International Journal of Information Security (2025) 24:134
https://doi.org/10.1007/s10207-025-01055-7

REGULAR CONTRIBUTION

Cyber defense in OCPP for EV charging security risks

Safa Hamdare[1] · David J. Brown[1] · Devki Nandan Jha[2] · Mohammad Aljaidi[3] · Yue Cao[4] · Sushil Kumar[5] · Rupak Kharel[6] · Manish Jugran[7] · Omprakash Kaiwartya[1]

Published online: 21 May 2025

Abstract
The Open Charge Point Protocol (OCPP) is a widely adopted communication standard that enables vendor-independent communication between charging points and Electric Vehicle (EV) charging station management systems. OCPP has significant cyber risks in terms of weak authentication mechanisms and improper session handling, exposing it to potential EV charging-related security threats. The backward incompatibility of the recent version of OCPP also poses challenges in the seamless adoption of the protocol. This paper introduces a comprehensive cyber defense framework to mitigate the security risks associated with OCPP. Through a detailed analysis of its vulnerabilities, the framework proposes targeted enhancements and mitigation strategies to further strengthen its security. The results demonstrate that the proposed OCPP significantly enhances both security and performance, surpassing its predecessor and current state-of-the-art security solutions for EV charging.

Keywords Electric vehicles · cyber secuirty · secuirty risk · OCPP

1 Introduction

The rise of Electric Vehicle Charging Station (EVCS) globally is closely linked to the increasing adoption of EVs [1]. Facilitating this growth is OCPP [2], a communication standard designed to enhance interactions between EVCS and central management system [3]. Fig. 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 [4], [5]. OCPP accommodates both SOAP/XML and WebSockets/JSON protocols, ensuring robust and flexible communication [6]. SOAP/XML provides compatibility with older systems, ensuring clear and structured messaging, while WebSockets/JSON enables faster, real-time interactions by maintaining open connections and minimizing delays. Despite the availability of OCPP new release with advanced features, OCPP cyber risks remains critical and it is widely adopted due to its ease of implementation, user-friendly interface, and extensive existing infrastructure [7].

In particular, OCPP has critical cyber security vulnerabilities [8]. For example, practical exploits involving unencrypted WebSocket communication within OCPP have demonstrated how attackers could terminate charging sessions, execute remote code, and deploy malicious firmware on Electric Vehicle Supply Equipment [6]. Saiflow have identified improper handling of multiple connections from a single Charge Point (CP) [9], potentially exposing charging infrastructure to DoS attacks. The investigation carried by them also uncovered inadequate authentication mechanisms within previous OCPP [9], leaving systems vulnerable to unauthorized access and potential exploitation by malicious actors. Flaws in the management of reservation IDs have been found [10], highlighting risks such as fraudulent

✉ Omprakash Kaiwartya
omprakash.kaiwartya@ntu.ac.uk

[1] Department of Computer Science, Nottingham Trent University, Clifton Lane, NG11 8NS Nottingham, United Kingdom

[2] Department of Computing, Newcastle University, NE1 7RU Newcastle, United Kingdom

[3] Department of Computer Science, Faculty of Information Technology, Zarqa University, 13110 Zarqa, Jordan

[4] School of Cyber Science and Engineering, Wuhan University, Wuhan, China

[5] School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi, India

[6] School of Computing and Engineering, University of Huddersfield, HD1 3DH Huddersfield, United Kingdom

[7] JMVL Limited, Jenkins Avenue, Bricket Wood, AL2 3SB London, United Kingdom

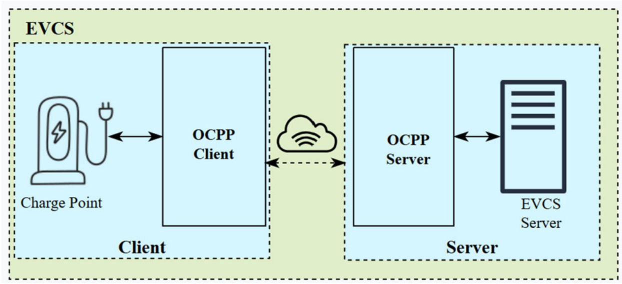


Fig. 1 OCPP Communication Architecture

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 Charge Point Server (CS) [11], making it possible for attackers to hijack connections using stolen user credentials and reservation IDs. Moreover, Weaknesses in the implementation of Transport Layer Security (TLS) make OCPP vulnerable to Man-in-the-Middle (MitM) attacks, allowing attackers to intercept and alter data during transmission [12]. Thus, insecure backend communication, firmware theft, and unauthorized access to EV data further compromise the confidentiality and integrity of charging transactions [11], [13].These tangible threats underscore the need for immediate security enhancements.

Mitigating these risks is essential to ensure the secure and reliable operation of EV charging infrastructure. As electric mobility becomes increasingly vital in the global effort to reduce carbon emissions and combat climate change [14], the security of charging networks becomes paramount. Unaddressed vulnerabilities could lead to operational disruptions, financial losses, and erosion of user trust in EV technologies [15]. This paper aims to provide a thorough analysis of the identified security risks in OCPP and propose effective enhancements and mitigation strategies to fortify the protocol against potential threats. By improving the security measures within OCPP, the aim is to support the continued growth and acceptance of electric mobility solutions while safeguarding against cyber threats. The key contribution of this paper can be summarized as follows:

1. Identify critical security threats in OCPP using the STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service (DoS), Elevation of Privilege) model.
2. Analyze each identified threats and map it to possible attacks using an attack tree.
3. Assess the potential security risks of each attack using the DREAD (Damage Potential, Reproducibility, Exploitability, Affected Users, Discover ability) model.
4. Mitigate the identified threats by proposing and implementing practical solutions to enhance OCPP security.
5. A comparative analysis is made using both the current state of the art and data generated from the model. The enhanced OCPP shows superior performance and security over the previous OCPP.

Table 1 Nomenclature

Notation	Description
EV	Electric Vehicle
OCPP	Open Charge Point Protocol
EVCS	Electric Vehicle Charging Station
CP	Charge Point
CS	Charge Point Server
TLS	Transport Layer Security
MitM	Man-in-the-Middle
DoS	Denial of Service
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of Privilege
DREAD	Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability
ECUs	Electronic Control Units
CAN	Controller Area Network
IAAM	Identify, Analyze, Assess, Mitigate
EVCMS	Electric Vehicle Charging Management System
OTP	One Time Password

The structure of the paper is as follows: Section 2 provides the research background. Section 3 outlines the proposed security framework methodology, which includes the identification and analysis of OCPP threats using the STRIDE model, their evaluation through the DREAD model, and the proposal of mitigation strategies via enhanced communication processes. Section 4 presents the result analysis, encompassing the experimental setup with server log analysis, implementation cost, and performance impact. Section 5 offers a comparison with state-of-the-art approaches, while Section 6 presents a comparative data evaluation of the improved OCPP against its previous version. Finally, Section 7 concludes the study and proposes directions for future research. Table 1 provides the nomenclature for key terms and acronyms used throughout the paper.

2 Background

This section provides an overview of the OCPP transaction flow, followed by an analysis of vulnerabilities identified by researchers using the STRIDE model.

