charge Point.	Discoverability is moderate, as it may not be immediately obvious without monitoring.
A4: Duplicate Booking	Creates unauthorized connections leading to conflicts and service disruptions.	Easily reproducible once reservation IDs are intercepted.	High exploitability, allowing attackers to bypass normal authorization procedures.	Affects legitimate users.	Low discoverability as it requires prior knowledge of reservation IDs.
A5: Conflicting Meter Values	Introduces inconsistencies in energy consumption data, causing billing errors for users and operators.	Moderate reproducibility as it depends on the attacker manipulating meter readings.	Moderate exploitability since it involves manipulating data sent from Charge Points.	Affects both users and operators by leading to incorrect billing.	Low discoverability, as the inconsistencies may only become apparent during audits.
A6: Flooding Requests	Disrupts normal operations, denying legitimate users access to services.	Highly reproducible as attackers can repeatedly send requests.	Highly exploitable since attackers can overwhelm the backend system with requests.	Affects all users trying to access services during the attack.	Moderate discoverability, as the attack can be detected with proper monitoring.
A7: Injection Attack	Allows attackers to modify system functionality, leading to unauthorized control of EV charging processes.	Moderate reproducibility due to the need for specific conditions to exploit firmware.	Moderate exploitability, requiring some level of expertise in exploiting firmware vulnerabilities.	Affects multiple users, particularly those relying on the integrity of the charging system.	Low discoverability, as such vulnerabilities may remain undetected for extended periods.

- **Low:** Scores from 1 to 4 reflect a minimal threat, indicating that the attack is unlikely to cause significant damage, is difficult to reproduce, exploit, or is not easily discoverable.

The DREAD model offers a structured approach to evaluate the impact (as shown in Equation 1) and likelihood (as shown in Equation 2) of different attacks, which can be particularly useful for assessing risks (as shown in Equation 3) in EVCS as mentioned by [113].

$$\text{Impact} = \frac{\sum_{i=1}^{2} X_i}{2} \tag{8.1}$$

$$\text{Likelihood} = \frac{\sum_{i=1}^{3} Y_i}{3} \tag{8.2}$$

$$\text{Risk} = \frac{(\text{Impact} + \text{Likelihood})}{2} \tag{8.3}$$

Where:

$$X_1 = \text{Damage Potential}, \quad X_2 = \text{Affected Users}$$

$$Y_1 = \text{Reproducibility}, \quad Y_2 = \text{exploit-ability}$$

$$Y_3 = \text{Discoverability}$$

Table 8.7 OCPP Attack Assessment: DREAD, Impact, Likelihood, and Risk

Attack	D	R	E	A	D	Impact	Likelihood	Risk
A1	8	9	6	8	9	8.0	8.0	8.0 (H)
A2	9	9	8	9	5	9.0	7.3	8.2 (H)
A3	7	9	8	6	6	6.5	7.7	7.1 (M)
A4	8	8	7	8	4	8.0	6.3	7.2 (M)
A5	8	7	6	8	4	8.0	5.7	6.8 (M)
A6	9	9	8	9	6	9.0	7.7	8.3 (H)
A7	10	7	7	9	4	9.5	6.0	7.8 (M)

Each attack identified in OCPP is evaluated using the DREAD categories are assigned a score. These scores help determine the impact and likelihood of each attack, which aids in identifying the overall risk as shown in (Table 8.7).The cumulative scores help in identifying high-risk vulnerabilities which can cause major disruptions.Below is the analysis of the outcomes achieved from this assessment:

- **A1:** Data Sniffing scores a high risk (8.0), indicating the exposure of sensitive information due to weak encryption. The exploit-ability is moderate (6), but the ease of discovery (9) makes it highly dangerous.

- **A2:** CP Cloning has a high risk (8.2), primarily due to its high damage potential (9) and ease of exploit-ability (8). The attack affects multiple users and can easily be replicated.
- **A3:** Duplicate Connection is rated with medium risk (7.1). Though it is highly reproducible (9), the affected users and damage are comparatively moderate, lowering its overall risk.
- **A4:** Duplicate Booking has medium risk (7.2). The attack is reproducible (8) but less impactful (8), making it less severe than other attacks.
- **A5:** Conflicting Meter Values has a medium risk (6.8). While the impact on billing is significant (8), exploit-ability (6) and discoverability (4) are relatively lower, making it less critical.
- **A6:** Flooding Requests poses a high risk (8.3). It can overwhelm systems easily (9 reproducibility) and has a broad impact (9), especially due to the ease of exploit-ability.
- **A7:** Injection Attack scores a medium risk (7.8), but its high damage potential (10) makes it a serious threat, despite the moderate likelihood due to the complexity of exploiting the attack.

Thus, the highest-risk attacks include Flooding Requests (A6) and CP Cloning (A2), both of which pose significant threats due to their potential for widespread disruption and ease of reproducibility. Conversely, attacks like Conflicting Meter Values (A5) and Duplicate Connection (A3) have lower risks but still warrant attention due to their medium-level impact.

8.6 Mitigate Threat: Upgrade in OCPP

This section addresses the vulnerabilities identified in the OCPP communication protocol based on the attacks analysed using the STRIDE framework and assessed through the DREAD model. The focus is on securing the OCPP communication flow to prevent attacks that exploit weaknesses during an active session. Specifically, we target A3: Duplicate Connection, A4: Duplicate Booking, A5: Conflicting Meter Values, and A6: Flooding Request, as these attacks directly manipulate OCPP messages after a legitimate connection is established.

We have assumed that A1: Data Sniffing and A2: CP Cloning occur before a secure OCPP connection is established, primarily exploiting weak TLS encryption and unauthorized CP registration. Since these attacks compromise initial authentication and identity verification, their mitigation requires enhancements in encryption protocols and device identity

management, which fall outside the scope of this work. Similarly, A7: Firmware Manipulation exploits vulnerabilities in the firmware update process rather than the live OCPP communication session. Addressing firmware security requires additional measures such as code signing, secure update mechanisms, and hardware-level integrity checks, which are not the primary focus of this mitigation strategy. By updating the communication flow within OCPP, the proposed mitigations aim to strengthen session management, ensuring that unauthorized duplicate connections, fraudulent transactions, and excessive message requests are prevented in real time.

8.6.1 OCPP BootNotification Request

In the vulnerability identified in the BootNotification process as shown in (Algorithm 8.1), we assume that the phantom CP_{A_i} has cloned the legitimate CP_A. Since there is no user authentication performed before the BootNotification, anyone with legitimate CP details can send a BootNotification request to the ChgSrv. This allows the phantom CP_{A_i} to send identical BootNotification requests, causing the ChgSrv to accept both connections, leading to cause an A3:Duplicate connection attack. Moreover, the phantom CP_{A_i} can repeatedly send BootNotification requests in quick succession, resulting in an A6: Flooding request attack. The absence of rate limiting on the number of BootNotification requests a ChgSrv can process presents a vulnerability that can be exploited to perform a DoS attack, overwhelming the ChgSrv with excessive requests.

Algorithm 8.1 Vulnerability in BootNotification

Input : $CPModel$, $CPVendor$, $User\_IdTag$, $Reservation\_Id$, CP_Id

Output : Attack: $A3$: Duplicate connection and $A6$: Flooding request

Legitimate Charge Point (CP) $CP_A \rightarrow$ ChgSrv: Sends BootNotification
BootNotification.req($CPModel_A$, $CPVendor_A$, $User\_IdTag_A$, $Reservation\_Id_A$, CP_Id_A)

Phantom Charge Point (CP) $CP_{A_i} \rightarrow$ ChgSrv: Sends BootNotification
BootNotification.req($CPModel_{A_i}$, $CPVendor_{A_i}$, $User\_IdTag_{A_i}$, $Reservation\_Id_{A_i}$, $CP\_Id_{A_i}$)

ChgSrv $\rightarrow$ Legitimate CP: Status = Accepted.
ChgSrv $\rightarrow$ Phantom CP: Status = Accepted.

Result: CP_{A_i} creates duplicate connection.

Flooding Attack:
Phantom $CP_{A_i} \rightarrow$ ChgSrv: Sends multiple BootNotification requests.
for $j \leftarrow 1$ **to** N **do**
    BootNotification.req$_j$($CPModel_{A_i}$, $CPVendor_{A_i}$, $User\_IdTag_{A_i}$, $Reservation\_Id_{A_i}$, $CP\_Id_{A_i}$)

Result: ChgSrv experiences flooding from multiple requests.

In the proposed mitigation to overcome the vulnerability identified in the BootNotification process as shown in (Algorithm 8.2), we assume as an initial step that user login to the CP_A using the email address and password used while booking. Upon successful authentication the app server will generate a JWT Token, which will contain the email address and Userid_tag of the user. This token will then be used by the CP_A to establish a Web Socket connection with the ChgSrv. The ChgSrv parses this token, validates it with the app server and finally if the user is authorised, it will send an OTP to the email address parsed out of this token.The CP_A then sends the BootNotification along with the OTP. The ChgSrv verifies the OTP against the saved session data. If the OTP is valid, the connection is securely established. If the OTP is invalid, the connection is terminated. This step introduces user authentication at the time of the BootNotification, ensuring that only users with a valid reservation can send connection requests to the ChgSrv. Even if a legitimate user's details are compromised and a phantom obtains the booking information, the phantom still cannot establish a connection. This is because the system employs multi-factor authentication by sending an OTP with an expiry, which authenticates the user during the connection process, adding an additional security layer.

Algorithm 8.2 Mitigation for BootNotification

Input : $CPModel$, $CPVendor$, $User\_IdTag$, $Reservation\_Id$, CP_Id

Output : Secure Connection

Send Token:

$CP_A \rightarrow ChgSrv$: Sends token with connection request.

$ChgSrv \rightarrow CP_A$: Checks if CP_A has a current booking.

Validate Token:

if *Token is valid (i.e., CP_A has a valid booking)* **then**

    Send OTP to user and save OTP on server with expiry.

else

    Reject Connection.

Send BootNotification:

$CP_A \rightarrow ChgSrv$: Sends BootNotification with OTP.

BootNotification.req($CPModel_A$, $CPVendor_A$, $User\_IdTag_A$, $Reservation\_Id_A$, CP_Id_A, OTP).

Verify OTP:

$ChgSrv$: Verifies OTP against saved session data.

if *OTP is valid* **then**

    Establish Connection.

else

    End Connection.

Limit Token Requests:

if *CP_A sends multiple token requests* **then**

    Limit requests to M per time interval.

8.6.2 OCPP Start Transaction Request

The vulnerability in the Start Transaction Request as shown in (Algorithm 8.3) arises because both the legitimate CP_A and the phantom CP_{A_i}, assuming the phantom has access to the legitimate booking details (such as the reservation ID), can send valid StartTransaction requests to the ChgSrv. In this scenario, the legitimate CP_A sends a StartTransaction request with the correct details, and the ChgSrv processes and validates the request. However, if the phantom CP_{A_i} also sends a StartTransaction request with the same $Reservation_Id_A$, the ChgSrv cannot differentiate between the legitimate and phantom transactions, as both appear valid. Once both transactions are validated by the ChgSrv, it accepts the transaction

requests from both CP_A and CP_{A_i}.This results in a situation where the ChgSrv accepts multiple transactions under the same reservation, leading to a A4:Duplicate booking attack.Its worth noting that lack of verification beyond basic authentication allows the phantom CP_{A_i} to exploit the system.

Algorithm 8.3 Vulnerability in Start Transaction Request

Input : $CP\_Id, User\_IdTag, Reservation\_Id, Timestamp$

Output : Attack: $A4$: Duplicate booking

Legitimate Transaction Request:
Legit $CP_A \rightarrow$ ChgSrv: Sends StartTransaction request.
StartTransaction.req($CP\_Id_A, User\_IdTag_A, Reservation\_Id_A, Timestamp_A$)

Phantom Transaction Request:
Phantom $CP_{A_i} \rightarrow$ ChgSrv: Sends StartTransaction request.
StartTransaction.req($CP\_Id_{A_i}, User\_IdTag_{A_i}, Reservation\_Id_{A_i}, Timestamp_{A_i}$)

Validate Transaction Request:
ChgSrv validates transaction request.
if *Transaction request is valid* **then**
    ChgSrv $\rightarrow$ Legit CP_A: Status = Accepted.
    if *Duplicate transaction request* **then**
        ChgSrv $\rightarrow$ Phantom CP_{A_i}: Status = Accepted.
else
    ChgSrv $\rightarrow CP_A$: Status = Rejected.

Result: Phantom CP_{A_i} creates duplicate booking.

The mitigation for the vulnerability in the Start Transaction Request as shown in (Algorithm 8.4),focuses on preventing $A4$: Duplicate Booking by implementing checks at the ChgSrv to ensure only one transaction is allowed per reservation at any given time. When a legitimate CP_A sends a StartTransaction request, the ChgSrv first verifies whether the $ReservationId_A$ is already active in another session. If it is, the ChgSrv immediately rejects the new transaction request. This ensures that no duplicate transactions can be initiated using the same reservation ID, effectively preventing the possibility of a phantom CP_{A_i} or even the legitimate CP from starting multiple transactions simultaneously on the same reservation. If the $ReservationId_A$ is not currently in use, the ChgSrv proceeds to assign a new unique $TransactionId$ to the session, ensuring that each transaction has a distinct identifier. The ChgSrv then accepts the request and associates the $ReservationId$ and the session with this new $TransactionId$. By limiting one transaction per reservation and actively monitoring the

status of ongoing sessions, this approach successfully mitigates the risk of duplicate bookings and prevents unauthorized or phantom transactions from exploiting the system.

Algorithm 8.4 Mitigation for Start Transaction Request

Input : CP_Id, $User\_IdTag$, $Reservation\_Id$, $Timestamp$

Output : Mitigation: Preventing $A4$: Duplicate Booking

1 **Send Start Transaction Request:**
$CP_A \rightarrow$ ChgSrv: Sends StartTransaction request.
StartTransaction.req(CP_Id_A, $User\_IdTag_A$, $Reservation\_Id_A$, $Timestamp_A$) **Check Active Reservation:**
ChgSrv checks if $Reservation\_Id$ is already active.
if *Reservation_Id is in an active session* **then**
2 ChgSrv $\rightarrow CP_A$: Status = Rejected.
3 **else**
4 **while** *New Transaction Request* **do**
5 ChgSrv assigns a new Transaction_Id.
ChgSrv $\rightarrow CP_A$: Status = Accepted.
Map ($Reservation\_Id$, $Session\_Id$) to new Transaction_Id.
6 **Result:** Only one transaction per session and per reservation is active at a time.

8.6.3 OCPP Stop Transaction Request

The vulnerability in the Stop Transaction Request as shown in (Algorithm 8.5) arises when both a legitimate CP_A and a phantom CP_{A_i} with the same reservation details send StopTransaction requests to the ChgSrv. Since both the legitimate and phantom CP share the same $ReservationId_A$, the ChgSrv processes the StopTransaction requests from both sources. When calculating the meter value at the end of a transaction, the ChgSrv sums up the meter stop values from all transactions associated with the reservation ID, including those initiated by the phantom CP. As a result, the legitimate user is at risk of being overcharged because the phantom transactions are included in the calculation of the total meter value. This creates a situation where the legitimate user's bill reflects not only their own usage but also the energy usage reported by the phantom CP, leading to unfair billing. This vulnerability, identified as $A5$: Conflicting Meter Values, exploits the lack of differentiation between legitimate and phantom transactions during the meter value calculation.

Algorithm 8.5 Vulnerability in Stop Transaction Request

Input: $Transaction\_Id_A$, $User\_IdTag_A$, $Timestamp_A$, $MeterStop$

Output: Attack: $A5$: Conflicting Meter Values

1 **Legit** $CP_A \rightarrow$ ChgSrv: Sends StopTransaction Request
StopTransaction.req($Transaction\_Id_A$, $User\_IdTag_A$, $Timestamp_A$, $MeterStop$)

2 **Phantom** $CP_{A_i} \rightarrow$ ChgSrv: Sends StopTransaction Request
StopTransaction.req($Transaction\_Id_{A_i}$, $User\_IdTag_{A_i}$, $Timestamp_{A_i}$, $MeterStop$)

3 **foreach** *Transaction with* $Reservation\_Id_A$ **do**

4 $MeterValue = \sum_{j=1}^{N} MeterStop_j$
All transactions related to CP_Id_A, including phantom transactions.

5 **Result**: Legitimate user will be charged extra as phantom transactions are sampled into their bill.

Algorithm 8.6 Mitigation of Stop Transaction Request

Input: $Transaction\_Id_A$, $User\_IdTag_A$, $Timestamp_A$, $MeterStop$, $Reservation\_Id$

Output: Reduced risk of A5: Conflicting Meter Values

1 **Legit** $CP_A \rightarrow$ ChgSrv: Sends StopTransaction Request
StopTransaction.req($Transaction\_Id_A$, $User\_IdTag_A$, $Timestamp_A$, $MeterStop$)
foreach *Transaction with* $Reservation\_Id_A$ **do**

2 **if** *Session is verified* **then**

3 MeterValue = $\sum_{j=1}^{N} MeterStop_j$

4 **Result**: Mitigation reduces the risk of phantom transactions affecting the legitimate user's bill.

The mitigation for the Stop Transaction Request vulnerability shown in (Algorithm 8.6) focuses on ensuring that only verified sessions contribute to the final meter value calculation. By implementing an extra level of authentication during the BootNotification, where an OTP is used to validate the CP and by preventing duplicate bookings in the Start Transaction process, we reduce the risk of phantom transactions. In this mitigation, when a legitimate CP_A sends a StopTransaction request, the ChgSrv processes the request but only sums the $MeterStop$ values from sessions that have been successfully verified. This is achieved by checking each session associated with the $ReservationId_A$ and ensuring that it has passed the OTP authentication during the BootNotification. Only those sessions that are verified

will have their *MeterStop* values included in the final meter reading. As a result, even if a phantom CP attempts to send a StopTransaction request, its session will not be included in the calculation, as it would not have passed the OTP authentication step. This mitigation reduces the risk of a legitimate user being overcharged due to conflicting meter values and ensures that only authenticated transactions are used for billing.

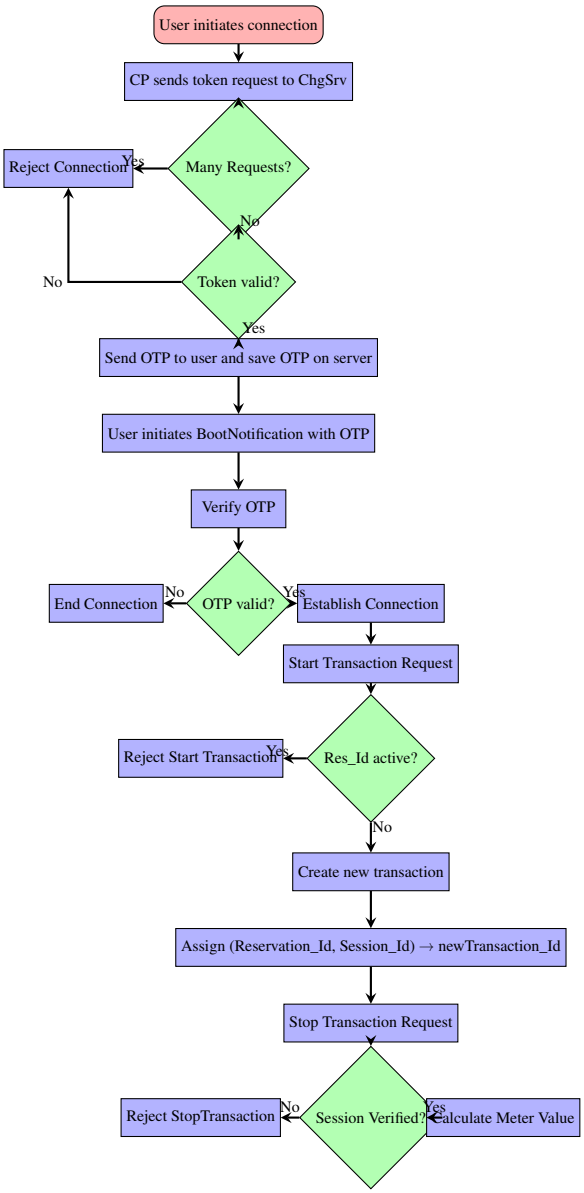


Fig. 8.6 Flowchart to show updated OCPP Mitigation Logic

The flowchart depicted in (Fig 8.6) illustrates the authentication, session management, and message handling processes within OCPP, integrating various mitigation steps to enhance security. The process begins with a user initiating a connection. The CP sends a token request

to the ChgSrv, which checks for rate limits to prevent excessive requests, effectively handling messages. If the token is valid, an OTP is sent to the user for further validation. The user then initiates a BootNotification, and the OTP is verified before the connection is established, thereby addressing the weak authentication vulnerabilities present in OCPP. Once the session is active, a Start Transaction Request can be made. If a reservation is valid, a new transaction is created and linked with both the Reservation ID and the Session ID, contributing to effective session management. In the Stop Transaction phase, session verification is mandatory before calculating meter values. If the session is verified, the summation of meter values occurs for each transaction within that session. Conversely, if the session is not verified, the request is rejected. These steps collectively ensure controlled session management and help prevent unauthorized actions.

8.7 Result Analysis

The result analysis highlights the improvements brought by upgrading the OCPP communication framework. It focuses on two critical aspects: server log updates and a comparative data-centric analysis. The updated server logs showcase enhanced security in authentication, session management, and transaction handling, ensuring a more robust interaction process between the CP and ChgSrv. Additionally, a data-centric comparison between the original and enhanced frameworks evaluates performance metrics such as time, power consumption, and fault detection rates, emphasizing the effectiveness of the upgraded framework in improving system reliability and security.

8.7.1 Environmental Setup

In this setup, we have a Chargebox simulator (web interface) which connects to the system using the OCPP to communicate with a central backend. The simulator represents key operational components of an Electric Vehicle Charging Management System (EVCMS). We are using 2 instances of the Chargebox simulator where Chargebox 1 acts as a legitimate client, while Chargebox 2 serves as a phantom client, designed to exploit protocol vulnerabilities. The Chargebox simulator's user interface is intuitive, facilitating seamless interaction between EV drivers, operators, and the charging infrastructure. Users can configure and manage backend connections through the simulator's authentication features and real-time status indicators. The simulator is integrated into a web-based EVCMS application, enabling smooth communication for tasks such as reservations and charging.

The Chargebox simulator is built in React.js and configured to communicate with the OCPP server using the ocpp-rpc library for Web Socket communication. The system is web-based, with a central application backend built using Node.js, overseeing the data flow between the Chargebox simulators and the OCPP server. The backend incorporates modules for authentication, charging control, cost determination, and reservations. The OCPP schema, which governs communication between the simulators and the backend, is implemented using JSON format and is publicly accessible on GitHub[181]. This repository includes the complete implementation of the OCPP schema in Node.js, along with updates made to the OCPP communication flow.

Given that the current implementation is completely software based, there are no physical charge points involved. However for hosting the system a 2GB RAM and 4vCPU should be enough. The system has a Docker based setup for deployment to cloud. Docker is used for creating images of each system (app server, ocpp server, reservation app and chargebox) and to run these software in sync we use Docker Compose. We've used Node.js with Express library to create a API for interaction between the ocpp server, chargebox and also the reservation app. The OCPP server is also a Node.js based server but uses ocpp-rpc library to setup a web socket based communication channel, this channel is then used by the chargebox to send and receive messages to and from the ocpp-server. For the database, we are self hosting a MongoDB database using their latest image available.

The security enhancement has been designed as part of the OCPP protocol itself to enforce it as a standard. However, the rest of the underlying protocol remains unmodified, making it easier to adopt in existing systems. This approach ensures that the core OCPP protocol remains intact while additional security measures, such as multi-factor authentication (OTP) and session management, are incorporated. Thus, enhancements have been made within the existing OCPP communication framework.

8.7.2 Updated OCPP Server logs

The server logs capture the interactions between the CP and the ChgSrv under the upgraded framework, highlighting key improvements in authentication, session management, and secure transaction handling. Key steps in the process involve the use of BootNotification, Start Transaction, Stop Transaction, and Meter Value Request, all of which have been enhanced with added security measures.

1. **Improved Authentication:** The logs indicate that before initiating any connection, the system authenticates the user with an One-Time Password (OTP), ensuring that only authorized users can establish a connection as shown in (Fig.8.7). This additional veri-

fication step significantly mitigates vulnerabilities such as CP Cloning and Duplicate Booking (A2 and A4). Before the BootNotification message is sent, a token is first transmitted to validate the user's booking. If the booking is valid, the system generates and sends an OTP to authenticate the user. This added layer of security helps to ensure that only legitimate users are able to connect and interact with the ChgSrv.

Fig. 8.7 Server log for Initial connection validation

2. **BootNotification Security Enhancement:** As shown in (Fig 8.8), the BootNotification request and response between the CP and ChgSrv now include an OTP as an additional security parameter.

   This enhancement introduces an extra layer of authentication, which was not part of the original OCPP specification. Traditionally, the BootNotification messages were primarily used to exchange information about the CP's hardware, firmware, and network status with the ChgSrv. However, by integrating the OTP, the new flow ensures that only authorized charging points with a valid booking can successfully complete the connection process. This modified BootNotification message now incorporates an authCode field that holds the OTP value. Upon receiving the BootNotification, the ChgSrv validates the OTP against the server-side generated code. If the OTP is verified, the status of the connection is set to "Accepted", confirming the legitimacy of the CP. This enhancement improves resilience against unauthorized or cloned CPs attempting to impersonate legitimate devices. It helps mitigate attacks like Charge Point Cloning by tying the CP to a specific user booking through a time-sensitive OTP mechanism.

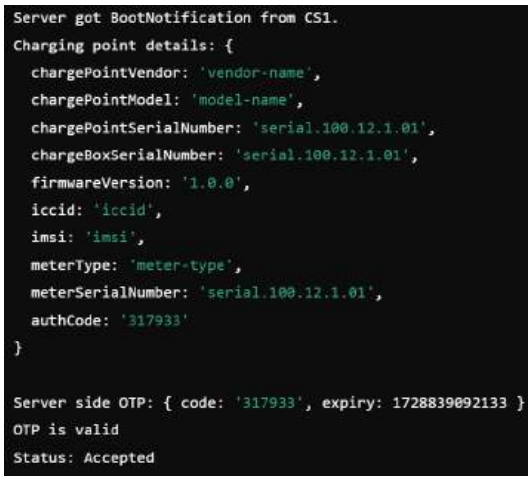


Fig. 8.8 Server log for BootNotification

3. **Improved Session Management:** In the Start Transaction process as shown in (Fig 8.9), the server implements a crucial check to determine if the reservation associated with the incoming request is already active in another transaction. When a CP sends a Start Transaction request, the server first queries its records to see if the specified reservation ID is currently in use. If it finds that the reservation is already linked to an active transaction, it promptly rejects the new Start Transaction request. This validation step is essential in preventing any overlap in transactions for the same reservation, effectively mitigating the risk of duplicate bookings or unauthorized usage. If the reservation ID is not active, the server proceeds to create a new transaction. However, it does not generate a completely separate transaction; instead, it links the new transaction to the same reservation ID and CP ID. This approach ensures that all transactions initiated under a particular reservation are identifiable and manageable within the system, maintaining a clear connection between them. By allowing multiple transactions to reference the same reservation ID while ensuring only one can be active at a time, the system enhances its integrity and security, preventing any potential exploitation or confusion that could arise from multiple active transactions associated with the same reservation.

4. **Transaction Termination Validation:** In the Stop Transaction process as shown in (Fig 8.10), the server again reinforces the integrity of the transaction system by verifying the completion of the transaction initiated under a specific reservation ID. When a Charge Point (CP) sends a Stop Transaction request, the server checks its records to confirm that the transaction ID provided in the request corresponds to an active transaction associated with that reservation ID. This validation is critical as it

Fig. 8.9 Server log for Start Transaction

ensures that only legitimate and ongoing transactions can be terminated, preventing unauthorized attempts to stop transactions that may not exist or are already completed. If the server confirms that the transaction ID is valid and associated with an active reservation, it proceeds to update the transaction status to 'completed.' This involves recording the final meter reading and any other relevant details before finalizing the transaction. The server also ensures that the reservation ID remains linked to this completed transaction, maintaining a clear record of all activities associated with that reservation. By enforcing this check, the system effectively prevents any potential misuse or manipulation of the transaction lifecycle, ensuring that each transaction is concluded accurately and securely.

Fig. 8.10 Server log for Stop Transaction

5. **Continuous Monitoring with Meter Values:** In the Meter Values process as shown in (Fig 8.11), the server plays a crucial role in continuously monitoring and recording the charging session's progress. When a CP sends Meter Values updates, the server validates that the incoming data corresponds to an active transaction associated with a

specific transaction ID and CP ID, which, in turn, is linked to a specific reservation ID. This validation ensures that the reported meter readings are relevant to an ongoing charging session and prevents any potential discrepancies or fraudulent reporting. Upon receiving the Meter Values updates, the server processes the information by checking the transaction ID against its records. If the transaction is valid and associated with the correct reservation ID, the server accepts the updates and logs the current meter readings along with their timestamps. This continuous monitoring allows the server to maintain an accurate record of power consumption throughout the charging session, which is essential for billing and reporting purposes. Moreover, by linking each Meter Values update to a specific transaction and reservation ID, the system can efficiently track and manage multiple charging sessions simultaneously. This linkage enhances the overall integrity of the charging management system, ensuring that users receive accurate data about their energy usage while safeguarding against unauthorized activities.

Fig. 8.11 Server log for Stop Transaction

8.7.3 Comparative Data Evaluation

This section presents an analysis of the charging performance and fault detection metrics by comparing two models: Model 1, based on OCPP, and Model 2, which utilizes an enhanced version of OCPP. The analysis evaluates data from three key tables—reservation data, CP data for Model 1, and CP data for Model 2—to highlight differences in performance, particularly in terms of actual time and power consumption of each reservation and fault rates, between the two models. To generate the data for this analysis, a simulation process was carried out based on 300 reservation entries as shown in (Table 8.8). (Table 8.8) contains reservation information, including the Reservation ID (ResrvId), start time of reservation (ResrvStartTime), end time of reservation (ResrvEndTime), reservation date (ResrvDate), expected time in mins to be taken (ExpectedTime) and expected power in Kw to be consumed (ExpectedPower). This data serves as a reference point for evaluating

each charging transaction and detecting potential conflicts where charging time and power consumption exceeds the reserved time and power window.

Table 8.8 EV charging Reservation Details for Charge Point

ResrvId	ResrvStartTime	ResrvEndTime	ResrvDate	ExpectedTime	ExpectedPower
R001	21:17	23:47	10/03/2024	150	44
R002	15:18	17:18	10/16/2024	120	36
R003	12:49	15:19	10/27/2024	150	44
R004	11:27	12:57	10/17/2024	90	27
R005	09:46	12:16	10/28/2024	150	44
⋮	⋮	⋮	⋮	⋮	⋮
R296	19:49	21:49	10/15/2024	120	36
R297	08:55	10:55	10/17/2024	120	36
R298	02:09	03:09	10/14/2024	60	18
R299	08:22	11:22	10/26/2024	180	53
R300	06:49	07:19	10/07/2024	30	9

Charging behaviours were modelled for both Model 1 and Model 2 using the reservation entries from (Table 8.8) to generate the transaction data found in Tables 8.9 and 8.10. (Table 8.9) presents the CP data for Model 1, where EVs connect using the OCPP protocol. The columns capture the CP Reservation ID (CP_ResrvId), Transaction ID (CP_TransId), the start and end times of each transaction (CP_StartTime and CP_EndTime), the charging date (CP_ChargeDate), and the actual time and power consumed (ActualTime and ActualPower).

Using this data, the total charging time and power consumption per reservation can be calculated. This helps identify conflicting transactions where the charging session exceeds the reserved duration or power consumption, which contributes to the calculation of the fault rate for Model 1. Similarly, (Table 8.10) documents the CP data for Model 2, where an enhanced version of OCPP is used. The table structure is similar to Table 2 but reflects improvements such as enhanced session management, resulting in fewer or no conflicting transactions and a reduced fault rate compared to Model 1. This data is essential for evaluating the impact of these enhancements on overall charging efficiency.

This analysis uses two key metrics to evaluate charging performance: total charging time and total power consumption. These metrics, along with the actual time and actual power from the reservation table, enable the calculation of conflicting transactions. Based on these conflicts, the fault detection rates for the two models can be determined.