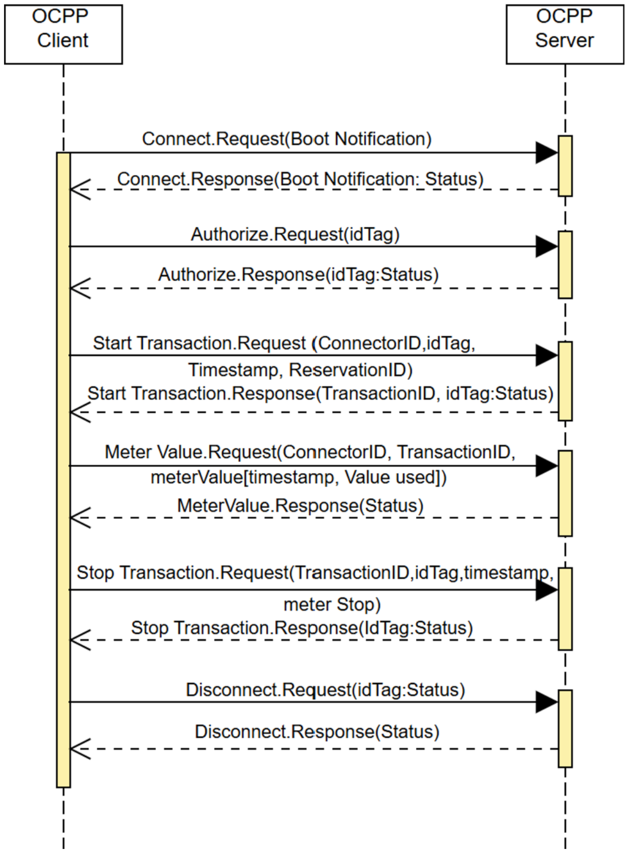

OCPP Client
OCPP Server
Connect.Request(Boot Notification)
Connect.Response(Boot Notification: Status)
Authorize.Request(idTag)
Authorize.Response(idTag:Status)
Start Transaction.Request (ConnectorID,idTag, Timestamp, ReservationID)
Start Transaction.Response(TransactionID, idTag:Status)
Meter Value.Request(ConnectorID, TransactionID, meterValue[timestamp, Value used])
MeterValue.Response(Status)
Stop Transaction.Request(TransactionID,idTag,timestamp, meter Stop)
Stop Transaction.Response(IdTag:Status)
Disconnect.Request(idTag:Status)
Disconnect.Response(Status)

Fig. 2 Transaction message flow in OCPP.

2.1 OCPP Transaction

The OCPP facilitates communication between CP and CS 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. 2, where CP initiate requests to CS 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 CS. This message ensures that the CP is recognized and can communicate with the CS. The corresponding response from the CS 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 CS to verify the credentials of the user. The CS processes this request to ensure that the user is permitted to use the charging service. The Authorize response from the CS 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 CS to initiate the charging session. This request includes details such as the user ID and the connector used. The CS 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 CS. This request contains the transaction ID, user ID, timestamp and meter reading. The CS 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 CS to report the current meter readings. This allows the CS to monitor the energy consumption in real-time. The CS 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 CS needs to be terminated, a Disconnect request is sent by the CP. This message informs the CS that the CP will no longer be available for communication. The CS responds with a Disconnect response, confirming that the disconnection process has been acknowledged and completed.

2.2 Vulnerability analysis in ev charging system

The STRIDE model, developed by Microsoft, is a widely used threat classification framework designed to identify security risks across six key categories [16]. Each category corresponds to a specific type of security threat, allowing for a comprehensive assessment of potential vulnerabilities within EVCS [17]. 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 2. 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.

Table 2 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 [18]	✓	-	-	✓	✓	✓	-	✓	✓
Rouf et al. [19]	✓	-	-	✓	✓	✓	✓	✓	-
Checkoway, S., et al. [20]	✓	-	-	✓	✓	✓	✓	✓	✓
Woo et al. [21]	✓	-	-	✓	✓	✓	✓	✓	-
Jafarnejad, S. et.al. [23]	✓	-	-	✓	✓	✓	-	✓	-
Garcia et al. [24]	✓	-	-	✓	✓	✓	-	✓	-
Mazloom, S. et.al. [25]	✓	-	-	✓	✓	-	✓	-	✓
Karthik, T. et.al. [26]	✓	-	-	✓	✓	-	✓	✓	-
Currie, R. et al. [27]	✓	-	-	✓	✓	✓	-	✓	-
Luo, Q. et.al. [28]	✓	-	-	✓	✓	-	✓	-	✓
Jouvray, C. et al. [29]	-	✓	-	✓	✓	✓	✓	✓	✓
Schneider [30]	-	✓	-	✓	✓	-	✓	-	-
Circontrol [31]	-	✓	-	✓	-	-	✓	-	-
P. van Aubel et al. [32]	-	✓	-	✓	✓	-	✓	-	-
Antoun, J., et.al [33]	-	✓	-	✓	✓	✓	✓	✓	✓
Acharya et al. [34]	-	✓	-	✓	✓	-	-	-	-
Girdhar, M. et al. [35]	-	✓	-	✓	✓	✓	-	-	-
Carryl, C. et.al [36]	✓	✓	-	✓	-	-	✓	✓	-
Baker, R. et.al. [37]	✓	✓	-	✓	-	-	✓	✓	-
Sayed, M.A. et.al [38]	✓	✓	-	✓	✓	-	-	-	✓
Rubio, J.E. et al. [12]	-	-	✓	✓	✓	-	✓	-	-
Alcaraz, C. et al. [10]	-	-	✓	✓	✓	-	✓	✓	-
Garofalaki, Z. et.al [8]	-	-	✓	✓	✓	-	✓	✓	✓
Gebauer, L. et al. [13]	-	-	✓	✓	✓	-	✓	✓	✓
Johnson et al. [6]	-	-	✓	✓	✓	✓	✓	✓	✓
Sarieddine, K. et al. [11]	-	-	✓	-	✓	✓	✓	✓	✓

Koscher et al. [18] 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. [19] 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. [20] 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. [21] analyzed Controller Area Network (CAN) vulnerabilities, highlighting how malicious applications could impersonate legitimate 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, along with Manderna et al.'s [22] focus on vehicular network intrusion detection, emphasize the systemic risks EVs face as they adopt increasingly complex communication protocols.

Jafarnejad et al. [23] analyzed 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. [24] 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. [25] analyzed 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. [26] 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. [27] 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. [28] 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. [29] 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 [30] 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 [31] 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. [32] 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. [33], using the STRIDE framework, analyzed 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. [34] 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. [35] 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. [36] 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. [37] 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. [38] 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. [12] 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. [10] 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. [8] applied the STRIDE framework to analyze 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. [13] 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. [6] 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 MitMattacks, 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. [11] 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 analyzed 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, which are the most widely adopted in existing EVCS infrastructure. Unlike newer OCPP versions, which introduce security improvements but require hardware upgrades due to a lack of backward compatibility, this work strengthens the security of older OCPP versions 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.