Total Charging Time: Total charging time refers to the cumulative time a CP is actively engaged in charging a vehicle during a reservation period. This metric is important for evaluating how long each transaction lasts and is calculated by subtracting the start time

Table 8.9 Charge Point Transaction Data using Model 1

CP_ResrvId	CP_TransId	CP_StartTime	CP_EndTime	CP_ChargeDate	ActualTime	ActualPower
R001	00T11	21:17	23:47	10/03/2024	150.0	45.00
R002	00T21	15:18	17:18	10/16/2024	120.0	36.00
R003	00T31	12:49	15:19	10/27/2024	150.0	45.00
R004	00T41	11:27	11:37	10/17/2024	10.0	3.00
R004	00T42	11:38	12:08	10/17/2024	30.0	9.00
⋮	⋮	⋮	⋮	⋮	⋮	⋮
R296	T2962	20:49	21:49	10/15/2024	59.1	17.73
R297	T2971	08:55	10:55	10/17/2024	120.0	36.00
R298	T2981	02:09	03:09	10/14/2024	60.0	18.00
R299	T2991	08:22	11:22	10/26/2024	180.0	54.00
R300	T3001	06:49	07:19	10/07/2024	30.0	9.00

Table 8.10 Charge Point Transaction Data using Model 2

CP_ResrvId	CP_TransId	CP_StartTime	CP_EndTime	CP_ChargeDate	ActualTime	ActualPower
R001	00T11	21:17	23:47	10/03/2024	150.00	44.00
R002	00T21	15:18	17:18	10/16/2024	120.00	36.00
R003	00T31	12:49	15:19	10/27/2024	150.00	44.00
R004	00T41	11:27	12:57	10/17/2024	90.00	27.00
R005	00T51	09:46	12:16	10/28/2024	150.00	44.00
⋮	⋮	⋮	⋮	⋮	⋮	⋮
R296	T2963	20:16	21:49	10/15/2024	92.04	27.61
R297	T2971	08:55	10:55	10/17/2024	120.00	36.00
R298	T2981	02:09	03:09	10/14/2024	15.00	4.50
R299	T2991	08:22	11:22	10/26/2024	180.00	53.00
R300	T3001	06:49	07:19	10/07/2024	30.00	9.00

from the end time of each transaction. Here m is the total number of transaction for each reservations.It is represented by the formula:

$$\text{Total Charging Time} = \sum_{i=1}^{m} (\text{CP\_EndTime}_i - \text{CP\_StartTime}_i)$$

Total Power Consumption: Total power consumption measures the amount of electrical energy consumed by a vehicle during the charging session. It is calculated as the sum of total power consumed in each transaction. This metric helps in determining how much power is drawn during each session. Here m is the total number of transaction for each reservations. The formula is:

$$\text{Total Power Consumption} = \sum_{i=1}^{m} (\text{ActualPower}_i)$$

Conflicting Transactions: Conflicting transactions occur when the actual charging time or actual power consumption exceeds the values reserved by the user. This metric

identifies discrepancies between expected and actual transaction behaviours. It captures any instances of over-consumption or extended charging beyond the expected limits. A conflicting transaction is defined as:

$$\text{Conflicting Transaction} = \begin{cases} 1 & \text{if Actual Time}_i > \text{Expected Time}_i \text{ or Actual Power}_i > \text{Expected Power}_i \\ 0 & \text{otherwise} \end{cases}$$

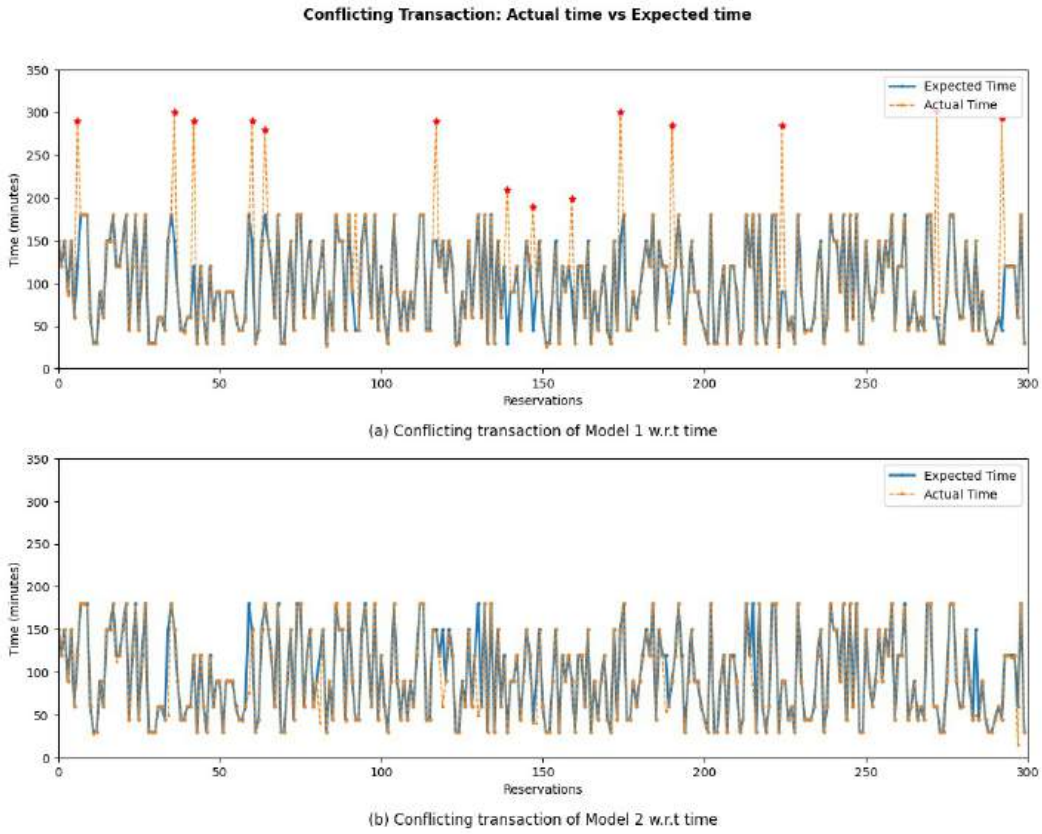


Fig. 8.12 (a) Conflicting Transaction of model 1 w.r.t time (b) Conflicting Transaction of model 2 w.r.t time

The (Fig.8.12) compares the actual charging time against the expected charging time across reservations for both models, to find the conflicting transactions.Model 1 as shown in (Fig.8.12 a), shows multiple red stars, which indicate instances of conflicting transactions where the actual charging time exceeds the expected time. These stars suggest suspicious activity, such as phantom users sharing the same CP as the legitimate user. When a phantom user utilizes the CP alongside the actual user, the cumulative actual time for these transactions exceeds the expected time, resulting in conflicting transactions. This issue points to a vulnerability in Model 1.Model 2 as shown in (Fig.8.12 b), in contrast, has no red stars, meaning no transaction exceeded the expected time. This suggests that Model 2 effectively

controls access to the CP, preventing phantom users the access. Model 2 demonstrates better reliability in restricting usage strictly to the authorized user.

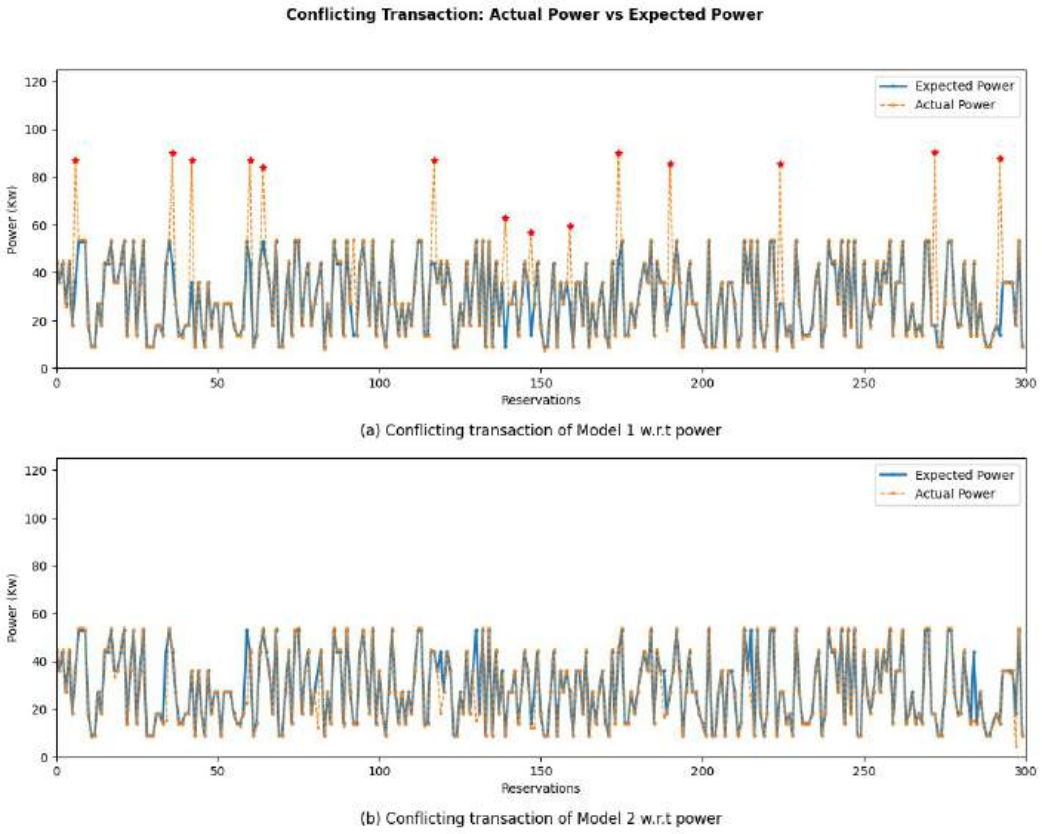


Fig. 8.13 (a) Conflicting Transaction of model 1 w.r.t power (b) Conflicting Transaction of model 2 w.r.t power

The (Fig.8.13) illustrates the actual power consumption versus expected power consumption for both models. Model 1 as shown in (Fig.8.13 a), again shows red stars, which represent instances where actual power consumption exceeds the expected levels. The excessive power consumption could also result from the phantom user's unauthorized usage of the CP. When both the legitimate user and the phantom user draw power simultaneously, the cumulative power usage exceeds the expected power levels, leading to conflicting transactions. This points to a vulnerability in Model 1 where phantom usage increases the overall power drawn from the CP. Model 2 as shown in (Fig.8.13 b), does not exhibit any red stars, meaning actual power consumption stays within expected limits for each reservation. This further supports Model 2's effectiveness in blocking phantom users, maintaining power consumption levels within the expected range.

Fault Detection Rate: The fault detection rate quantifies the proportion of conflicting transactions relative to the total number of reservations. It measures the reliability of the

charging system by identifying how often the actual charging behaviour deviates from the expected behaviour. Here, z=1 is a constant accounting for external factors or additional error margins that may influence the fault rate.The formula is:

$$\text{Fault Detection Rate} = \frac{\text{Number of Conflicting Transactions} + z}{\text{Total Number of Reservations}}$$

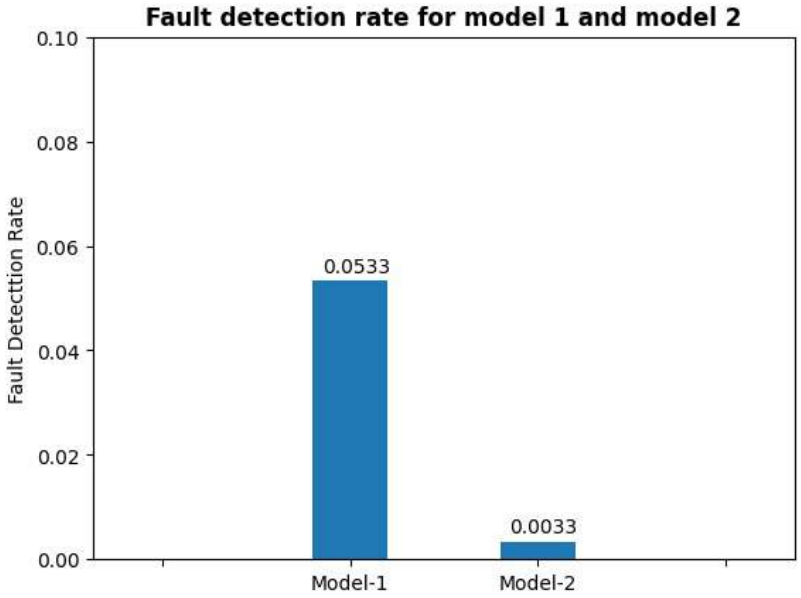


Fig. 8.14 Fault detection rate

The fault detection rate shown in (Fig.8.14) quantifies the proportion of conflicting transactions relative to total reservations. Model 1 has a significantly higher fault detection rate calculated as 0.0533, suggesting frequent occurrences of conflicting transactions. The high rate reflects the model's vulnerability to phantom user activity, which disrupts expected charging behaviour and results in cumulative time and power exceeding authorized limits. Model 2 has a much lower fault detection rate calculated as 0.0033, indicating very few conflicting transactions. This low rate implies that Model 2 effectively mitigates phantom usage, maintaining charging behaviour in line with the expected values and providing greater reliability.

8.8 State-of-the-Art Comparison

OCPP serves as a critical communication standard between CP and CS. However, various security vulnerabilities associated with this protocol pose significant risks. Our proposed

work builds upon an analysis of these vulnerabilities using the DREAD model, which identified authentication, session management, and message handling as high-risk factors. The (Table 9.1) presents a comparative analysis of our proposed work against other state-of-the-art research in the field. By evaluating the effectiveness of authentication mechanisms, session management practices, and message handling protocols, we aim to highlight the strengths and weaknesses of existing solutions in addressing the vulnerabilities inherent in OCPP communication. Additionally, the experimental analysis conducted by researchers is crucial for assessing the practical outcomes of these vulnerabilities and verifying the effectiveness of their proposed mitigations.

Table 8.11 State of the Art Analysis for OCPP Communication Vulnerability

Paper	Vulnerability	Attack Type	Auth.	Session Mgmt.	Msg Handling	Exp. Analysis
Proposed	OCPP comm.	DoS, MitM, Conn. Hijacking	Yes	Yes	Yes	Yes
[104]	OCPP comm.	MitM	Yes	No	Yes	Yes
[13]	OCPP comm.	DoS, MitM	Yes	Yes	Yes	Yes
[10]	OCPP comm.	DoS, MitM	Yes	No	Yes	No
[166]	OCPP comm.	DoS, MITM, Code Inj.	Yes	No	Yes	Yes
[21]	OCPP comm.	DoS, MITM, Firmware Theft	Yes	No	Yes	Yes
[19]	OCPP CP	Malformed Req./Resp. Inj.	No	No	Yes	Yes

Rubio et al. [104] focus on enhancing security in EV charging systems through secure key exchange mechanisms that strengthen authentication processes, ensuring only authorized parties can communicate. While they effectively implement secret sharing techniques to safeguard message integrity and confidentiality, session management is not addressed, leaving a critical gap. Their evaluation is based on simulations, which test the robustness of their solutions against potential threats, including DoS and MitM attacks. Alcaraz et al. [13] emphasize robust authentication achieved through strong cryptography and secure protocols, mitigating unauthorized access and impersonation risks. They examine session management, specifically heartbeat intervals and protocols governing message exchanges, which helps maintain secure interactions. Their research also addresses message handling vulnerabilities, particularly message tampering and spoofing, with experimental validation conducted in

a simulated OCPP environment using ocppjs and tools like Ettercap to test various attack scenarios. Garofalaki et al. [10] explore vulnerabilities in OCPP, highlighting issues like ARP spoofing and RKE cloning, and propose enhancements using TLS and blockchain technologies for authentication and billing processes. However, their work does not address session management, focusing primarily on existing studies to improve OCPP security based on a literature review rather than experimental validation.

Johnson et al. [166] investigate vulnerabilities in OCPP, pointing out unencrypted web socket communications and potential remote code execution. They discuss attack vectors such as DoS and MitM while emphasizing the need for secure communications, noting weaknesses in authentication due to reliance on unencrypted methods. Although session management is not covered, they provide proof-of-concept exploits demonstrated in a controlled environment, suggesting enhancements like secure shell tunnels for better protection. Sarieddine et al. [21] identify six zero-day vulnerabilities in EVCS and OCPP backend communications, discussing MitM, DoS attacks, firmware theft, and data poisoning. They underscore security weaknesses in authentication mechanisms and the importance of message handling, but do not specifically address session management. Their findings are supported by a developed test bed that showcases the feasibility of attacks against the power grid via compromised EVCSs, providing critical insights into EV charging ecosystem security.Gebauer et al. [19] discuss vulnerabilities in OCPP charge points, particularly the injection of malformed requests/responses, which can be exploited by malicious central systems. They highlight the importance of secure message handling but do not explicitly address authentication or session management. Their study employs a simulated OCPP charge point for monitoring and testing, with plans for further penetration testing on real charge points. This comparative framework demonstrates the varying approaches and findings across different research works, contributing to a deeper understanding of the vulnerabilities within OCPP communication.

8.9 Chapter Summary

This chapter has identified and thoroughly analysed the critical security vulnerabilities in OCPP, a widely adopted communication protocol for EVCS. These vulnerabilities, such as weak session management, inadequate authentication, and poor message handling, expose the EV charging infrastructure to attacks like DoS, unauthorized access, and transaction tampering. Through the application of the STRIDE and DREAD frameworks, the vulnerabilities were mapped to potential threats, and their severity was assessed. To mitigate these risks, the paper proposed enhancements such as introducing multi-factor authentication via OTP

during the BootNotification process, improving session management to prevent duplicate connections and bookings, and incorporating measures to validate sessions and meter values. Additionally, a data-centric analysis was conducted to compare the charging performance and fault detection metrics between two models: one using OCPP and the other an enhanced version of OCPP. Simulation-based evaluation of 300 reservation entries highlighted significant improvements in performance and fault rates with the enhanced protocol. These mitigation strategies and analyses aim to strengthen OCPP's security and ensure a more robust and reliable EV charging infrastructure as the demand for electric mobility continues to grow.The comparative analysis of existing research clearly indicates that while several studies have made significant contributions to securing OCPP communications, critical gaps remain, particularly in areas such as session management and comprehensive attack mitigation strategies. Our proposed work addresses these gaps by leveraging robust authentication techniques, improving session continuity, and improving message integrity protections. Furthermore, by incorporating experimental validation, we ensure that the proposed solutions are grounded in practical, real-world scenarios.

Chapter 9

Conclusion and Future Work

9.1 Thesis Summary

The primary goal of this research was to develop a comprehensive and secure architecture for EV charging systems, with a focus on enhancing cyber security resilience. The project aimed to address vulnerabilities in the OCPP, improve the overall security of charging networks, and streamline real-time communication between EVs, charging stations, and servers. To achieve this goal, the research undertook the following steps:

1. A thorough analysis of existing EV charging protocols and security frameworks was conducted, identifying critical vulnerabilities.

2. Designed and implemented a cloud-ready, centralized EV charging infrastructure based on OCPP, incorporating robust security enhancements.

3. The centralized infrastructure was scaled to a distributed EV charging infrastructure, applying security measures against potential threats.

4. A series of experiments were performed to test the security resilience of the infrastructure, including assessments of attack vulnerabilities and performance improvements over existing systems.

5. Security enhancements to the OCPP protocol were proposed, with a comparison between the standard and enhanced versions in terms of attack mitigation and system robustness.