2.3 Overview of common attacks in evcs and their impacts

Recent studies have highlighted various cyber-attacks that can be maliciously executed to exploit vulnerabilities in EVCS systems, particularly those using the OCPP. These attacks can cause significant disruptions in both the charging infrastructure and the power grid, leading to financial losses, system instability, and grid failures.

One common attack is the **MITM attack**, where an attacker intercepts communication between the EVCS and the central server. By doing so, the attacker can manipulate critical data, such as energy consumption figures, to steal electricity or alter billing information. Such attacks can result in economic losses for charging service providers and disrupt the accuracy of energy usage records. This issue is particularly relevant in OCPP-based systems, where encrypted communication is essential for secure data exchange [10]. **Data-driven cyber-attacks** are another threat, where attackers exploit the data exchanged between Plug-in Electric Vehicles and the EVCS to cause disruptions. These attacks can manipulate charging data or overload the grid, affecting energy markets and creating inefficiencies in power distribution. For example, by injecting incorrect charging data, attackers can alter the load profile and affect the energy consumption of EVs, leading to misaligned charging schedules and grid instability [34].The EV botnet attack is another emerging threat. In this attack, a large number of compromised EVCS devices are used to launch coordinated attacks on the power grid. This can result in a **Distributed DoS attack** or cause massive surges in demand across the grid, overwhelming power distribution systems and causing widespread power outages. The creation of an EV botnet can lead to significant strain on the electrical infrastructure, as demonstrated in studies involving IEEE 33-bus and 39-bus models [39].

Dynamic load alteration attacks manipulate the charging behavior of multiple EVs, causing sudden fluctuations in the grid's load. These fluctuations can destabilize the grid, resulting in frequency and voltage instability. This type of attack can compromise the operational stability of the power system and lead to cascading failures if the grid cannot handle the sudden changes in demand [40]. Similarly, attacks on load demand manipulation directly affect the load profile by altering the energy usage patterns at EVCS locations. These actions can lead to inefficiencies in energy distribution and negatively impact the power system's stability. By controlling the demand at multiple locations, attackers can influence the overall load on the grid, creating potential risks for power outages or system imbalances [41]. **Frequency-related attacks** can occur when attackers manipulate the charging profiles or operation of EVs, pushing the grid frequency outside safe operational limits. This can lead to grid instability, triggering frequency drops below critical thresholds or causing surges that threaten the integrity of the grid. Such disturbances can cause load shedding, blackouts, or equipment damage [42]. Finally, **coordinated charging attacks** involve controlling a large number of charging stations to impose abnormal loads on the grid. When charging actions are synchronized maliciously, it can lead to excessive energy consumption in a short time, placing significant strain on local transformers and substations. This type of attack can cause widespread disruptions in power distribution networks and potentially lead to cascading failures [43].

These attacks can have profound economic and operational impacts. They can compromise the reliability of the grid, increase power losses, and cause financial damage to both utility providers and EVCS operators. Researchers have found that such vulnerabilities can severely affect the performance of distribution networks, making it critical to improve security protocols, such as enhancing the OCPP encryption and adopting better session management practices, to mitigate these risks.

3 Proposed iaam security framework: methodology

In this section, we propose the Identify, Analyze, Assess, Mitigate (IAAM) framework as shown in Fig. 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.

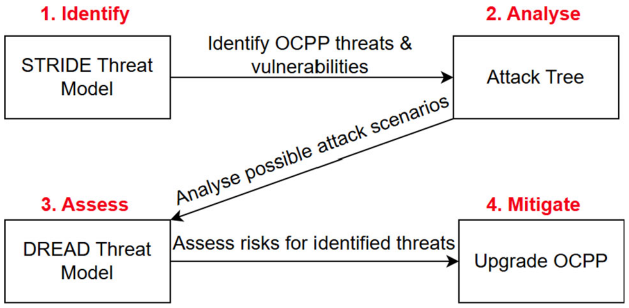

1. Identify
2. Analyse
STRIDE Threat Model
Identify OCPP threats & vulnerabilities
Attack Tree
Analyse possible attack scenarios
3. Assess
4. Mitigate
DREAD Threat Model
Assess risks for identified threats
Upgrade OCPP

Fig. 3 IAAM Framework.

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. **Analyze:** Once vulnerabilities are identified, the next step is to analyze 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's root cause and possible escalation paths.
3. **Assess:** After analyzing vulnerabilities, the framework assesses the severity and risk of each threat using the DREAD model. This phase evaluates the potential damage, reproducibility, exploitability, 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.

3.1 Identify Threat: STRIDE

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 3. Understanding these following threats is crucial for developing effective security measures to mitigate vulnerabilities in OCPP:

Th1: Cloning Attack: Attackers clone a Charge Point, enabling unauthorized access to EVs or EVCS. This can lead to unauthorized charging, data manipulation, or even control over an EV.

Th2: Identity Theft: This threat targets the Backend Server, where attackers steal or reuse credentials from OCPP communications. This allows them to impersonate legitimate entities, gaining unauthorized access to system functionalities.

Th3: Replay Attack: Attackers exploit the Communication Channel by capturing and resending OCPP messages (e.g., Start Transaction requests). This allows them to initiate unauthorized charging sessions or disrupt legitimate operations.

Th4: Data Tampering: Attackers modify OCPP Communication Messages exchanged between EVs and EVCS, leading to erroneous transactions, incorrect billing, or altered charging parameters.

Th5: Firmware Injection: Targeting the Charge Point Firmware, attackers exploit vulnerabilities in OCPP to inject unauthorized firmware updates via the communication channel. This can compromise the integrity of the charging station.

Th6: Payment Fraud: This threat affects Transaction Records, where compromised EVCS or OCPP systems manipulate transaction details, causing financial discrepancies, unauthorized billing, or denial of legitimate payments.

Th7: Data Breach: Weak encryption or inadequate security in User Data storage and transmission can expose sensitive vehicle or user information. Although TLS 1.1 is used, it is considered less secure, making it a potential attack vector.

Th8: DoS: Attackers flood the Charging Infrastructure with excessive connection requests through the OCPP interface, causing disruptions in service availability and preventing legitimate users from charging.

Th9: Privilege Escalation: Due to the lack of authentication during the Access Control System boot notification, attackers can bypass security controls and gain unauthorized high-level privileges, leading to system misuse.