9.2 Key Outputs

This thesis research contributes to the development of a safe and secure EV charging solution for the industry partner **JMVL Ltd.**. Through experimental and data-driven validation, we assessed the cyber risks in the EV charging network. Subsequently, an **EVCMS framework** was developed, where the optimization of the charging reservation process was achieved through effective management of charging schedules, improved resource allocation, and enhanced user experience, all of which contributed to reduced charging costs. Building on this foundation, the **H-EVCMS framework** was introduced to further optimize resource allocation across multiple stations, improving operational efficiency and enabling load balancing.

As EV infrastructure expands, securing communication between charge points (CPs) and charge stations (CSs) becomes increasingly critical. The widely used OCPP protocol faces significant security vulnerabilities, exposing systems to DoS and MitM attacks that undermine the confidentiality, integrity, and availability of communication during the charging process. To address these vulnerabilities, both **DoS and MitM attacks** were performed on the H-EVCMS framework to test its security. As a result, the **framework was enhanced** to become more robust against these attacks, incorporating stronger authentication and improved session management. These updates have bolstered the security and resilience of the EV charging system. The outcome of this work provides a more secure, efficient, and reliable EV charging infrastructure, capable of meeting the growing demand for EV charging while mitigating emerging cyber threats.

This section reflects on the objectives outlined earlier in the research, detailing the progress and outcomes achieved with respect to each objective. The corresponding chapters in which these objectives were addressed are also highlighted.

1. **Analyse EVCS for Cyber security Vulnerabilities:** To identify security flaws within the EVCS infrastructure and communication protocols like OCPP, ensuring the protection of data and system functionality.

   Reflection: This objective was addressed in **Chapter 2** and **Chapter 3**, where a comprehensive analysis of the EVCS was conducted. These chapters focused on identifying critical cybersecurity vulnerabilities within the EVCS infrastructure, particularly in communication protocols such as OCPP. The analysis revealed key threats such as weak authentication and session management, setting the foundation for further enhancements in the security of EV charging systems.

2. **Design and Develop a cloud ready EVCMS Framework:** To create a robust and secure EVCMS framework that addresses the cybersecurity needs of EVCS that could be deployed on cloud platform.

   Reflection: The development of the EVCMS framework was completed in **Chapter 4**. This chapter presented the design and implementation of a robust and secure framework for managing EV charging sessions. The EVCMS framework provided an effective solution for optimizing resource allocation, securing the reservation process, and addressing the cybersecurity needs of the EVCS.

3. **Scale EVCMS Framework to a Distributed Cloud Service:** Enhance the scalability of the EVCMS framework to support a growing number of EV charging stations globally while maintaining robust security. Although the framework is designed to be cloud-ready, it has currently been deployed locally and has not yet been implemented in a cloud environment.

   Reflection: This objective was addressed in **Chapter 5**, where the framework's architecture was developed to be cloud-ready, supporting future deployment on distributed cloud infrastructure. While currently deployed locally, the design assumes that once integrated with real-time cloud services, the system will be able to handle a larger number of EV charging stations efficiently by leveraging improved scalability, load balancing, and resource allocation capabilities inherent to cloud environments.

4. **Test, Validate, and Generate Experimental Benchmarking Performance Data:** To validate the performance and security of the EVCMS framework, generating benchmark data to assess its operational efficiency and resilience.

   Reflection: Testing, validation, and benchmarking were addressed across **Chapter 6**, **Chapter 7**, and **Chapter 8**. **Chapter 6** and **Chapter 7** focused on the testing of the EVCMS framework's performance and security, including assessments of DoS and MitM attack resilience. In **Chapter 8**, OCPP was enhanced to address the identified vulnerabilities, and its performance was benchmarked against data-centric approach and the current state-of-the-art solutions. The updated version of OCPP was compared with the standard OCPP to evaluate improvements in security, efficiency, and resilience against cyber threats, demonstrating the effectiveness of the proposed enhancements.

5. **Integrate Smart and Secure EV Charging into JMVL's Existing Application:** JMVL already had an application that identified EV charging stations based on their location. They required an integrated solution that would allow users to book charging slots and access real-time charging features while maintaining security.

Reflection: This objective was fulfilled through the development of a web application that provides optimized charging plans, load-balanced charging station management, and real-time charging features using OCPP. The existing application, which serves 2,000 users, was enhanced with secure booking and real-time charging functionalities. Additionally, OCPP was upgraded to ensure secure communication between users and charging stations, strengthening the overall cybersecurity of the EV charging infrastructure.

These objectives were successfully met, resulting in a secure, scalable, and efficient EV charging infrastructure that addresses the growing demand for electric vehicle charging while enhancing resilience against cyber threats and contributing to the development of smart city ecosystems.

9.3 Limitations

- The research is limited to older versions of OCPP as its open source and widely adopted by charging Infrastructure.
- The analysis of DoS and MitM attacks focuses on specific scenarios and may not generalize to all potential attack vectors.
- The hybrid framework assumes a certain level of interoperability between distributed charging stations, which may not be universally feasible.
- Implementation and validation are conducted in simulated environments, and results may vary in real-world conditions.
- The proposed upgrade in OCPP framework requires updates to the existing hardware of charging stations to accommodate protocol changes, which limits its immediate applicability in the current EVCS infrastructure.

9.4 Directions for Further Research

While this research has significantly advanced the security and efficiency of EV charging infrastructure, several areas remain for further exploration and improvement. The proposed system can be expanded in multiple directions to address evolving challenges and enhance its applicability. Future research may focus on the following:

- **Cloud Deployment for Real-Time Data Handling:** Although the current Hybrid EVCMS framework is cloud-ready, future work involves deploying it on a distributed cloud infrastructure to support real-time processing and management of charging data. This deployment is expected to enhance scalability, load balancing, and responsiveness, enabling efficient handling of growing EV charging demands across multiple locations.
- **Advancing Security Protocols:** Although OCPP has been enhanced, some version of OCPP still relies on TLS 1.2, which has known vulnerabilities. In contrast, updated OCPP version incorporates upgraded encryption algorithms. To balance compatibility and security, the encryption mechanisms in OCPP could be updated to match the advancements in later version of OCPP, allowing users to benefit from improved security without migrating to the newer protocol.
- **Testing and Securing Autonomous EV Charging Systems:** Future work will focus on testing the autonomous EV charging system to identify vulnerabilities in its charging process. Building on the improvements made to secure the communication channel in OCPP, this effort will involve developing and implementing advanced security measures to address the specific challenges of autonomous EV charging.
- **Integration of AI and Machine Learning:** Leveraging AI and machine learning algorithms can significantly enhance anomaly detection in EV charging systems across charging stations. Developing intelligent algorithms capable of adapting to traffic patterns and user behaviour would strengthen security and enable robust monitoring and threat mitigation. Similarly, these technologies can be applied to autonomous EV charging systems, providing advanced, adaptive security measures tailored to their unique operational requirements.

By exploring these areas, future research can continue to improve the resilience, efficiency, and scalability of EV charging infrastructures, supporting the growing global transition to electric vehicles while safeguarding against emerging cybersecurity threats.

References

[1] "The uk govt strategy, "road to zero" by 2050," Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/739460/road-to-zero.pdf, accessed on 21st January 2023.

[2] IEA. Global ev outlook 2020. Available at: https://www.iea.org/reports/global-ev-outlook-2020. Accessed on 22nd January 2023.

[3] U. CCC, "Net zero: The uk's contribution to stopping global warming," *UK Climate Change Committee, London*, 2019.

[4] M. A. Sayed, R. Atallah, C. Assi, and M. Debbabi, "Electric vehicle attack impact on power grid operation," *International Journal of Electrical Power & Energy Systems*, vol. 137, p. 107784, 2022.

[5] E. Remotely Controlled, "Home chargers—the threats and vulnerabilities," 2018.

[6] Schneider Electric. (2018) EVLink parking. Available at: https://us-cert.cisa.gov/ics/advisories/ICSA-19-031-01. Accessed on 2nd March 2023.

[7] S. Orcioni, L. Buccolini, A. Ricci, and M. Conti, "Electric vehicles charging reservation based on ocpp," in *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*. IEEE, 2018, pp. 1–6.

[8] S. Hsaini, M. Ghogho, and M. E. H. Charaf, "An ocpp-based approach for electric vehicle charging management," *Energies*, vol. 15, no. 18, p. 6735, 2022.

[9] O. C. Alliance, "Open charge point protocol 1.6," *Open Charge Alliance Publications*, 2016, published in 2015.

[10] Z. Garofalaki, D. Kosmanos, S. Moschoyiannis, D. Kallergis, and C. Douligeris, "Electric vehicle charging: A survey on the security issues and challenges of the open charge point protocol (ocpp)," *IEEE Communications Surveys & Tutorials*, 2022.

[11] M. E. Hansen, "Implementation and test of the open charge point protocol in an autonomous charger for electric vehicles," *DTU Wind-M-0756*, 2024.

[12] Monta. (2023) Upgrade to ocpp 2.0.1: The key to advancing the ev charging infrastructure. Last updated: 23 May, 2023; accessed December 4, 2024. [Online]. Available: https://monta.com/uk/blog/upgrade-to-ocpp-2-0-1/

[13] C. Alcaraz, J. Lopez, and S. Wolthusen, "Ocpp protocol: Security threats and challenges," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2452–2459, 2017.

[14] S. Orcioni and M. Conti, "Ev smart charging with advance reservation extension to the ocpp standard," *Energies*, vol. 13, no. 12, p. 3263, 2020.

[15] M. Elkasrawy, S. O. Abdellatif, G. A. Ebrahim, and H. A. Ghali, "Real-time optimization in electric vehicle stations using artificial neural networks," *Electrical Engineering*, vol. 105, no. 1, pp. 79–89, 2023.

[16] S. Hamdare, D. J. Brown, Y. Cao, M. Aljaidi, S. Kumar, R. Alanazi, M. Jugran, P. Vyas, and O. Kaiwartya, "A novel charging management and security framework for the electric vehicle (ev) ecosystem," *World Electric Vehicle Journal*, vol. 15, no. 9, 2024. [Online]. Available: https://www.mdpi.com/2032-6653/15/9/392

[17] S. Hamdare, D. J. Brown, Y. Cao, M. Aljaidi, O. Kaiwartya, R. Yadav, P. Vyas, and M. Jugran, "Ev charging management and security for multi-charging stations environment," *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 807–824, 2024.

[18] I. García-Magariño, R. Lacuesta, M. Rajarajan, and J. Lloret, "Security in networks of unmanned aerial vehicles for surveillance with an agent-based approach inspired by the principles of blockchain," *Ad Hoc Networks*, vol. 86, pp. 72–82, 2019.

[19] L. Gebauer, H. Trsek, and G. Lukas, "Evil steve: An approach to simplify penetration testing of ocpp charge points," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2022, pp. 1–4.

[20] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on intelligent transportation systems*, vol. 16, no. 2, pp. 993–1006, 2014.

[21] K. Sarieddine, M. A. Sayed, S. Torabi, R. Attallah, D. Jafarigiv, C. Assi, and M. Debbabi, "Uncovering covert attacks on ev charging infrastructure: How ocpp backend vulnerabilities could compromise your system," *ACM*, 2024.

[22] T. Karthik, A. Brown, S. Awwad, D. McCoy, R. Bielawski, C. Mott, S. Lauzon, A. Weimerskirch, and J. Cappos, "Uptane: Securing software updates for automobiles," in *International Conference on Embedded Security in Car*, 2016, pp. 1–11.

[23] R. Currie, "Hacking the can bus: basic manipulation of a modern automobile through can bus reverse engineering," *SANS Institute*, 2017.

[24] C. Carryl, M. Ilyas, I. Mahgoub, and M. Rathod, "The pev security challenges to the smart grid: Analysis of threats and mitigation strategies," in *2013 International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 2013, pp. 300–305.

[25] J. Johnson, T. Berg, B. Anderson, and B. Wright, "Review of electric vehicle charger cybersecurity vulnerabilities, potential impacts, and defenses," *Energies*, vol. 15, no. 11, p. 3931, 2022.

[26] O. Apata, P. N. Bokoro, and G. Sharma, "The risks and challenges of electric vehicle integration into smart cities," *Energies*, vol. 16, no. 14, p. 5274, 2023.

[27] (2020) Elaadnl open dataset. (Accessed on 30th March 2023). [Online]. Available: https://platform.elaad.io/analyses/ElaadNL_opendata.php

[28] S. Hamdare, O. Kaiwartya, M. Aljaidi, M. Jugran, Y. Cao, S. Kumar, M. Mahmud, D. Brown, and J. Lloret, "Cybersecurity risk analysis of electric vehicles charging stations," *Sensors*, vol. 23, no. 15, p. 6716, 2023.

[29] I. G. E. Outlook, "Accelerating ambitions despite the pandemic," *International Energy Agency: Paris, France*, 2021.

[30] S. S. Rangarajan, S. P. Sunddararaj, A. Sudhakar, C. K. Shiva, U. Subramaniam, E. R. Collins, and T. Senjyu, "Lithium-ion batteries—the crux of electric vehicles with opportunities and challenges," *Clean Technologies*, vol. 4, no. 4, pp. 908–930, 2022.

[31] S. Jayashree, K. Malarvizhi, and R. Pradeep, "Impact of hybrid electric vehicle penetration and its challenges on distribution system," *Advances in Natural and Applied Sciences*, vol. 11, no. 5 SI, pp. 101–109, 2017.

[32] S. Nayak and A. K. Bohre, "Status of electric vehicles charging methods," *International Journal of Engineering, Science and Technology*, vol. 14, no. 3, pp. 132–143, 2022.

[33] A. V. Shrivastav, S. S. Khan, R. K. Gupta, P. R. Ekshinge, and N. Parmeshwar, "Electric vehicle charging station (case study on infrastructure of ev charging station)," *J. Emerg. Technol. Innov. Res*, vol. 7, pp. 2017–2033, 2020.

[34] S. Bhattacharjee, S. Batool, C. Nandi, and U. Pakdeetrakulwong, "Investigating electric vehicle (ev) charging station locations for agartala, india," *Proceedings of the 11th NPRU National Academic Conference*, 2017.

[35] H. Lee and A. Clark, "Charging the future: Challenges and opportunities for electric vehicle adoption," *HKS Working Paper No. RWP18-026*, 2018.

[36] U.S. Department of Energy. (2021) Charging Levels and Connector Types. Available at: https://www.energy.gov/eere/electricvehicles/charging-levels-and-connector-types. Energy.gov. Accessed on 13th February 2023.

[37] U.S.Department of Energy. (2021) DC Fast Charging. https://afdc.energy.gov/vehicles/electric_dc_fast_charge.html. Alternative Fuels Data Center. Accessed on 15th February 2023.

[38] S. Holzer. (2022) Public vs. at-home ev charging stations: The pros and cons. Available at: https://www.bonney.com/2022/03/the-pros-and-cons-of-public-vs-at-home-ev-charging-stations/#:~:text=A%20public%20charging%20station%20refers,electricity%20bills%20associated%20with%20it. Bonney. Accessed on 29th January 2023.

[39] U.S. Department of Energy. (2021) U.S. Public and Private Electric Vehicle Charging Infrastructure. Available at: https://afdc.energy.gov/data/10327. Alternative Fuels Data Center. Accessed on 31st January 2023.

[40] N. Deb, R. Singh, R. R. Brooks, and K. Bai, "A review of extremely fast charging stations for electric vehicles," *Energies*, vol. 14, no. 22, p. 7566, 2021.

[41] M. Amjad, M. Farooq-i Azam, Q. Ni, M. Dong, and E. A. Ansari, "Wireless charging systems for electric vehicles," *Renewable and Sustainable Energy Reviews*, vol. 167, p. 112730, 2022.

[42] M. S. Hossain Lipu, M. S. Miah, S. Ansari, S. B. Wali, T. Jamal, R. M. Elavarasan, S. Kumar, M. Naushad Ali, M. R. Sarker, A. Aljanad *et al.*, "Smart battery management technology in electric vehicle applications: analytical and technical assessment toward emerging future directions," *Batteries*, vol. 8, no. 11, p. 219, 2022.

[43] K. Sheng, M. Dibaj, and M. Akrami, "Analysing the cost-effectiveness of charging stations for electric vehicles in the uk's rural areas," *World Electric Vehicle Journal*, vol. 12, no. 4, p. 232, 2021.

[44] W. Leal Filho, I. R. Abubakar, R. Kotter, T. S. Grindsted, A.-L. Balogun, A. L. Salvia, Y. A. Aina, and F. Wolf, "Framing electric mobility for urban sustainability in a circular economy context: An overview of the literature," *Sustainability*, vol. 13, no. 14, p. 7786, 2021.

[45] T. Mazhar, R. N. Asif, M. A. Malik, M. A. Nadeem, I. Haq, M. Iqbal, M. Kamran, and S. Ashraf, "Electric vehicle charging system in the smart grid using different machine learning methods," *Sustainability*, vol. 15, no. 3, p. 2603, 2023.

[46] S. S. Shuvo and Y. Yilmaz, "Predictive maintenance for increasing ev charging load in distribution power system," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2020, pp. 1–6.

[47] S. R. Etesami, W. Saad, N. B. Mandayam, and H. V. Poor, "Smart routing of electric vehicles for load balancing in smart grids," *Automatica*, vol. 120, p. 109148, 2020.

[48] A. Bourass, S. Cherkaoui, and L. Khoukhi, "Secure communication scheme for electric vehicles in the smart grid," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–5.

[49] A. Bahrami, "Ev charging definitions, modes, levels, communication protocols and applied standards," *Changes*, vol. 1, pp. 1–10, 2020.

[50] K. Bao, H. Valev, M. Wagner, and H. Schmeck, "A threat analysis of the vehicle-to-grid charging protocol iso 15118," *Computer Science-Research and Development*, vol. 33, no. 1, pp. 3–12, 2018.