3.2 Analyze threat

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 CS. The Table 4 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 5. 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. 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 3 Threats Mapped to STRIDE Categories

STRIDE Category	Threat Name	Threat Actor	Targeted Component
Spoofing	Th1: Cloning Attack	EV / EVCS	Charge Point
	Th2: Identity Theft	OCPP	Backend Server
	Th3: Replay Attack	OCPP	Communication Channel
Tampering	Th4: Data Tampering	OCPP	Communication Messages
	Th5: Firmware Injection	EV	Charge Point Firmware
Repudiation	Th6: Payment Fraud	OCPP	Transaction Records
Information Disclosure	Th7: Data Breach	OCPP	User Data
Denial of Service	Th8: DoS	OCPP	Charging Infrastructure
Elevation of Privilege	Th9: Privilege Escalation	OCPP	Access Control System

Table 4 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

Table 5 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

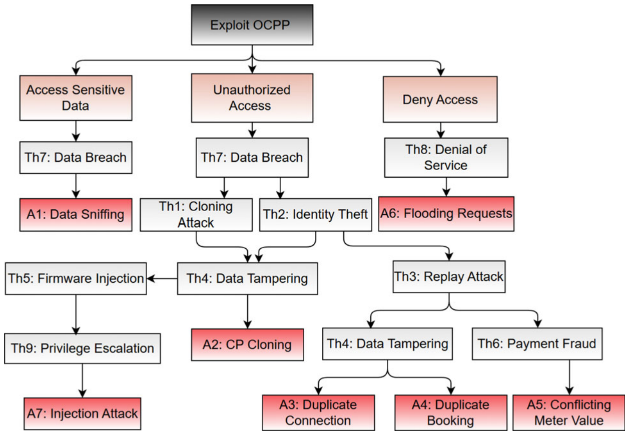


Fig. 4 OCPP Attack Tree.

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.

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).

These attacks arise from identified vulnerabilities, as outlined in the Table 6. 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:** 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 CS establish a connection using the BootNotification message. This lack of authentication means that anyone can attempt to connect to a CS, 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 CS, attackers can initiate multiple Connect.Request messages without any verification. This ability to flood the system with connection requests can overwhelm the CS 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 CS, further jeopardizing the overall security and functionality of the EV ecosystem.

Table 6 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 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 vulnerabilities highlight the urgent need for improved security measures within OCPP to protect against potential attacks and ensure the integrity and reliability of EVCS. The attack workflow is illustrated in Fig. 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). For this attack workflow, it is assumed that due to weak TLS encryption, A1: Data Sniffing has occurred. The attacker intercepts sensitive

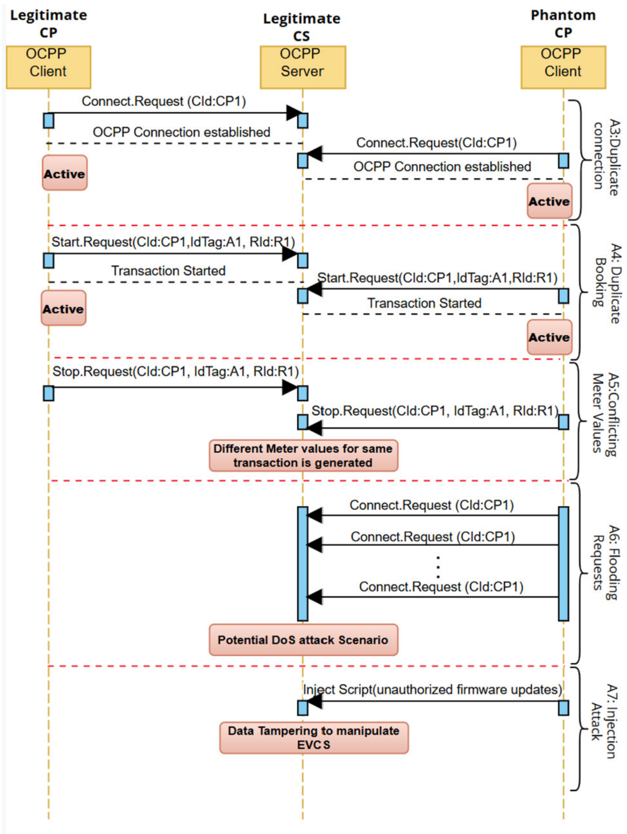

Legitimate CP
OCPP Client
Legitimate CS
OCPP Server
Phantom CP
OCPP Client
Connect.Request (CId:CP1)
OCPP Connection established
Connect.Request(CId:CP1)
OCPP Connection established
Active
Active
A3:Duplicate connection
Start.Request(CId:CP1,IdTag:A1, RId:R1)
Transaction Started
Start.Request(CId:CP1,IdTag:A1,RId:R1)
Transaction Started
Active
Active
A4: Duplicate Booking
Stop.Request(CId:CP1, IdTag:A1, RId:R1)
Stop.Request(CId:CP1, IdTag:A1, RId:R1)
Different Meter values for same transaction is generated
A5:Conflicting Meter Values
Connect.Request (CId:CP1)
Connect.Request (CId:CP1)
Connect.Request (CId:CP1)
Potential DoS attack Scenario
A6: Flooding Requests
Inject Script(unauthorized firmware updates)
Data Tampering to manipulate EVCS
A7: Injection Attack

Fig. 5 Attack workflow of OCPP vulnerabilities.

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, creating a phantom CP 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 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 backend's 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.

3.3 Assess Threat:DREAD

DREAD is a security risk assessment model used to evaluate and prioritize threats by categorizing them into five distinct factors [45], [46].

Table 7 Assessing OCPP Attacks with DREAD

OCPP Attack	Damage Potential	Reproducibility	Exploitability	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 exploitability 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.

The Table 7 summarizes how each attack maps to the DREAD categories, providing insights into their potential damage, reproducibility, exploitability, 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 [45], [46]:

- **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.
- **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.

Table 8 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)

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 [45].

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

$$\text{Likelihood} = \frac{\sum_{i=1}^{3} Y_i}{3} \tag{2}$$

$$\text{Risk} = \frac{(\text{Impact} + \text{Likelihood})}{2} \tag{3}$$

Where:

$$X_1 = \text{Damage Potential}, \quad X_2 = \text{Affected Users}$$
$$Y_1 = \text{Reproducibility}, \quad Y_2 = \text{Exploitability}$$
$$Y_3 = \text{Discoverability}$$

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.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 exploitability 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 exploitability (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), exploitability (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 exploitability.

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.

3.4 Mitigate Threat: Upgrade in OCPP

This section addresses the vulnerabilities identified in the OCPP communication protocol based on the attacks analyzed 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.

1. **OCPP Boot Notification Request:** The Boot Notification process lacks user authentication, allowing a

Algorithm 1: Mitigation for Boot Notification

```
Input: $CPModel, CPVendor, User\_IdTag,$
       $Reservation\_Id, CP\_Id$
Output: Secure Connection
$CP_A \rightarrow CS$: Send token with connection request.
$CS \rightarrow CP_A$: Check if $CP_A$ has a current booking.
If Token valid ($CP_A$ has valid booking):
    Send OTP to user and save OTP on server with expiry.
Else:
    Reject Connection.
$CP_A \rightarrow CS$: Sends BootNotification with OTP.
BootNotification.req($CPModel_A, CPVendor_A,$
  $User\_IdTag_A, Reservation\_Id_A, CP\_Id_A, OTP$).
$CS$: Verify OTP against saved session data.
If OTP valid:
    Establish Connection.
Else:
    End Connection.
If $CP_A$ sends multiple token requests:
    Limit requests to $M$ per time interval.
```

phantom CP_{A_i} to clone a legitimate CP_A and send identical BootNotification requests. This leads to an A3: Duplicate connection attack as the CS accepts both connections. Additionally, the phantom CP_{A_i} can flood the CS with repeated BootNotification requests, causing an A6: Flooding request attack. The absence of rate limiting makes the CS vulnerable to DoS attacks due to excessive requests.