[51] P. Van Den Bossche, "Iec 61851-1: Electric vehicle conductive charging system-part 1: General requirements," in *2*. Iec, 2010, pp. 1–99.

[52] H. S. Das, M. M. Rahman, S. Li, and C. Tan, "Electric vehicles standards, charging infrastructure, and impact on grid integration: A technological review," *Renewable and Sustainable Energy Reviews*, vol. 120, p. 109618, 2020.

[53] K. Dimitriadou, N. Rigogiannis, S. Fountoukidis, F. Kotarela, A. Kyritsis, and N. Papanikolaou, "Current trends in electric vehicle charging infrastructure; opportunities and challenges in wireless charging integration," *Energies*, vol. 16, no. 4, p. 2057, 2023.

[54] V. Gowri and P. Sivraj, "A centralized management system software framework to aid in ev charging," in *2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*. IEEE, 2021, pp. 703–707.

[55] A. I. Aygun and S. Kamalasadan, "Centralized charging approach to manage electric vehicle fleets for balanced grid," in *2022 IEEE International Conference on Power Electronics, Smart Grid, and Renewable Energy (PESGRE)*. IEEE, 2022, pp. 1–6.

[56] Y. Cao, O. Kaiwartya, Y. Zhuang, N. Ahmad, Y. Sun, and J. Lloret, "A decentralized deadline-driven electric vehicle charging recommendation," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3410–3421, 2018.

[57] M. Moschella, M. A. A. Murad, E. Crisostomi, and F. Milano, "Decentralized charging of plug-in electric vehicles and impact on transmission system dynamics," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1772–1781, 2020.

[58] I. Aravena, S. J. Chapin, and C. Ponce, "Decentralized failure-tolerant optimization of electric vehicle charging," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4068–4078, 2021.

[59] A. Paudel, S. A. Hussain, R. Sadiq, H. Zareipour, and K. Hewage, "Decentralized cooperative approach for electric vehicle charging," *Journal of Cleaner Production*, vol. 364, p. 132590, 2022.

[60] M. S. Mastoi, S. Zhuang, H. M. Munir, M. Haris, M. Hassan, M. Usman, S. S. H. Bukhari, and J.-S. Ro, "An in-depth analysis of electric vehicle charging station infrastructure, policy implications, and future trends," *Energy Reports*, vol. 8, pp. 11 504–11 529, 2022.

[61] D. A. Chekired and L. Khoukhi, "Smart grid solution for charging and discharging services based on cloud computing scheduling," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3312–3321, 2017.

[62] Y. Hua, D. Zhao, X. Wang, and X. Li, "Joint infrastructure planning and fleet management for one-way electric car sharing under time-varying uncertain demand," *Transportation Research Part B: Methodological*, vol. 128, pp. 185–206, 2019.

[63] K. Zhou, L. Cheng, L. Wen, X. Lu, and T. Ding, "A coordinated charging scheduling method for electric vehicles considering different charging demands," *Energy*, vol. 213, p. 118882, 2020.

[64] D. Said, S. Cherkaoui, and L. Khoukhi, "Queuing model for evs charging at public supply stations," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2013, pp. 65–70.

[65] C. B. Jones, W. Vining, M. Lave, T. Haines, C. Neuman, J. Bennett, and D. R. Scoffield, "Impact of electric vehicle customer response to time-of-use rates on distribution power grids," *Energy Reports*, vol. 8, pp. 8225–8235, 2022.

[66] H. Lin, J. Dang, H. Zheng, L. Yao, Q. Yan, S. Yang, H. Guo, and A. Anvari-Moghaddam, "Two-stage electric vehicle charging optimization model considering dynamic virtual price-based demand response and a hierarchical non-cooperative game," *Sustainable Cities and Society*, p. 104715, 2023.

[67] A. Colmenar-Santos, A.-M. Muñoz-Gómez, E. Rosales-Asensio, and Á. López-Rey, "Electric vehicle charging strategy to support renewable energy sources in europe 2050 low-carbon scenario," *Energy*, vol. 183, pp. 61–74, 2019.

[68] X. Lyu, T. Liu, X. Liu, C. He, L. Nan, and H. Zeng, "Low-carbon robust economic dispatch of park-level integrated energy system considering price-based demand response and vehicle-to-grid," *Energy*, vol. 263, p. 125739, 2023.

[69] E. A. ElGhanam, M. S. Hassan, and A. H. Osman, "Deployment optimization of dynamic wireless electric vehicle charging systems: A review," in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. IEEE, 2020, pp. 1–7.

[70] K. Harnett, B. Harris, D. Chin, G. Watson *et al.*, "Doe/dhs/dot volpe technical meeting on electric vehicle and charging station cybersecurity report," John A. Volpe National Transportation Systems Center (US), Tech. Rep., 2018.

[71] V. Stykas. (2022) Ev charging points hacked to show explicit material - cities today. (Accessed on 19th March 2023). [Online]. Available: https://cities-today.com/ev-charging-points-hacked-to-show-explicit-material/

[72] M. Dibaei, X. Zheng, K. Jiang, R. Abbas, S. Liu, Y. Zhang, Y. Xiang, and S. Yu, "Attacks and defences on intelligent connected vehicles: A survey," *Digital Communications and Networks*, vol. 6, no. 4, pp. 399–421, 2020.

[73] H. Booth, "Draft nistir 8138, vulnerability description ontology (vdo)," *National Institute of Standards and Technology (NIST), Tech. Rep*, 2016.

[74] Idaho National Laboratory, "Cyber assessment report of level 2 ac powered electric vehicle supply equipment," avt.inl.gov, Tech. Rep., 2018, (Accessed on 15th February 2023). [Online]. Available: https://avt.inl.gov/sites/default/files/pdf/reports/Level2EVSECyberReport.pdf

[75] S. Dmitry, "Chargepoint home security research," 2018.

[76] N. Bhusal, M. Gautam, and M. Benidris, "Cybersecurity of electric vehicle smart charging management systems," in *2020 52nd North American Power Symposium (NAPS)*. IEEE, 2021, pp. 1–6.

[77] S. Fu, Z. Zhang, Y. Jiang, J. Chen, X. Peng, and W. Zhao, "An automatic rf-emf radiated immunity test system for electricity meters in power monitoring sensor networks." *Ad Hoc Sens. Wirel. Networks*, vol. 50, no. 1-4, pp. 173–192, 2021.

[78] R. Baker and I. Martinovic, "Losing the car keys: Wireless {PHY-Layer} insecurity in {EV} charging," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 407–424.

[79] B. R. Anderson and J. B. Johnson, "Securing vehicle charging infrastructure." Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2021.

[80] S. Poyyamani Sunddararaj, S. S. Rangarajan, S. Nallusamy, E. R. Collins, and T. Senjyu, "A brief survey on important interconnection standards for photovoltaic systems and electric vehicles," *World Electric Vehicle Journal*, vol. 12, no. 3, p. 117, 2021.

[81] J. Foster. (2022) Evsec automates cybersecurity for ev ecosystem, electric vehicle charging & infrastructure. (Accessed on 12th February 2023). [Online]. Available: https://www.evcandi.com/products/evsec-automates-cybersecurity-ev-ecosystem

[82] S. Shirvani, "Electric vehicles and charging infrastructure security," *Scholar Research Repository UNM Libraries*, 2023.

[83] S. Acharya, Y. Dvorkin, H. Pandžić, and R. Karri, "Cybersecurity of smart electric vehicle charging: A power grid perspective," *IEEE access*, vol. 8, pp. 214 434–214 453, 2020.

[84] K. Sarieddine, M. A. Sayed, S. Torabi, R. Atallah, and C. Assi, "Investigating the security of ev charging mobile applications as an attack surface," *ACM Transactions on Cyber-Physical Systems*, vol. 7, no. 4, pp. 1–28, 2023.

[85] C. Carter, I. Onunkwo, P. Cordeiro, and J. Johnson, "Cyber security assessment of distributed energy resources," in *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*. IEEE, 2017, pp. 2135–2140.

[86] T. V. Phan and M. Park, "Efficient distributed denial-of-service attack defense in sdn-based cloud," *IEEE Access*, vol. 7, pp. 18 701–18 714, 2019.

[87] J. Qin, M. Li, L. Shi, and X. Yu, "Optimal denial-of-service attack scheduling with energy constraint over packet-dropping networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 6, pp. 1648–1663, 2017.

[88] S. Dey and M. Khanra, "Cybersecurity of plug-in electric vehicles: Cyberattack detection during charging," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 1, pp. 478–487, 2020.

[89] K. Sarieddine, M. A. Sayed, C. Assi, R. Atallah, S. Torabi, J. Khoury, M. S. Pour, and E. Bou-Harb, "Ev charging infrastructure discovery to contextualize its deployment security," *IEEE Transactions on Network and Service Management*, 2023.

[90] Z. El-Rewini, K. Sadatsharan, D. F. Selvaraj, S. J. Plathottam, and P. Ranganathan, "Cybersecurity challenges in vehicular communications," *Vehicular Communications*, vol. 23, p. 100214, 2020.

[91] Z. Muhammad, Z. Anwar, B. Saleem, and J. Shahid, "Emerging cybersecurity and privacy threats to electric vehicles and their impact on human and environmental sustainability," *Energies*, vol. 16, no. 3, p. 1113, 2023.

[92] C. Jewers, "Russian motorways electric vehicle chargers are hacked to display message supporting ukraine," *Mailonline*, 2022, online, Accessed: 12 May 2024. [Online]. Available: https://shorturl.at/irvAX

[93] BBC News, "Isle of wight: Council's electric vehicle chargers hacked to show porn site," *BBC News*, 2022, online, Accessed: 14 May 2024. [Online]. Available: https://www.bbc.com/news/uk-england-hampshire-61006816

[94] M. Akuchie, "Hacked electrify america charger exposes major cybersecurity risk," *ScreenRant*, January 2023, online, Accessed: 23 May 2024. [Online]. Available: https://screenrant.com/electrify-america-hacked-charger-cybersecurity-risk/

[95] F. S. Dantas Silva, E. Silva, E. P. Neto, M. Lemos, A. J. Venancio Neto, and F. Esposito, "A taxonomy of ddos attack mitigation approaches featured by sdn technologies in iot scenarios," *Sensors*, vol. 20, no. 11, p. 3078, 2020.

[96] B. Sieklik, R. Macfarlane, and W. J. Buchanan, "Evaluation of tftp ddos amplification attack," *computers & security*, vol. 57, pp. 67–92, 2016.

[97] F. Yihunie, E. Abdelfattah, and A. Odeh, "Analysis of ping of death dos and ddos attacks," in *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. IEEE, 2018, pp. 1–4.

[98] S. I. Walad, M. Zarlis, and M. I. S. Efendi, "Analysis of denial of service attack on web security systems," in *Journal of Physics: Conference Series*, vol. 1811, no. 1. IOP Publishing, 2021, p. 012127.

[99] H. S. Salunkhe, S. Jadhav, and V. Bhosale, "Analysis and review of tcp syn flood attack on network with its detection and performance metrics," *International Journal of Engineering Research & Technology (IJERT)*, vol. 6, no. 01, pp. 2278–0181, 2017.

[100] N. Muraleedharan and B. Janet, "Behaviour analysis of http based slow denial of service attack," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2017, pp. 1851–1856.

[101] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," *International Journal of Distributed Sensor Networks*, vol. 13, no. 12, p. 1550147717741463, 2017.

[102] L. Xuefeng and Z. Wei, "Risks of cyber threats and developing robust security protocols within electric vehicle charging infrastructure," *Journal of Sustainable Urban Futures*, vol. 12, no. 12, pp. 16–31, 2022.

[103] G. S. Morrison, "Threats and mitigation of ddos cyberattacks against the us power grid via ev charging," Master's thesis, Wright State University, 2018.

[104] J. Rubio, C. Alcaraz, and J. Lopez, "Addressing security in ocpp: Protection against man-in-the-middle attacks," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, February 2018, pp. 1–5.

[105] M. Conti, D. Donadel, R. Poovendran, and F. Turrin, "Evexchange: A relay attack on electric vehicle charging system," in *European Symposium on Research in Computer Security*. Cham: Springer International Publishing, September 2022, pp. 488–508.

[106] T. Nasr, S. Torabi, E. Bou-Harb, C. Fachkha, and C. Assi, "Chargeprint: A framework for internet-scale discovery and security analysis of ev charging management systems," in *NDSS*, 2023.

[107] B. Vaidya and H. Mouftah, "Deployment of secure ev charging system using open charge point protocol," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, June 2018, pp. 922–927.

[108] L. R. Saposnik, "Hijacking ev charge points to cause dos," https://www.saiflow.com/blog/hijacking-chargers-identifier-to-cause-dos/, 2024, accessed: June 27, 2024.

[109] A. Friedland, "Security and privacy in the current e-mobility charging infrastructure," in *Proceedings of the DeepSec*, Vienna, Austria, 2016, p. 31.

[110] A. Morosan and F. Pop, "Ocpp security-neural network for detecting malicious traffic," in *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, September 2017, pp. 190–195.

[111] M. Girdhar, J. Hong, H. Lee, and T.-J. Song, "Hidden markov models-based anomaly correlations for the cyber-physical security of ev charging stations," *IEEE Transactions on Smart Grid*, vol. 13, no. 5, pp. 3903–3914, 2021.

[112] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "Stride-based threat modeling for cyber-physical systems," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, 2017, pp. 1–6.

[113] G. Kavallieratos and S. Katsikas, "Managing cyber security risks of the cyber-enabled ship," *Journal of Marine Science and Engineering*, vol. 8, no. 10, p. 768, 2020.

[114] A. Chandwani, S. Dey, and A. Mallik, "Cybersecurity of onboard charging systems for electric vehicles—review, challenges and countermeasures," *IEEE access*, vol. 8, pp. 226 982–226 998, 2020.

[115] Y. Cao, T. Wang, X. Zhang, O. Kaiwartya, M. H. Eiza, and G. Putrus, "Toward anycasting-driven reservation system for electric vehicle battery switch service," *IEEE Systems Journal*, vol. 13, no. 1, pp. 906–917, 2018.

[116] Y. Cao, H. Song, O. Kaiwartya, B. Zhou, Y. Zhuang, Y. Cao, and X. Zhang, "Mobile edge computing for big-data-enabled electric vehicle charging," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 150–156, 2018.

[117] V. S. Kumaran and G. Ananthi, "Artificial noise aided polar code with optimal jamming position for physical layer security in mondrian loss integrated rayleigh wireless relay channel." *Adhoc & Sensor Wireless Networks*, vol. 51, 2022.

[118] S. B. KR *et al.*, "An enhanced optimal fair exchange protocol for enhancing security and authenticity in a three-tier wban architecture." *Adhoc & Sensor Wireless Networks*, vol. 47, 2020.

[119] A. Rego, A. Canovas, J. M. Jiménez, and J. Lloret, "An intelligent system for video surveillance in iot environments," *IEEE Access*, vol. 6, pp. 31 580–31 598, 2018.

[120] Bhawana, S. Kumar, R. S. Rathore, M. Mahmud, O. Kaiwartya, and J. Lloret, "Best—blockchain-enabled secure and trusted public emergency services for smart cities environment," *Sensors*, vol. 22, no. 15, p. 5733, 2022.

[121] R. S. Rathore, C. Hewage, O. Kaiwartya, and J. Lloret, "In-vehicle communication cyber security: challenges and solutions," *Sensors*, vol. 22, no. 17, p. 6679, 2022.

[122] M. Ghaderzadeh and M. Aria, "Management of covid-19 detection using artificial intelligence in 2020 pandemic," in *proceedings of the 5th international conference on medical and health informatics*, 2021, pp. 32–38.

[123] M. Gheisari, F. Ebrahimzadeh, M. Rahimi, M. Moazzamigodarzi, Y. Liu, P. K. Dutta Pramanik, M. A. Heravi, A. Mehbodniya, M. Ghaderzadeh, M. R. Feylizadeh *et al.*, "Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey," *CAAI Transactions on Intelligence Technology*, vol. 8, no. 3, pp. 581–606, 2023.

[124] Q. Chen and K. A. Folly, "Application of artificial intelligence for ev charging and discharging scheduling and dynamic pricing: A review," *Energies*, vol. 16, no. 1, p. 146, 2022.

[125] M. Muratori, M. Alexander, D. Arent, M. Bazilian, P. Cazzola, E. M. Dede, J. Farrell, C. Gearhart, D. Greene, A. Jenn *et al.*, "The rise of electric vehicles—2020 status and future expectations," *Progress in Energy*, vol. 3, no. 2, p. 022002, 2021.

[126] W. Liu, T. Placke, and K. Chau, "Overview of batteries and battery management for electric vehicles," *Energy Reports*, vol. 8, pp. 4058–4084, 2022.

[127] T. Lieven and B. Hügler, "Did electric vehicle sales skyrocket due to increased environmental awareness while total vehicle sales declined during covid-19?" *Sustainability*, vol. 13, no. 24, p. 13839, 2021.

[128] T. Chen, X.-P. Zhang, J. Wang, J. Li, C. Wu, M. Hu, and H. Bian, "A review on electric vehicle charging infrastructure development in the uk," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 2, pp. 193–205, 2020.

[129] V. N. S. S. Chimakurthi, "An optimal cloud based electric vehicle charging system," *Asia Pacific Journal of Energy and Environment*, vol. 8, no. 2, pp. 29–38, 2021.