In the proposed mitigation to overcome the vulnerability identified in the Boot Notification process as shown in Algorithm 1, 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 WebSocket connection with the CS. The CS 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 CS 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 CS. 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.

2. **OCPP Start Transaction Request:** The Start Transaction vulnerability arises when a phantom CP_{A_i}, with access to a legitimate reservation ID, sends a valid Start-Transaction request. Since the CS cannot distinguish between legitimate and phantom requests, it accepts multiple transactions under the same reservation, leading to an A4: Duplicate booking attack. The lack of verification beyond basic authentication enables this exploitation.
   The mitigation for the vulnerability in the Start Transaction Request as shown in Algorithm 2, focuses on preventing $A4$: Duplicate Booking by implementing checks at the CS to ensure only one transaction is allowed per reservation at any given time. When a legitimate CP_A sends a StartTransaction request, the CS first verifies whether the $ReservationId_A$ is already active in another session. If it is, the CS 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 CS proceeds to assign a new unique $TransactionId$ to the session, ensuring that each transaction has a distinct identifier. The CS 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.
3. **OCPP Stop Transaction Request:** The Stop Transaction vulnerability occurs when a phantom CP_{A_i}, sharing a legitimate reservation ID, sends a StopTransaction request. The CS processes both legitimate and phantom requests, summing all meter stop values under the same reservation. This leads to A5: Conflicting Meter Values, where the legitimate user is overcharged due to phantom-reported usage, exploiting the lack of transaction differentiation.
   The mitigation for the Stop Transaction Request vulnerability shown in Algorithm 3 focuses on ensuring that only verified sessions contribute to the final meter value calculation. By implementing an extra level of authentication during the Boot Notification, 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 CS 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 boot notification. 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.

Algorithm 2: Mitigation for Start Transaction Request

```
Input: $CP\_Id$, $User\_IdTag$, $Reservation\_Id$, $Timestamp$
Output: Mitigation: Preventing $A4$: Duplicate Booking
$CP_A$ → CS: Sends StartTransaction Request
StartTransaction.req($CP\_Id_A$, $User\_IdTag_A$, $Reservation\_Id_A$, $Timestamp_A$)
CS: Checks if Reservation_Id is already active.
if Reservation_Id is in an active session then
    CS → $CP_A$: Status = Rejected.
else
    while New Transaction Request do
        CS: Assigns a new Transaction_Id.
        CS → $CP_A$: Status = Accepted.
        (Reservation_Id,Session_Id) → new Transaction_Id.
Result→ One transaction per session and per reseravtion is active at a time
```

Algorithm 3: 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
Legit $CP_A$ → CS: Sends StopTransaction Request
StopTransaction.req($Transaction\_Id_A$, $User\_IdTag_A$, $Timestamp_A$, $MeterStop$)
foreach Transaction with $Reservation\_Id_A$ do
    if Session is verified then
        MeterValue = $\sum_{j=1}^{N} MeterStop_j$
Result: Mitigation reduces the risk of phantom transactions affecting the legitimate user's bill.
```

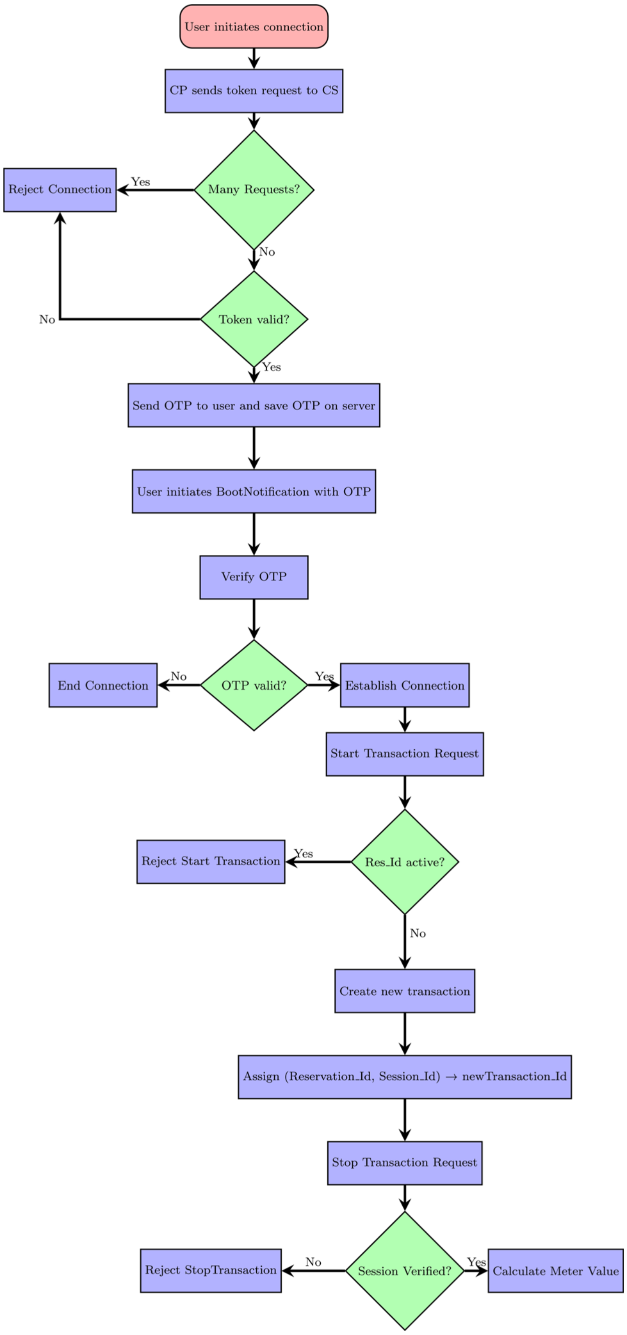


Fig. 6 Flowchart to show updated OCPP Mitigation Logic.

The flowchart depicted in Fig. 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 CS, 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.

4 Result analysis

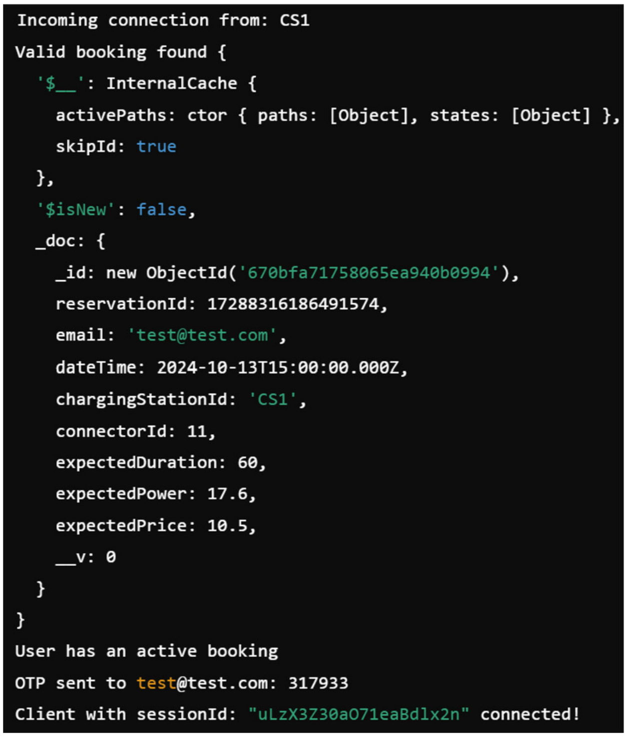
This section presents an analysis of the results achieved by upgrading the OCPP communication workflow, focusing on server log analysis to demonstrate how the implemented security enhancements address the identified vulnerabilities. Additionally, the experimental setup is included to provide context for the testing environment. Furthermore, the implementation cost and performance impact of these upgrades are discussed.

4.1 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 WebSocket 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 [44]. 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 softwares 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.



```
Incoming connection from: CS1
Valid booking found {
  '$__': InternalCache {
    activePaths: ctor { paths: [Object], states: [Object] },
    skipId: true
  },
  '$isNew': false,
  _doc: {
    _id: new ObjectId('670bfa71758065ea940b0994'),
    reservationId: 17288316186491574,
    email: 'test@test.com',
    dateTime: 2024-10-13T15:00:00.000Z,
    chargingStationId: 'CS1',
    connectorId: 11,
    expectedDuration: 60,
    expectedPower: 17.6,
    expectedPrice: 10.5,
    __v: 0
  }
}
User has an active booking
OTP sent to test@test.com: 317933
Client with sessionId: "uLzX3Z30a071eaBdlx2n" connected!
```

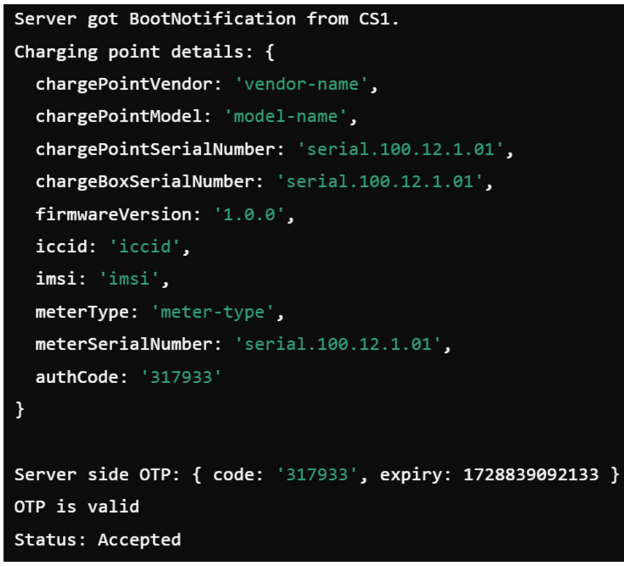
Fig. 7 Server log for Initial connection validation.

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.

4.2 Server log analysis

The server logs capture the interactions between the CP and the CS under the upgraded framework, highlighting key improvements in authentication, session management, and secure transaction handling. Key steps in the process involve the use of Boot Notification, 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. 7. This additional verification step significantly



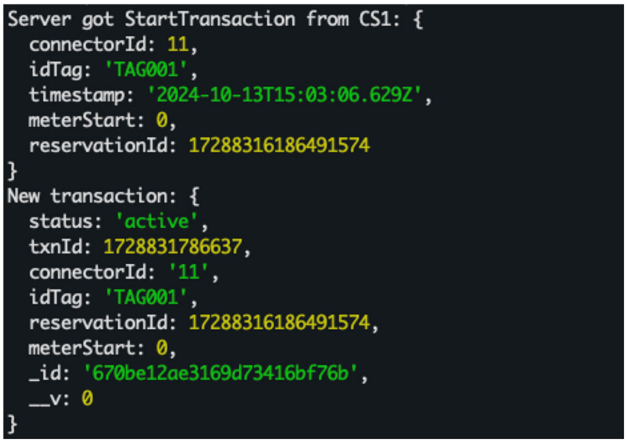
```
Server got BootNotification from CS1.
Charging point details: {
  chargePointVendor: 'vendor-name',
  chargePointModel: 'model-name',
  chargePointSerialNumber: 'serial.100.12.1.01',
  chargeBoxSerialNumber: 'serial.100.12.1.01',
  firmwareVersion: '1.0.0',
  iccid: 'iccid',
  imsi: 'imsi',
  meterType: 'meter-type',
  meterSerialNumber: 'serial.100.12.1.01',
  authCode: '317933'
}

Server side OTP: { code: '317933', expiry: 1728839092133 }
OTP is valid
Status: Accepted
```

Fig. 8 Server log for Boot notification.

mitigates vulnerabilities such as CP Cloning and Duplicate Booking (A2 and A4). Before the Boot Notification 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 CS.

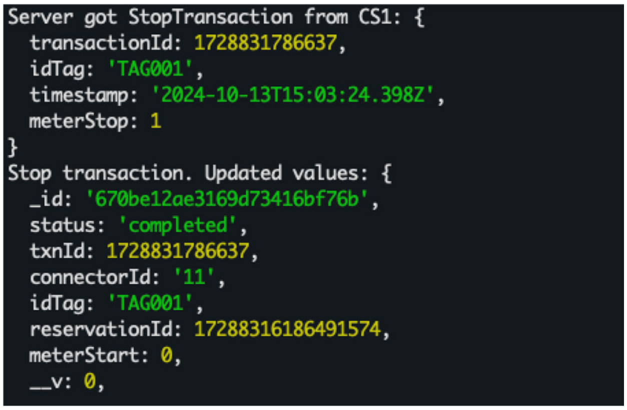
2. **Boot Notification Security Enhancement:** As shown in Fig. 8, the Boot Notification request and response between the CP and CS now include an OTP as an additional security parameter. This enhancement introduces an extra layer of authentication, which was not part of the original OCPP specification. Traditionally, the Boot Notification messages were primarily used to exchange information about the CP's hardware, firmware, and network status with the CS. 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 Boot Notification message now incorporates an authCode field that holds the OTP value. Upon receiving the Boot Notification, the CS 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.
3. **Improved Session Management:** In the Start Transaction process as shown in Fig. 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.