[130] Y. Cao, S. Liu, Z. He, X. Dai, X. Xie, R. Wang, and S. Yu, "Electric vehicle charging reservation under preemptive service," in *2019 1st International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2019, pp. 1–6.

[131] R. Flocea, A. Hîncu, A. Robu, S. Senocico, A. Traciu, B. M. Remus, M. S. Răboacă, and C. Filote, "Electric vehicle smart charging reservation algorithm," *Sensors*, vol. 22, no. 8, p. 2834, 2022.

[132] M. M. Hussain, M. S. Alam, and M. S. Beg, "Plug-in electric vehicle to cloud data analytics for charging management," *IJET*, vol. 9, no. 3, pp. 361–370, 2017.

[133] "React - a javascript library for building user interfaces," Available: https://legacy.reactjs.org/, 2023, accessed on 2023-07-18.

[134] I. Kainu, "Optimization in react. js: Methods, tools, and techniques to improve performance of modern web applications," B.S. thesis, 2022.

[135] H. Shah and T. R. Soomro, "Node. js challenges in implementation," *Global Journal of Computer Science and Technology*, vol. 17, no. 2, pp. 73–83, 2017.

[136] C.-A. Staicu, M. Pradel, and B. Livshits, "Understanding and automatically preventing injection attacks on node. js," in *Network and Distributed System Security Symposium (NDSS)*, 2018.

[137] A. Mardan and A. Mardan, "Intro to mongodb," *Full Stack JavaScript: Learn Backbone. js, Node. js, and MongoDB*, pp. 239–256, 2018.

[138] Mikuso, "Ocpp-rpc: A node.js client and server implementation of the wamp-like rpc-over-websocket system defined in the ocpp-j protocols," Available: https://github.com/mikuso/ocpp-rpc, 2023, accessed on 2023-07-15.

[139] Victormunoz, "Ocpp-1.6-chargebox-simulator: A simple chargepoint simulator, working with ocpp 1.6," Available: https://github.com/victormunoz/OCPP-1.6-Chargebox-Simulator, 2023, accessed on 2023-08-03.

[140] D. Devendra, S. Mante, D. Niteesh, and A. M. Hussain, "Electric vehicle charging station using open charge point protocol (ocpp) and onem2m platform for enhanced functionality," in *TENCON 2021-2021 IEEE Region 10 Conference (TENCON)*. IEEE, 2021, pp. 01–05.

[141] T. V. Pruthvi, N. Dutta, P. B. Bobba, and B. S. Vasudeva, "Implementation of ocpp protocol for electric vehicle applications," in *E3S Web of Conferences*, vol. 87. EDP Sciences, 2019, p. 01008.

[142] M. Saqib, M. M. Hussain, M. S. Alam, M. S. Beg, and A. Sawant, "Smart electric vehicle charging through cloud monitoring and management," *Technology and Economics of Smart Grids and Sustainable Energy*, vol. 2, pp. 1–10, 2017.

[143] P. Patil, "The future of electric vehicles: A comprehensive review of technological advancements, market trends, and environmental impacts," *Journal of Artificial Intelligence and Machine Learning in Management*, vol. 4, no. 1, pp. 56–68, 2020.

[144] N. Zart, "Batteries keep on getting cheaper," *Dosegljivo: https://cleantechnica.com/2017/12/11/batteries-keep-getting-cheaper/[Dostopano: 8.11. 2018]*, 2017.

[145] D. Said, M. Elloumi, and L. Khoukhi, "Cyber-attack on p2p energy transaction between connected electric vehicles: A false data injection detection based machine learning model," *IEEE Access*, vol. 10, pp. 63 640–63 647, 2022.

[146] G. Napoli, A. Polimeni, S. Micari, L. Andaloro, and V. Antonucci, "Optimal allocation of electric vehicle charging stations in a highway network: Part 1. methodology and test application," *Journal of Energy Storage*, vol. 27, p. 101102, 2020.

[147] S. Abdullah-Al-Nahid, T. A. Khan, M. A. Taseen, and T. Aziz, "A consumer-friendly electric vehicle charging scheme for residential consumers," in *2020 International Conference on Smart Grids and Energy Systems (SGES)*. IEEE, 2020, pp. 893–897.

[148] S. Aghajan-Eshkevari, S. Azad, M. Nazari-Heris, M. T. Ameli, and S. Asadi, "Charging and discharging of electric vehicles in power systems: An updated and detailed review of methods, control structures, objectives, and optimization methodologies," *Sustainability*, vol. 14, no. 4, p. 2137, 2022.

[149] Victormunoz, "Ocpp-1.6-chargebox-simulator: A simple chargepoint simulator, working with ocpp 1.6," Available: https://github.com/victormunoz/OCPP-1.6-Chargebox-Simulator, 2023, accessed on 2023-08-03.

[150] T. Zhang, X. Chen, B. Wu, M. Dedeoglu, J. Zhang, and L. Trajkovic, "Stochastic modeling and analysis of public electric vehicle fleet charging station operations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9252–9265, 2021.

[151] X. Chen, H. Wang, F. Wu, Y. Wu, M. C. González, and J. Zhang, "Multimicrogrid load balancing through ev charging networks," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5019–5026, 2021.

[152] X. Chen, T. Zhang, W. Ye, Z. Wang, and H. H.-C. Iu, "Blockchain-based electric vehicle incentive system for renewable energy consumption," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 396–400, 2020.

[153] O. A. Lawal and J. Teh, "Dynamic line rating forecasting algorithm for a secure power system network," *Expert Systems with Applications*, vol. 219, p. 119635, 2023.

[154] ——, "A framework for modelling the reliability of dynamic line rating operations in a cyber–physical power system network," *Sustainable Energy, Grids and Networks*, vol. 35, p. 101140, 2023.

[155] T. Song and J. Teh, "Coordinated integration of wind energy in microgrids: A dual strategy approach leveraging dynamic thermal line rating and electric vehicle scheduling," *Sustainable Energy, Grids and Networks*, p. 101299, 2024.

[156] Y. Su, J. Teh, and C. Chen, "Optimal dispatching for ac/dc hybrid distribution systems with electric vehicles: Application of cloud-edge-device cooperation," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[157] Y. Su, J. Teh, and W. Liu, "Hierarchical and distributed energy management framework for ac/dc hybrid distribution systems with massive dispatchable resources," *Electric Power Systems Research*, vol. 225, p. 109856, 2023.

[158] Y. Su and J. Teh, "Two-stage optimal dispatching of ac/dc hybrid active distribution systems considering network flexibility," *Journal of Modern Power Systems and Clean Energy*, vol. 11, no. 1, pp. 52–65, 2022.

[159] A. Novak and A. Ivanov, "Network security vulnerabilities in smart vehicle-to-grid systems identifying threats and proposing robust countermeasures," *Journal of Artificial Intelligence and Machine Learning in Management*, vol. 7, no. 1, pp. 48–80, 2023.

[160] T. Nasr, S. Torabi, E. Bou-Harb, C. Fachkha, and C. Assi, "Power jacking your station: In-depth security analysis of electric vehicle charging station management systems," *Computers & Security*, vol. 112, p. 102511, 2022.

[161] M. Ammarah Cheema, A. Hafiz, M. Khan, F. Ahmad, and M. Anwar, "Prevention techniques against distributed denial of service attacks in heterogeneous: A systematic review. security and communication," *Networks*, 2022.

[162] S. Mendon, "Slow dos attack: Why it is dangerous and how to detect using a siem," *Cyber Security & Information Security Services*, 2016.

[163] N. Vishnu, R. S. Batth, and G. Singh, "Denial of service: types, techniques, defence mechanisms and safe guards," in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*. IEEE, 2019, pp. 695–700.

[164] X. de Carne de Carnavalet and P. van Oorschot, "A survey and analysis of tls interception mechanisms and motivations: Exploring how end-to-end tls is made "end-to-me" for web traffic," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–40, 2023.

[165] D. Elmo, G. Fragkos, J. Johnson, K. Rohde, S. Salinas, and J. Zhang, "Disrupting ev charging sessions and gaining remote code execution with dos, mitm, and code injection exploits using ocpp 1.6," in *2023 Resilience Week (RWS)*. IEEE, November 2023, pp. 1–8.

[166] J. Johnson, D. E. II, G. Fragkos, J. Zhang, K. W. Rohde, and S. C. Salinas, "Disrupting ev charging sessions and gaining remote code execution with dos, mitm, and code injection exploits using ocpp 1.6," Idaho National Laboratory (INL), Idaho Falls, ID (United States), Tech. Rep., 2023.

[167] S. Hussain, A. Kamal, S. Ahmad, G. Rasool, and S. Iqbal, "Threat modelling methodologies: a survey," *Sci. Int.(Lahore)*, vol. 26, no. 4, pp. 1607–1609, 2014.

[168] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *2010 IEEE symposium on security and privacy*. IEEE, 2010, pp. 447–462.

[169] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and privacy vulnerabilities of {In-Car} wireless networks: A tire pressure monitoring system case study," in *19th USENIX Security Symposium (USENIX Security 10)*, 2010.

[170] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *20th USENIX security symposium (USENIX Security 11)*, 2011.

[171] S. Jafarnejad, L. Codeca, W. Bronzi, R. Frank, and T. Engel, "A car hacking experiment: When connectivity meets vulnerability," in *2015 IEEE globecom workshops (GC Wkshps)*. IEEE, 2015, pp. 1–6.

[172] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlidès, "Lock it and still lose it—on the ({In) Security} of automotive remote keyless entry systems," in *25th USENIX security symposium (USENIX Security 16)*, 2016.

[173] S. Mazloom, M. Rezaeirad, A. Hunter, and D. McCoy, "A security analysis of an {in-vehicle} infotainment and app platform," in *10th USENIX workshop on offensive technologies (WOOT 16)*, 2016.

[174] Q. Luo and J. Liu, "Wireless telematics systems in emerging intelligent and connected vehicles: Threats and solutions," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 113–119, 2018.

[175] C. Jouvray, G. Pellischek, and M. Tiguercha, "Impact of a smart grid to the electric vehicle ecosystem from a privacy and security perspective," *World Electric Vehicle Journal*, vol. 6, no. 4, pp. 1115–1124, 2013.

[176] Circontrol, "Circarlife," https://uscert.cisa.gov/ics/advisories/ICSA-18-305-03, 2019, [Online; accessed 18-sep-2024].

[177] P. Van Aubel, E. Poll, and J. Rijneveld, "Non-repudiation and end-to-end security for electric-vehicle charging," in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*. IEEE, 2019, pp. 1–5.

[178] J. Antoun, M. E. Kabir, B. Moussa, R. Atallah, and C. Assi, "A detailed security assessment of the ev charging ecosystem," *IEEE Network*, vol. 34, no. 3, pp. 200–207, 2020.

[179] S. Acharya, Y. Dvorkin, and R. Karri, "Public plug-in electric vehicles+ grid data: Is a new cyberattack vector viable?" *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5099–5113, 2020.

[180] C. Alcaraz, J. Cumplido, and A. Trivino, "Ocpp in the spotlight: threats and countermeasures for electric vehicle charging infrastructures 4.0," *International Journal of Information Security*, vol. 22, no. 5, pp. 1395–1421, 2023.

[181] S. Hamdare, "ocpp-rpc: Ocpp schema implementation using json in node.js," *GitHub*, 2025, online. [Online]. Available: https://github.com/SHamdare/ocpp-rpc