```
Server got StartTransaction from CS1: {
  connectorId: 11,
  idTag: 'TAG001',
  timestamp: '2024-10-13T15:03:06.629Z',
  meterStart: 0,
  reservationId: 17288316186491574
}
New transaction: {
  status: 'active',
  txnId: 1728831786637,
  connectorId: '11',
  idTag: 'TAG001',
  reservationId: 17288316186491574,
  meterStart: 0,
  _id: '670be12ae3169d73416bf76b',
  __v: 0
}
```

Fig. 9 Server log for Start Transaction.

4. **Transaction Termination Validation:** In the Stop Transaction process as shown in Fig. 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 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 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



```
Server got StopTransaction from CS1: {
  transactionId: 1728831786637,
  idTag: 'TAG001',
  timestamp: '2024-10-13T15:03:24.398Z',
  meterStop: 1
}
Stop transaction. Updated values: {
  _id: '670be12ae3169d73416bf76b',
  status: 'completed',
  txnId: 1728831786637,
  connectorId: '11',
  idTag: 'TAG001',
  reservationId: 17288316186491574,
  meterStart: 0,
  __v: 0,
```

Fig. 10 Server log for Stop Transaction.

’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.

5. **Continuous Monitoring with Meter Values:**In the Meter Values process as shown in Fig. 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.

4.3 Implementation cost and performance impact

While the proposed enhancements to the OCPP protocol, such as multi-factor authentication (OTP), session management, and request filtering, significantly improve the security posture of the EV charging infrastructure, it is important to consider their impact on system performance.

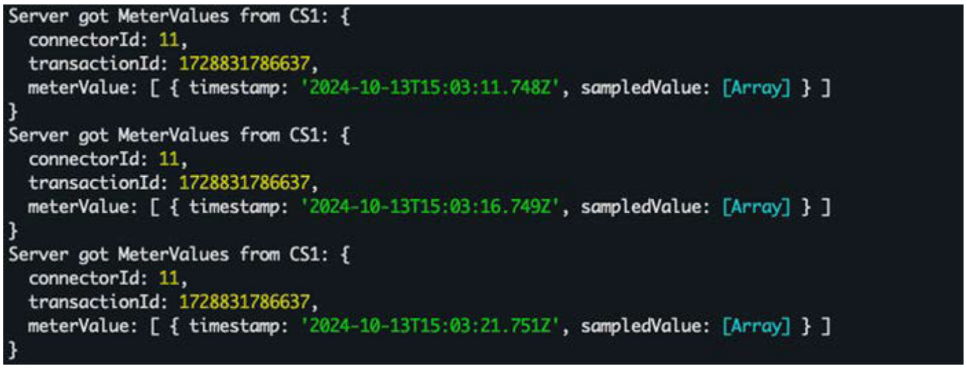
These security mechanisms, though crucial for protecting the infrastructure from cyber threats, may incur additional computational overhead. The OTP-based authentication process, for example, requires generating and verifying time-sensitive one-time passcodes, which could increase the time required for user authentication during the Boot Notification process [47]. Similarly, the session management enhancements, including duplicate connection and booking checks, require additional communication overhead to track and validate sessions, potentially increasing latency [48].

For resource-constrained devices such as charging stations with limited processing power and memory, these additional processes could affect the overall response time and throughput of the system [49]. However, we note that the computational cost of these mechanisms is generally low when compared to the security benefits they provide [50]. OTP authentication can be optimized by utilizing lightweight cryptographic algorithms [49] and performing some operations offloaded to more capable backend servers, thus minimizing the computational burden on the charging station itself.

To further mitigate performance concerns, request filtering mechanisms can be implemented selectively, such as only applying heavy filtering techniques for high-risk requests or during peak usage times [51]. This approach ensures that security enhancements do not excessively degrade system performance during periods of normal operation. While there is an inherent trade-off between security and performance, the proposed security measures provide a robust defense against a wide range of cyber threats, ensuring that the EV charging infrastructure remains both secure and operational. For resource-constrained environments, additional strategies like server-side processing [51] and lightweight cryptography [49] could be employed to minimize the impact on system performance without sacrificing security. These enhancements are designed to be scalable and can be tailored to fit the specific capabilities of each device in the system.

5 State-of-the-art comparison

OCPP serves as a critical communication standard between CP and CS. However, various security vulnerabilities asso-