Appendix A

State Life Cycle of RPC used for OCPP connection

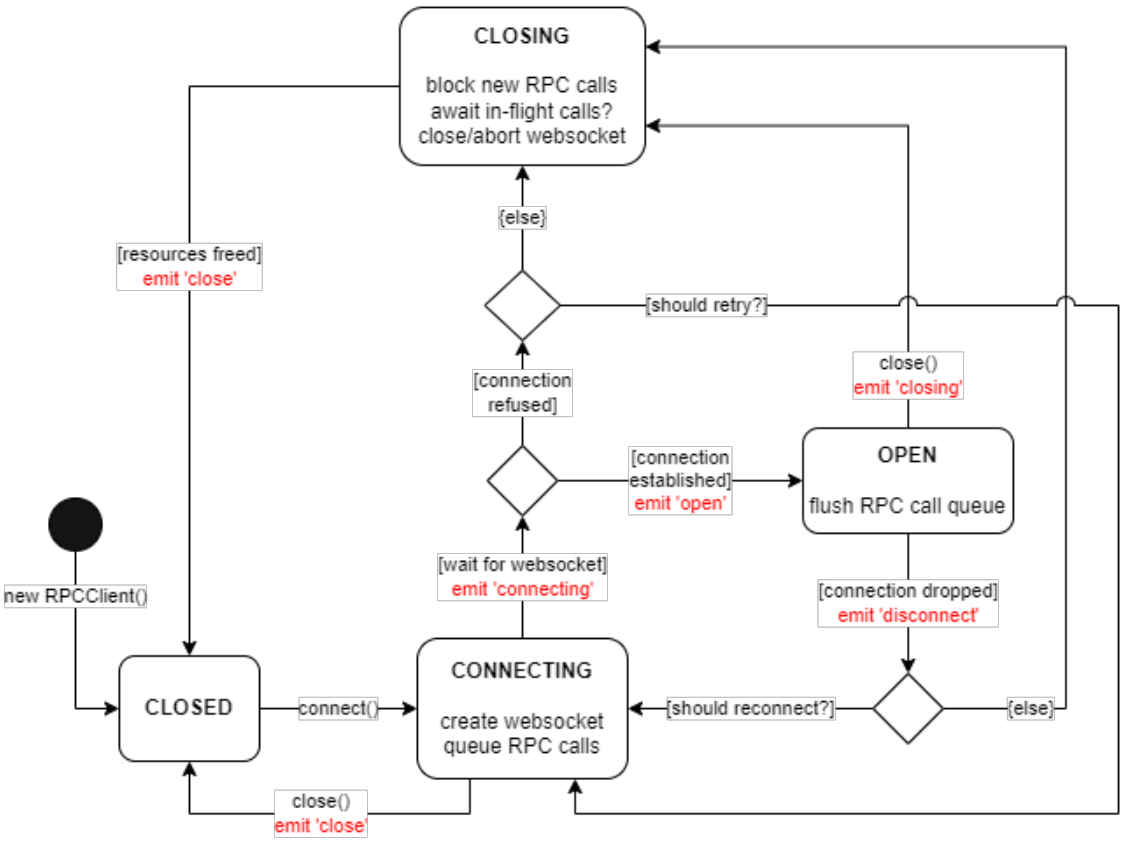


Fig. A.1 State Life cycle of OCPP

Appendix B

Json Schema for OCPP

The following JSON schema defines the structure for all the messages used in the system, such as Boot Notification, Connect, Start, Stop, Meter Request, and others.

```
[
    {
        "$id": "urn:Authorize.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "idTag": {
                "type": "string",
                "maxLength": 20
            }
        },
        "required": [
            "idTag"
        ],
        "type": "object"
    },
    {
        "$id": "urn:Authorize.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "idTagInfo": {
                "type": "object",
```

```
            "properties": {
                "expiryDate": {
                    "type": "string",
                    "format": "date-time"
                },
                "parentIdTag": {
                    "type": "string",
                    "maxLength": 20
                },
                "status": {
                    "type": "string",
                    "enum": [
                        "Accepted",
                        "Blocked",
                        "Expired",
                        "Invalid",
                        "ConcurrentTx"
                    ]
                }
            },
            "additionalProperties": false,
            "required": [
                "status"
            ]
        }
    },
    "required": [
        "idTagInfo"
    ],
    "type": "object"
},
{
    "$id": "urn:BootNotification.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
```

```
"chargePointVendor": {
    "type": "string",
    "maxLength": 20
},
"chargePointModel": {
    "type": "string",
    "maxLength": 20
},
"chargePointSerialNumber": {
    "type": "string",
    "maxLength": 25
},
"chargeBoxSerialNumber": {
    "type": "string",
    "maxLength": 25
},
"firmwareVersion": {
    "type": "string",
    "maxLength": 50
},
"iccid": {
    "type": "string",
    "maxLength": 20
},
"imsi": {
    "type": "string",
    "maxLength": 20
},
"meterType": {
    "type": "string",
    "maxLength": 25
},
"meterSerialNumber": {
    "type": "string",
    "maxLength": 25
},
```

```
            "authCode": {
                "type": "string"
            }
        },
        "required": [
            "chargePointVendor",
            "chargePointModel",
            "authCode"
        ],
        "type": "object"
    },
    {
        "$id": "urn:BootNotification.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "status": {
                "type": "string",
                "enum": [
                    "Accepted",
                    "Pending",
                    "Rejected"
                ]
            },
            "currentTime": {
                "type": "string",
                "format": "date-time"
            },
            "interval": {
                "type": "integer"
            }
        },
        "required": [
            "status",
            "currentTime",
            "interval"
```

```
    ],
    "type": "object"
},
{
    "$id": "urn:CancelReservation.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "reservationId": {
            "type": "integer"
        }
    },
    "required": [
        "reservationId"
    ],
    "type": "object"
},
{
    "$id": "urn:CancelReservation.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "Accepted",
                "Rejected"
            ]
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
{
```

```
    "$id": "urn:CertificateSigned.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "certificateChain": {
            "type": "string",
            "maxLength": 10000
        }
    },
    "required": [
        "certificateChain"
    ],
    "type": "object"
},
{
    "$id": "urn:CertificateSigned.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "definitions": {
        "CertificateSignedStatusEnumType": {
            "type": "string",
            "enum": [
                "Accepted",
                "Rejected"
            ]
        }
    },
    "properties": {
        "status": {
            "$ref": "#/definitions/CertificateSignedStatusEnumType"
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
```

```
},
{
    "$id": "urn:ChangeAvailability.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "connectorId": {
            "type": "integer"
        },
        "type": {
            "type": "string",
            "enum": [
                "Inoperative",
                "Operative"
            ]
        }
    },
    "required": [
        "connectorId",
        "type"
    ],
    "type": "object"
},
{
    "$id": "urn:ChangeAvailability.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "Accepted",
                "Rejected",
                "Scheduled"
            ]
        }
```

```
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:ChangeConfiguration.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "key": {
                "type": "string",
                "maxLength": 50
            },
            "value": {
                "type": "string",
                "maxLength": 500
            }
        },
        "required": [
            "key",
            "value"
        ],
        "type": "object"
    },
    {
        "$id": "urn:ChangeConfiguration.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "status": {
                "type": "string",
                "enum": [
                    "Accepted",
                    "Rejected",
```

```
                "RebootRequired",
                "NotSupported"
            ]
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
{
    "$id": "urn:ClearCache.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {},
    "type": "object"
},
{
    "$id": "urn:ClearCache.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "Accepted",
                "Rejected"
            ]
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
{
```

```
        "$id": "urn:ClearChargingProfile.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "id": {
                "type": "integer"
            },
            "connectorId": {
                "type": "integer"
            },
            "chargingProfilePurpose": {
                "type": "string",
                "enum": [
                    "ChargePointMaxProfile",
                    "TxDefaultProfile",
                    "TxProfile"
                ]
            },
            "stackLevel": {
                "type": "integer"
            }
        },
        "type": "object"
    },
    {
        "$id": "urn:ClearChargingProfile.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "status": {
                "type": "string",
                "enum": [
                    "Accepted",
                    "Unknown"
                ]
            }
```

```
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:DataTransfer.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "vendorId": {
                "type": "string",
                "maxLength": 255
            },
            "messageId": {
                "type": "string",
                "maxLength": 50
            },
            "data": {
                "type": "string"
            }
        },
        "required": [
            "vendorId"
        ],
        "type": "object"
    },
    {
        "$id": "urn:DataTransfer.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "status": {
                "type": "string",
                "enum": [
```

```
                "Accepted",
                "Rejected",
                "UnknownMessageId",
                "UnknownVendorId"
            ]
        },
        "data": {
            "type": "string"
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
{
    "$id": "urn:DeleteCertificate.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "definitions": {
        "HashAlgorithmEnumType": {
            "type": "string",
            "enum": [
                "SHA256",
                "SHA384",
                "SHA512"
            ]
        },
        "CertificateHashDataType": {
            "type": "object",
            "additionalProperties": false,
            "properties": {
                "hashAlgorithm": {
                    "$ref": "#/definitions/HashAlgorithmEnumType"
                },
                "issuerNameHash": {
```

```
                    "type": "string",
                    "maxLength": 128
                },
                "issuerKeyHash": {
                    "type": "string",
                    "maxLength": 128
                },
                "serialNumber": {
                    "type": "string",
                    "maxLength": 40
                }
            },
            "required": [
                "hashAlgorithm",
                "issuerNameHash",
                "issuerKeyHash",
                "serialNumber"
            ]
        }
    },
    "properties": {
        "certificateHashData": {
            "$ref": "#/definitions/CertificateHashDataType"
        }
    },
    "required": [
        "certificateHashData"
    ],
    "type": "object"
},
{
    "$id": "urn:DeleteCertificate.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "definitions": {
        "DeleteCertificateStatusEnumType": {
```

```
                "type": "string",
                "enum": [
                    "Accepted",
                    "Failed",
                    "NotFound"
                ]
            }
        },
        "properties": {
            "status": {
                "$ref": "#/definitions/DeleteCertificateStatusEnumType"
            }
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:DiagnosticsStatusNotification.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "status": {
                "type": "string",
                "enum": [
                    "Idle",
                    "Uploaded",
                    "UploadFailed",
                    "Uploading"
                ]
            }
        },
        "required": [
            "status"
        ],
```

```
        "type": "object"
    },
    {
        "$id": "urn:DiagnosticsStatusNotification.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {},
        "type": "object"
    },
    {
        "$id": "urn:ExtendedTriggerMessage.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "definitions": {
            "MessageTriggerEnumType": {
                "type": "string",
                "enum": [
                    "BootNotification",
                    "LogStatusNotification",
                    "FirmwareStatusNotification",
                    "Heartbeat",
                    "MeterValues",
                    "SignChargePointCertificate",
                    "StatusNotification"
                ]
            }
        },
        "properties": {
            "requestedMessage": {
                "$ref": "#/definitions/MessageTriggerEnumType"
            },
            "connectorId": {
                "type": "integer"
            }
        },
        "required": [
```

```
            "requestedMessage"
        ],
        "type": "object"
    },
    {
        "$id": "urn:ExtendedTriggerMessage.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "definitions": {
            "TriggerMessageStatusEnumType": {
                "type": "string",
                "enum": [
                    "Accepted",
                    "Rejected",
                    "NotImplemented"
                ]
            }
        },
        "properties": {
            "status": {
                "$ref": "#/definitions/TriggerMessageStatusEnumType"
            }
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:FirmwareStatusNotification.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "status": {
                "type": "string",
                "enum": [
```

```
                "Downloaded",
                "DownloadFailed",
                "Downloading",
                "Idle",
                "InstallationFailed",
                "Installing",
                "Installed"
            ]
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
{
    "$id": "urn:FirmwareStatusNotification.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {},
    "type": "object"
},
{
    "$id": "urn:GetCompositeSchedule.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "connectorId": {
            "type": "integer"
        },
        "duration": {
            "type": "integer"
        },
        "chargingRateUnit": {
            "type": "string",
            "enum": [
```

```
                "A",
                "W"
            ]
        }
    },
    "required": [
        "connectorId",
        "duration"
    ],
    "type": "object"
},
{
    "$id": "urn:GetCompositeSchedule.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "Accepted",
                "Rejected"
            ]
        },
        "connectorId": {
            "type": "integer"
        },
        "scheduleStart": {
            "type": "string",
            "format": "date-time"
        },
        "chargingSchedule": {
            "type": "object",
            "properties": {
                "duration": {
                    "type": "integer"
                },
```

```
"startSchedule": {
    "type": "string",
    "format": "date-time"
},
"chargingRateUnit": {
    "type": "string",
    "enum": [
        "A",
        "W"
    ]
},
"chargingSchedulePeriod": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "startPeriod": {
                "type": "integer"
            },
            "limit": {
                "type": "number",
                "multipleOf": 0.1
            },
            "numberPhases": {
                "type": "integer"
            }
        },
        "additionalProperties": false,
        "required": [
            "startPeriod",
            "limit"
        ]
    }
},
"minChargingRate": {
    "type": "number",
```

```
                    "multipleOf": 0.1
                }
            },
            "additionalProperties": false,
            "required": [
                "chargingRateUnit",
                "chargingSchedulePeriod"
            ]
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
{
    "$id": "urn:GetConfiguration.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "key": {
            "type": "array",
            "items": {
                "type": "string",
                "maxLength": 50
            }
        }
    },
    "type": "object"
},
{
    "$id": "urn:GetConfiguration.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "configurationKey": {
```

```
                "type": "array",
                "items": {
                    "type": "object",
                    "properties": {
                        "key": {
                            "type": "string",
                            "maxLength": 50
                        },
                        "readonly": {
                            "type": "boolean"
                        },
                        "value": {
                            "type": "string",
                            "maxLength": 500
                        }
                    },
                    "additionalProperties": false,
                    "required": [
                        "key",
                        "readonly"
                    ]
                }
            },
            "unknownKey": {
                "type": "array",
                "items": {
                    "type": "string",
                    "maxLength": 50
                }
            }
        },
        "type": "object"
    },
    {
        "$id": "urn:GetDiagnostics.req",
        "$schema": "http://json-schema.org/draft-07/schema",
```

```
        "additionalProperties": false,
        "properties": {
            "location": {
                "type": "string",
                "format": "uri"
            },
            "retries": {
                "type": "integer"
            },
            "retryInterval": {
                "type": "integer"
            },
            "startTime": {
                "type": "string",
                "format": "date-time"
            },
            "stopTime": {
                "type": "string",
                "format": "date-time"
            }
        },
        "required": [
            "location"
        ],
        "type": "object"
    },
    {
        "$id": "urn:GetDiagnostics.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "fileName": {
                "type": "string",
                "maxLength": 255
            }
        },
```

```
    "type": "object"
},
{
    "$id": "urn:GetInstalledCertificateIds.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "definitions": {
        "CertificateUseEnumType": {
            "type": "string",
            "enum": [
                "CentralSystemRootCertificate",
                "ManufacturerRootCertificate"
            ]
        }
    },
    "properties": {
        "certificateType": {
            "$ref": "#/definitions/CertificateUseEnumType"
        }
    },
    "required": [
        "certificateType"
    ],
    "type": "object"
},
{
    "$id": "urn:GetInstalledCertificateIds.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "definitions": {
        "GetInstalledCertificateStatusEnumType": {
            "type": "string",
            "enum": [
                "Accepted",
                "NotFound"
            ]
```

```
},
"HashAlgorithmEnumType": {
    "type": "string",
    "enum": [
        "SHA256",
        "SHA384",
        "SHA512"
    ]
},
"CertificateHashDataType": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
        "hashAlgorithm": {
            "$ref": "#/definitions/HashAlgorithmEnumType"
        },
        "issuerNameHash": {
            "type": "string",
            "maxLength": 128
        },
        "issuerKeyHash": {
            "type": "string",
            "maxLength": 128
        },
        "serialNumber": {
            "type": "string",
            "maxLength": 40
        }
    },
    "required": [
        "hashAlgorithm",
        "issuerNameHash",
        "issuerKeyHash",
        "serialNumber"
    ]
}
```

```
    },
    "properties": {
        "certificateHashData": {
            "type": "array",
            "items": {
                "$ref": "#/definitions/CertificateHashDataType"
            },
            "minItems": 1
        },
        "status": {
            "$ref": "#/definitions/GetInstalledCertificateStatusEnumType"
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
{
    "$id": "urn:GetLocalListVersion.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {},
    "type": "object"
},
{
    "$id": "urn:GetLocalListVersion.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "listVersion": {
            "type": "integer"
        }
    },
    "required": [
        "listVersion"
```

```
        ],
        "type": "object"
    },
    {
        "$id": "urn:GetLog.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "definitions": {
            "LogEnumType": {
                "type": "string",
                "enum": [
                    "DiagnosticsLog",
                    "SecurityLog"
                ]
            },
            "LogParametersType": {
                "type": "object",
                "additionalProperties": false,
                "properties": {
                    "remoteLocation": {
                        "type": "string",
                        "maxLength": 512
                    },
                    "oldestTimestamp": {
                        "type": "string",
                        "format": "date-time"
                    },
                    "latestTimestamp": {
                        "type": "string",
                        "format": "date-time"
                    }
                },
                "required": [
                    "remoteLocation"
                ]
            }
```

```
    },
    "properties": {
        "log": {
            "$ref": "#/definitions/LogParametersType"
        },
        "logType": {
            "$ref": "#/definitions/LogEnumType"
        },
        "requestId": {
            "type": "integer"
        },
        "retries": {
            "type": "integer"
        },
        "retryInterval": {
            "type": "integer"
        }
    },
    "required": [
        "logType",
        "requestId",
        "log"
    ],
    "type": "object"
},
{
    "$id": "urn:GetLog.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "definitions": {
        "LogStatusEnumType": {
            "type": "string",
            "enum": [
                "Accepted",
                "Rejected",
                "AcceptedCanceled"
```

```
                ]
            }
        },
        "properties": {
            "status": {
                "$ref": "#/definitions/LogStatusEnumType"
            },
            "filename": {
                "type": "string",
                "maxLength": 255
            }
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:Heartbeat.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {},
        "type": "object"
    },
    {
        "$id": "urn:Heartbeat.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "currentTime": {
                "type": "string",
                "format": "date-time"
            }
        },
        "required": [
            "currentTime"
```

```
    ],
    "type": "object"
},
{
    "$id": "urn:InstallCertificate.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "definitions": {
        "CertificateUseEnumType": {
            "type": "string",
            "enum": [
                "CentralSystemRootCertificate",
                "ManufacturerRootCertificate"
            ]
        }
    },
    "properties": {
        "certificateType": {
            "$ref": "#/definitions/CertificateUseEnumType"
        },
        "certificate": {
            "type": "string",
            "maxLength": 5500
        }
    },
    "required": [
        "certificateType",
        "certificate"
    ],
    "type": "object"
},
{
    "$id": "urn:InstallCertificate.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "definitions": {
```

```
            "InstallCertificateStatusEnumType": {
                "type": "string",
                "enum": [
                    "Accepted",
                    "Failed",
                    "Rejected"
                ]
            }
        },
        "properties": {
            "status": {
                "$ref": "#/definitions/InstallCertificateStatusEnumType"
            }
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:LogStatusNotification.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "definitions": {
            "UploadLogStatusEnumType": {
                "type": "string",
                "enum": [
                    "BadMessage",
                    "Idle",
                    "NotSupportedOperation",
                    "PermissionDenied",
                    "Uploaded",
                    "UploadFailure",
                    "Uploading"
                ]
            }
```

```
        },
        "properties": {
            "status": {
                "$ref": "#/definitions/UploadLogStatusEnumType"
            },
            "requestId": {
                "type": "integer"
            }
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:LogStatusNotification.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "type": "object"
    },
    {
        "$id": "urn:MeterValues.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "connectorId": {
                "type": "integer"
            },
            "transactionId": {
                "type": "integer"
            },
            "meterValue": {
                "type": "array",
                "items": {
                    "type": "object",
                    "properties": {
```

```
"timestamp": {
    "type": "string",
    "format": "date-time"
},
"sampledValue": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "value": {
                "type": "string"
            },
            "context": {
                "type": "string",
                "enum": [
                    "Interruption.Begin",
                    "Interruption.End",
                    "Sample.Clock",
                    "Sample.Periodic",
                    "Transaction.Begin",
                    "Transaction.End",
                    "Trigger",
                    "Other"
                ]
            },
            "format": {
                "type": "string",
                "enum": [
                    "Raw",
                    "SignedData"
                ]
            },
            "measurand": {
                "type": "string",
                "enum": [
                    "Energy.Active.Export.Register",
```

```
                "Energy.Active.Import.Register",
                "Energy.Reactive.Export.Register",
                "Energy.Reactive.Import.Register",
                "Energy.Active.Export.Interval",
                "Energy.Active.Import.Interval",
                "Energy.Reactive.Export.Interval",
                "Energy.Reactive.Import.Interval",
                "Power.Active.Export",
                "Power.Active.Import",
                "Power.Offered",
                "Power.Reactive.Export",
                "Power.Reactive.Import",
                "Power.Factor",
                "Current.Import",
                "Current.Export",
                "Current.Offered",
                "Voltage",
                "Frequency",
                "Temperature",
                "SoC",
                "RPM"
            ]
        },
        "phase": {
            "type": "string",
            "enum": [
                "L1",
                "L2",
                "L3",
                "N",
                "L1-N",
                "L2-N",
                "L3-N",
                "L1-L2",
                "L2-L3",
                "L3-L1"
```

```
                ]
            },
            "location": {
                "type": "string",
                "enum": [
                    "Cable",
                    "EV",
                    "Inlet",
                    "Outlet",
                    "Body"
                ]
            },
            "unit": {
                "type": "string",
                "enum": [
                    "Wh",
                    "kWh",
                    "varh",
                    "kvarh",
                    "W",
                    "kW",
                    "VA",
                    "kVA",
                    "var",
                    "kvar",
                    "A",
                    "V",
                    "K",
                    "Celcius",
                    "Celsius",
                    "Fahrenheit",
                    "Percent"
                ]
            }
        },
        "additionalProperties": false,
```

```
                            "required": [
                                "value"
                            ]
                        }
                    }
                },
                "additionalProperties": false,
                "required": [
                    "timestamp",
                    "sampledValue"
                ]
            }
        }
    },
    "required": [
        "connectorId",
        "meterValue"
    ],
    "type": "object"
},
{
    "$id": "urn:MeterValues.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {},
    "type": "object"
},
{
    "$id": "urn:RemoteStartTransaction.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "connectorId": {
            "type": "integer"
        },
        "idTag": {
```

```
            "type": "string",
            "maxLength": 20
        },
        "chargingProfile": {
            "type": "object",
            "properties": {
                "chargingProfileId": {
                    "type": "integer"
                },
                "transactionId": {
                    "type": "integer"
                },
                "stackLevel": {
                    "type": "integer"
                },
                "chargingProfilePurpose": {
                    "type": "string",
                    "enum": [
                        "ChargePointMaxProfile",
                        "TxDefaultProfile",
                        "TxProfile"
                    ]
                },
                "chargingProfileKind": {
                    "type": "string",
                    "enum": [
                        "Absolute",
                        "Recurring",
                        "Relative"
                    ]
                },
                "recurrencyKind": {
                    "type": "string",
                    "enum": [
                        "Daily",
                        "Weekly"
```

```
        ]
    },
    "validFrom": {
        "type": "string",
        "format": "date-time"
    },
    "validTo": {
        "type": "string",
        "format": "date-time"
    },
    "chargingSchedule": {
        "type": "object",
        "properties": {
            "duration": {
                "type": "integer"
            },
            "startSchedule": {
                "type": "string",
                "format": "date-time"
            },
            "chargingRateUnit": {
                "type": "string",
                "enum": [
                    "A",
                    "W"
                ]
            },
            "chargingSchedulePeriod": {
                "type": "array",
                "items": {
                    "type": "object",
                    "properties": {
                        "startPeriod": {
                            "type": "integer"
                        },
                        "limit": {
```

```
                                    "type": "number",
                                    "multipleOf": 0.1
                                },
                                "numberPhases": {
                                    "type": "integer"
                                }
                            },
                            "additionalProperties": false,
                            "required": [
                                "startPeriod",
                                "limit"
                            ]
                        }
                    },
                    "minChargingRate": {
                        "type": "number",
                        "multipleOf": 0.1
                    }
                },
                "additionalProperties": false,
                "required": [
                    "chargingRateUnit",
                    "chargingSchedulePeriod"
                ]
            }
        },
        "additionalProperties": false,
        "required": [
            "chargingProfileId",
            "stackLevel",
            "chargingProfilePurpose",
            "chargingProfileKind",
            "chargingSchedule"
        ]
    }
},
```

```
        "required": [
            "idTag"
        ],
        "type": "object"
    },
    {
        "$id": "urn:RemoteStartTransaction.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "status": {
                "type": "string",
                "enum": [
                    "Accepted",
                    "Rejected"
                ]
            }
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:RemoteStopTransaction.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "transactionId": {
                "type": "integer"
            }
        },
        "required": [
            "transactionId"
        ],
        "type": "object"
```

```
},
{
    "$id": "urn:RemoteStopTransaction.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "Accepted",
                "Rejected"
            ]
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
{
    "$id": "urn:ReserveNow.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "connectorId": {
            "type": "integer"
        },
        "expiryDate": {
            "type": "string",
            "format": "date-time"
        },
        "idTag": {
            "type": "string",
            "maxLength": 20
        },
        "parentIdTag": {
```

```
            "type": "string",
            "maxLength": 20
        },
        "reservationId": {
            "type": "integer"
        }
    },
    "required": [
        "connectorId",
        "expiryDate",
        "idTag",
        "reservationId"
    ],
    "type": "object"
},
{
    "$id": "urn:ReserveNow.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "Accepted",
                "Faulted",
                "Occupied",
                "Rejected",
                "Unavailable"
            ]
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
```

```
{
    "$id": "urn:Reset.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "Hard",
                "Soft"
            ]
        }
    },
    "required": [
        "type"
    ],
    "type": "object"
},
{
    "$id": "urn:Reset.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "Accepted",
                "Rejected"
            ]
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
```

```
{
    "$id": "urn:SecurityEventNotification.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "type": {
            "type": "string",
            "maxLength": 50
        },
        "timestamp": {
            "type": "string",
            "format": "date-time"
        },
        "techInfo": {
            "type": "string",
            "maxLength": 255
        }
    },
    "required": [
        "type",
        "timestamp"
    ],
    "type": "object"
},
{
    "$id": "urn:SecurityEventNotification.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "type": "object"
},
{
    "$id": "urn:SendLocalList.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "listVersion": {
```

```
        "type": "integer"
    },
    "localAuthorizationList": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "idTag": {
                    "type": "string",
                    "maxLength": 20
                },
                "idTagInfo": {
                    "type": "object",
                    "properties": {
                        "expiryDate": {
                            "type": "string",
                            "format": "date-time"
                        },
                        "parentIdTag": {
                            "type": "string",
                            "maxLength": 20
                        },
                        "status": {
                            "type": "string",
                            "enum": [
                                "Accepted",
                                "Blocked",
                                "Expired",
                                "Invalid",
                                "ConcurrentTx"
                            ]
                        }
                    },
                    "additionalProperties": false,
                    "required": [
                        "status"
```

```
                        ]
                    }
                },
                "additionalProperties": false,
                "required": [
                    "idTag"
                ]
            }
        },
        "updateType": {
            "type": "string",
            "enum": [
                "Differential",
                "Full"
            ]
        }
    },
    "required": [
        "listVersion",
        "updateType"
    ],
    "type": "object"
},
{
    "$id": "urn:SendLocalList.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "Accepted",
                "Failed",
                "NotSupported",
                "VersionMismatch"
            ]
```

```
            }
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:SetChargingProfile.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "connectorId": {
                "type": "integer"
            },
            "csChargingProfiles": {
                "type": "object",
                "properties": {
                    "chargingProfileId": {
                        "type": "integer"
                    },
                    "transactionId": {
                        "type": "integer"
                    },
                    "stackLevel": {
                        "type": "integer"
                    },
                    "chargingProfilePurpose": {
                        "type": "string",
                        "enum": [
                            "ChargePointMaxProfile",
                            "TxDefaultProfile",
                            "TxProfile"
                        ]
                    },
                    "chargingProfileKind": {
```

```
            "type": "string",
            "enum": [
                "Absolute",
                "Recurring",
                "Relative"
            ]
        },
        "recurrencyKind": {
            "type": "string",
            "enum": [
                "Daily",
                "Weekly"
            ]
        },
        "validFrom": {
            "type": "string",
            "format": "date-time"
        },
        "validTo": {
            "type": "string",
            "format": "date-time"
        },
        "chargingSchedule": {
            "type": "object",
            "properties": {
                "duration": {
                    "type": "integer"
                },
                "startSchedule": {
                    "type": "string",
                    "format": "date-time"
                },
                "chargingRateUnit": {
                    "type": "string",
                    "enum": [
                        "A",
```

```
                    "W"
                ]
            },
            "chargingSchedulePeriod": {
                "type": "array",
                "items": {
                    "type": "object",
                    "properties": {
                        "startPeriod": {
                            "type": "integer"
                        },
                        "limit": {
                            "type": "number",
                            "multipleOf": 0.1
                        },
                        "numberPhases": {
                            "type": "integer"
                        }
                    },
                    "additionalProperties": false,
                    "required": [
                        "startPeriod",
                        "limit"
                    ]
                }
            },
            "minChargingRate": {
                "type": "number",
                "multipleOf": 0.1
            }
        },
        "additionalProperties": false,
        "required": [
            "chargingRateUnit",
            "chargingSchedulePeriod"
        ]
```

```
                }
            },
            "additionalProperties": false,
            "required": [
                "chargingProfileId",
                "stackLevel",
                "chargingProfilePurpose",
                "chargingProfileKind",
                "chargingSchedule"
            ]
        }
    },
    "required": [
        "connectorId",
        "csChargingProfiles"
    ],
    "type": "object"
},
{
    "$id": "urn:SetChargingProfile.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "Accepted",
                "Rejected",
                "NotSupported"
            ]
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
```

```
},
{
    "$id": "urn:SignCertificate.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "csr": {
            "type": "string",
            "maxLength": 5500
        }
    },
    "required": [
        "csr"
    ],
    "type": "object"
},
{
    "$id": "urn:SignCertificate.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "definitions": {
        "GenericStatusEnumType": {
            "type": "string",
            "enum": [
                "Accepted",
                "Rejected"
            ]
        }
    },
    "properties": {
        "status": {
            "$ref": "#/definitions/GenericStatusEnumType"
        }
    },
    "required": [
        "status"
```

```
    ],
    "type": "object"
},
{
    "$id": "urn:SignedFirmwareStatusNotification.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "definitions": {
        "FirmwareStatusEnumType": {
            "type": "string",
            "enum": [
                "Downloaded",
                "DownloadFailed",
                "Downloading",
                "DownloadScheduled",
                "DownloadPaused",
                "Idle",
                "InstallationFailed",
                "Installing",
                "Installed",
                "InstallRebooting",
                "InstallScheduled",
                "InstallVerificationFailed",
                "InvalidSignature",
                "SignatureVerified"
            ]
        }
    },
    "properties": {
        "status": {
            "$ref": "#/definitions/FirmwareStatusEnumType"
        },
        "requestId": {
            "type": "integer"
        }
    },
```

```
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:SignedFirmwareStatusNotification.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "type": "object"
    },
    {
        "$id": "urn:SignedUpdateFirmware.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "definitions": {
            "FirmwareType": {
                "type": "object",
                "additionalProperties": false,
                "properties": {
                    "location": {
                        "type": "string",
                        "maxLength": 512
                    },
                    "retrieveDateTime": {
                        "type": "string",
                        "format": "date-time"
                    },
                    "installDateTime": {
                        "type": "string",
                        "format": "date-time"
                    },
                    "signingCertificate": {
                        "type": "string",
                        "maxLength": 5500
                    },
```

```
                "signature": {
                    "type": "string",
                    "maxLength": 800
                }
            },
            "required": [
                "location",
                "retrieveDateTime",
                "signingCertificate",
                "signature"
            ]
        }
    },
    "properties": {
        "retries": {
            "type": "integer"
        },
        "retryInterval": {
            "type": "integer"
        },
        "requestId": {
            "type": "integer"
        },
        "firmware": {
            "$ref": "#/definitions/FirmwareType"
        }
    },
    "required": [
        "requestId",
        "firmware"
    ],
    "type": "object"
},
{
    "$id": "urn:SignedUpdateFirmware.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
```

```
        "additionalProperties": false,
        "definitions": {
            "UpdateFirmwareStatusEnumType": {
                "type": "string",
                "enum": [
                    "Accepted",
                    "Rejected",
                    "AcceptedCanceled",
                    "InvalidCertificate",
                    "RevokedCertificate"
                ]
            }
        },
        "properties": {
            "status": {
                "$ref": "#/definitions/UpdateFirmwareStatusEnumType"
            }
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:StartTransaction.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "connectorId": {
                "type": "integer"
            },
            "idTag": {
                "type": "string",
                "maxLength": 20
            },
            "meterStart": {
```

```
            "type": "integer"
        },
        "reservationId": {
            "type": "integer"
        },
        "timestamp": {
            "type": "string",
            "format": "date-time"
        }
    },
    "required": [
        "connectorId",
        "idTag",
        "meterStart",
        "timestamp"
    ],
    "type": "object"
},
{
    "$id": "urn:StartTransaction.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "idTagInfo": {
            "type": "object",
            "properties": {
                "expiryDate": {
                    "type": "string",
                    "format": "date-time"
                },
                "parentIdTag": {
                    "type": "string",
                    "maxLength": 20
                },
                "status": {
                    "type": "string",
```

```
                    "enum": [
                        "Accepted",
                        "Blocked",
                        "Expired",
                        "Invalid",
                        "ConcurrentTx"
                    ]
                }
            },
            "additionalProperties": false,
            "required": [
                "status"
            ]
        },
        "transactionId": {
            "type": "integer"
        }
    },
    "required": [
        "idTagInfo",
        "transactionId"
    ],
    "type": "object"
},
{
    "$id": "urn:StatusNotification.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "connectorId": {
            "type": "integer"
        },
        "errorCode": {
            "type": "string",
            "enum": [
                "ConnectorLockFailure",
```

```
            "EVCommunicationError",
            "GroundFailure",
            "HighTemperature",
            "InternalError",
            "LocalListConflict",
            "NoError",
            "OtherError",
            "OverCurrentFailure",
            "PowerMeterFailure",
            "PowerSwitchFailure",
            "ReaderFailure",
            "ResetFailure",
            "UnderVoltage",
            "OverVoltage",
            "WeakSignal"
        ]
    },
    "info": {
        "type": "string",
        "maxLength": 50
    },
    "status": {
        "type": "string",
        "enum": [
            "Available",
            "Preparing",
            "Charging",
            "SuspendedEVSE",
            "SuspendedEV",
            "Finishing",
            "Reserved",
            "Unavailable",
            "Faulted"
        ]
    },
    "timestamp": {
```

```
                "type": "string",
                "format": "date-time"
            },
            "vendorId": {
                "type": "string",
                "maxLength": 255
            },
            "vendorErrorCode": {
                "type": "string",
                "maxLength": 50
            }
        },
        "required": [
            "connectorId",
            "errorCode",
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:StatusNotification.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {},
        "type": "object"
    },
    {
        "$id": "urn:StopTransaction.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "idTag": {
                "type": "string",
                "maxLength": 20
            },
            "meterStop": {
```

```
            "type": "integer"
        },
        "timestamp": {
            "type": "string",
            "format": "date-time"
        },
        "transactionId": {
            "type": "integer"
        },
        "reason": {
            "type": "string",
            "enum": [
                "EmergencyStop",
                "EVDisconnected",
                "HardReset",
                "Local",
                "Other",
                "PowerLoss",
                "Reboot",
                "Remote",
                "SoftReset",
                "UnlockCommand",
                "DeAuthorized"
            ]
        },
        "transactionData": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "timestamp": {
                        "type": "string",
                        "format": "date-time"
                    },
                    "sampledValue": {
                        "type": "array",
```

```
            "items": {
                "type": "object",
                "properties": {
                    "value": {
                        "type": "string"
                    },
                    "context": {
                        "type": "string",
                        "enum": [
                            "Interruption.Begin",
                            "Interruption.End",
                            "Sample.Clock",
                            "Sample.Periodic",
                            "Transaction.Begin",
                            "Transaction.End",
                            "Trigger",
                            "Other"
                        ]
                    },
                    "format": {
                        "type": "string",
                        "enum": [
                            "Raw",
                            "SignedData"
                        ]
                    },
                    "measurand": {
                        "type": "string",
                        "enum": [
                            "Energy.Active.Export.Register",
                            "Energy.Active.Import.Register",
                            "Energy.Reactive.Export.Register",
                            "Energy.Reactive.Import.Register",
                            "Energy.Active.Export.Interval",
                            "Energy.Active.Import.Interval",
                            "Energy.Reactive.Export.Interval",
```

```
                "Energy.Reactive.Import.Interval",
                "Power.Active.Export",
                "Power.Active.Import",
                "Power.Offered",
                "Power.Reactive.Export",
                "Power.Reactive.Import",
                "Power.Factor",
                "Current.Import",
                "Current.Export",
                "Current.Offered",
                "Voltage",
                "Frequency",
                "Temperature",
                "SoC",
                "RPM"
            ]
        },
        "phase": {
            "type": "string",
            "enum": [
                "L1",
                "L2",
                "L3",
                "N",
                "L1-N",
                "L2-N",
                "L3-N",
                "L1-L2",
                "L2-L3",
                "L3-L1"
            ]
        },
        "location": {
            "type": "string",
            "enum": [
                "Cable",
```

```
                                "EV",
                                "Inlet",
                                "Outlet",
                                "Body"
                            ]
                        },
                        "unit": {
                            "type": "string",
                            "enum": [
                                "Wh",
                                "kWh",
                                "varh",
                                "kvarh",
                                "W",
                                "kW",
                                "VA",
                                "kVA",
                                "var",
                                "kvar",
                                "A",
                                "V",
                                "K",
                                "Celcius",
                                "Celsius",
                                "Fahrenheit",
                                "Percent"
                            ]
                        }
                    },
                    "additionalProperties": false,
                    "required": [
                        "value"
                    ]
                }
            }
        },
```

```
                "additionalProperties": false,
                "required": [
                    "timestamp",
                    "sampledValue"
                ]
            }
        }
    },
    "required": [
        "transactionId",
        "timestamp",
        "meterStop"
    ],
    "type": "object"
},
{
    "$id": "urn:StopTransaction.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "idTagInfo": {
            "type": "object",
            "properties": {
                "expiryDate": {
                    "type": "string",
                    "format": "date-time"
                },
                "parentIdTag": {
                    "type": "string",
                    "maxLength": 20
                },
                "status": {
                    "type": "string",
                    "enum": [
                        "Accepted",
                        "Blocked",
```

```
                        "Expired",
                        "Invalid",
                        "ConcurrentTx"
                    ]
                }
            },
            "additionalProperties": false,
            "required": [
                "status"
            ]
        }
    },
    "type": "object"
},
{
    "$id": "urn:TriggerMessage.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "requestedMessage": {
            "type": "string",
            "enum": [
                "BootNotification",
                "DiagnosticsStatusNotification",
                "FirmwareStatusNotification",
                "Heartbeat",
                "MeterValues",
                "StatusNotification"
            ]
        },
        "connectorId": {
            "type": "integer"
        }
    },
    "required": [
        "requestedMessage"
```

```
        ],
        "type": "object"
    },
    {
        "$id": "urn:TriggerMessage.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "status": {
                "type": "string",
                "enum": [
                    "Accepted",
                    "Rejected",
                    "NotImplemented"
                ]
            }
        },
        "required": [
            "status"
        ],
        "type": "object"
    },
    {
        "$id": "urn:UnlockConnector.req",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {
            "connectorId": {
                "type": "integer"
            }
        },
        "required": [
            "connectorId"
        ],
        "type": "object"
    },
```

```
{
    "$id": "urn:UnlockConnector.conf",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "Unlocked",
                "UnlockFailed",
                "NotSupported"
            ]
        }
    },
    "required": [
        "status"
    ],
    "type": "object"
},
{
    "$id": "urn:UpdateFirmware.req",
    "$schema": "http://json-schema.org/draft-07/schema",
    "additionalProperties": false,
    "properties": {
        "location": {
            "type": "string",
            "format": "uri"
        },
        "retries": {
            "type": "integer"
        },
        "retrieveDate": {
            "type": "string",
            "format": "date-time"
        },
        "retryInterval": {
```

```
                "type": "integer"
            }
        },
        "required": [
            "location",
            "retrieveDate"
        ],
        "type": "object"
    },
    {
        "$id": "urn:UpdateFirmware.conf",
        "$schema": "http://json-schema.org/draft-07/schema",
        "additionalProperties": false,
        "properties": {},
        "type": "object"
    }
]
```