```
Server got MeterValues from CS1: {
  connectorId: 11,
  transactionId: 1728831786637,
  meterValue: [ { timestamp: '2024-10-13T15:03:11.748Z', sampledValue: [Array] } ]
}
Server got MeterValues from CS1: {
  connectorId: 11,
  transactionId: 1728831786637,
  meterValue: [ { timestamp: '2024-10-13T15:03:16.749Z', sampledValue: [Array] } ]
}
Server got MeterValues from CS1: {
  connectorId: 11,
  transactionId: 1728831786637,
  meterValue: [ { timestamp: '2024-10-13T15:03:21.751Z', sampledValue: [Array] } ]
}
```

Fig. 11 Server log for Stop Transaction.

Table 9 Comparative Analysis for OCPP communication Vulnerability

Paper	Vulnerability	Attack Type	Authentication	Session Management	Message Handling	Experiment Analysis
Proposed	OCPP communication	DoS, MitM, connection Hijacking	Yes	Yes	Yes	Yes
[12]	OCPP communication	MitM	Yes	No	Yes	Yes
[10]	OCPP communication	DoS, MitM	Yes	Yes	Yes	Yes
[8]	OCPP communications	DoS, MitM	Yes	No	Yes	No
[6]	OCPP communications	DoS, MITM, Code Injection	Yes	No	Yes	Yes
[11]	OCPP communication	DoS, MITM, Firmware Theft	Yes	No	Yes	Yes
[13]	OCPP Charge Point	Injection of malformed requests/responses	No	No	Yes	Yes

ciated 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 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.

Rubio et al. [12] 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. [10] 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. [8] explore vulnerabilities in OCPP, highlighting issues like ARP spoofing and Remote Keyless Entry 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. [6] investigate vulnerabilities in OCPP, pointing out unencrypted websocket 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. [11] 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 testbed that showcases the feasibility of attacks against the power grid via compromised EVCSs, providing critical insights into EV charging ecosystem security.Gebauer et al. [13] 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.

6 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 10. This table 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.

Charging behaviors were modeled for both Model 1 and Model 2 using the reservation entries from Table 10 to generate the transaction data found in Tables 11 and 12. Table 11 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 12 documents the CP data for Model 2, where an enhanced version of OCPP is used. The table structure is similar to Table 11 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 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 behaviors. It captures any instances of over-consumption or extended charging beyond the expected limits. A conflicting transaction is defined as:

Table 10 Reservation Details

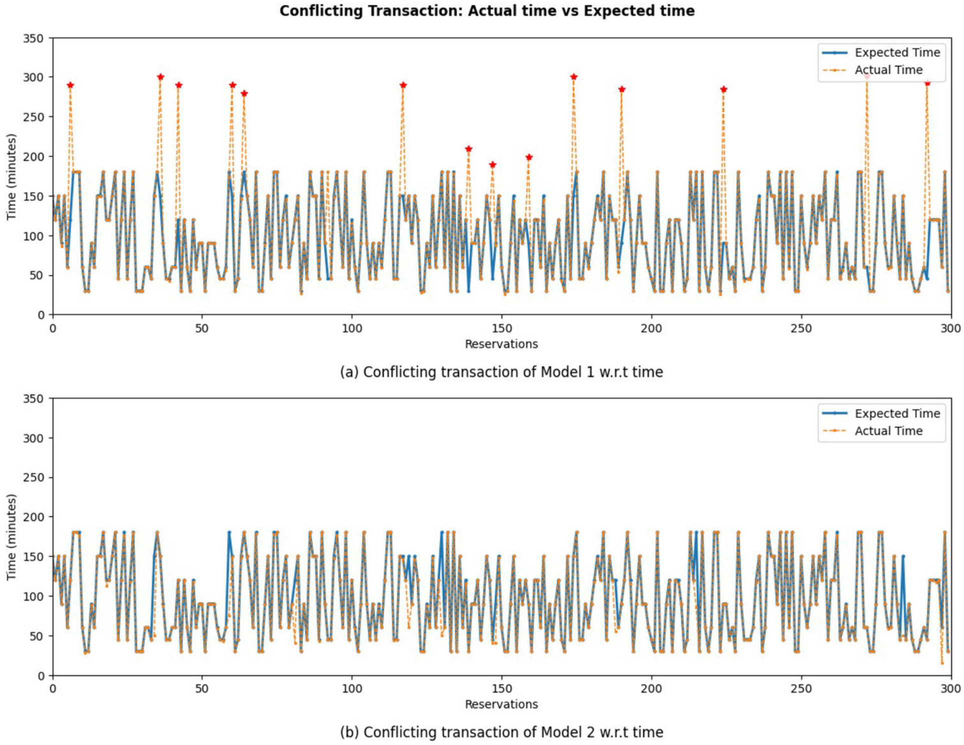
ResrvId	ResrvStartTime	ResrvEndTime	ResrvDate	ExpectedTime	ExpectedPower
R001	21:17	23:47	10/03/2024	150	44
R002	15:18	17:18	10/16/2024	120	36
R003	12:49	15:19	10/27/2024	150	44
R004	11:27	12:57	10/17/2024	90	27
R005	09:46	12:16	10/28/2024	150	44
⋮	⋮	⋮	⋮	⋮	⋮
R296	19:49	21:49	10/15/2024	120	36
R297	08:55	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

Table 11 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 12 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



(a) Conflicting transaction of Model 1 w.r.t time

(b) Conflicting transaction of Model 2 w.r.t time

Fig. 12 (a) Conflicting Transaction of model 1 w.r.t time (b) Conflicting Transaction of model 2 w.r.t time.

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

The Fig. 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. 12a, 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. 12b, 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.

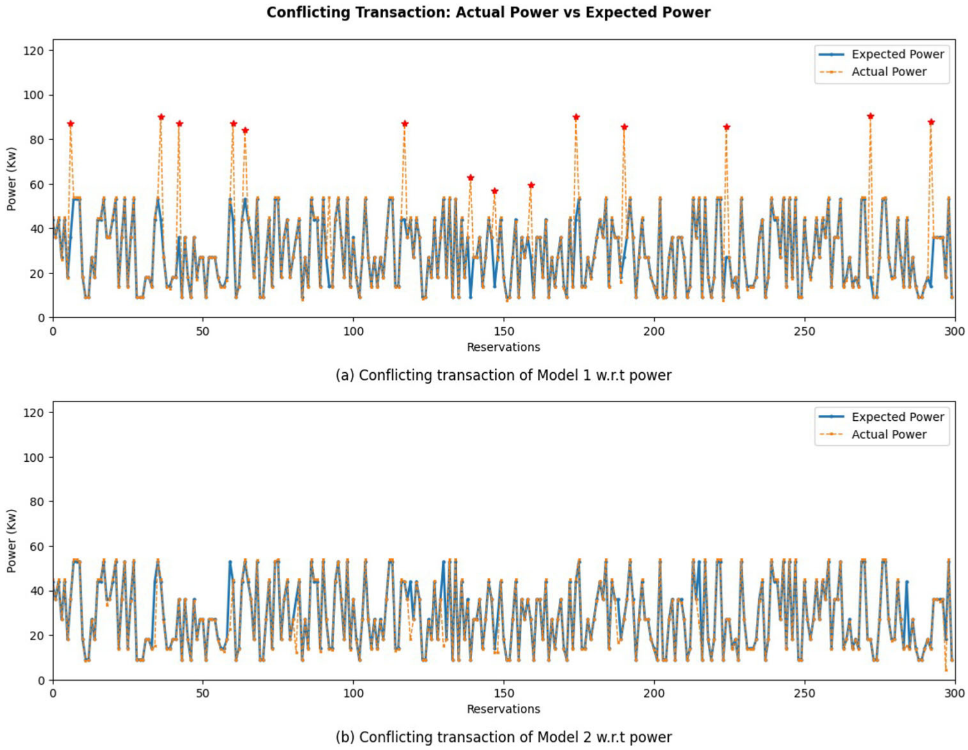
The Fig. 13 illustrates the actual power consumption versus expected power consumption for both models. Model 1 as shown in Fig. 13 a, again shows red stars, which represent instances where actual power consumption exceeds the expected levels. The excessive power consumption could also result from the 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. 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 behavior deviates from the expected behavior. Here, z=1 is a constant accounting for external factors or additional error margins that may influence the fault rate.The formula is:

$$\text{Fault Detection Rate} = \frac{\text{Number of Conflicting Transactions} + z}{\text{Total Number of Reservations}}$$

The fault detection rate shown in Fig. 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

Fig. 13 (a) Conflicting Transaction of model 1 w.r.t power (b) Conflicting Transaction of model 2 w.r.t power.



(a) Conflicting transaction of Model 1 w.r.t power

(b) Conflicting transaction of Model 2 w.r.t power

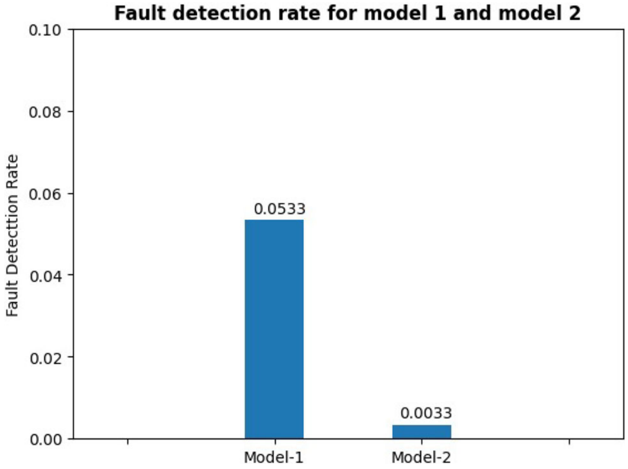


Fig. 14 Fault detection rate.

charging behavior 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 behavior in line with the expected values and providing greater reliability.

7 Conclusion

This paper has identified and thoroughly analyzed the critical security vulnerabilities in OCPP, a widely adopted communication protocol for EVCS. The vulnerabilities, including weak session management, inadequate authentication, and poor message handling, expose the EV charging infrastructure to various risks such as DoS attacks, unauthorized access, and transaction tampering. Through the application of the STRIDE and DREAD frameworks, these vulnerabilities were mapped to potential threats, and their severity was assessed. In response to these issues, the paper proposes a set of enhancements, including multi-factor authentication via OTP during the Boot Notification process, improved session management to prevent duplicate connections and bookings, and the incorporation of measures to validate sessions and meter values. These mitigations, outlined in the proposed algorithms, aim to strengthen the security of OCPP and ensure a more robust and reliable EV charging infrastructure.

While the paper focuses on addressing these vulnerabilities through the IAAM framework, it primarily provides a systematic risk assessment approach. This approach goes beyond just detecting anomalies. However, future research could explore integrating AI/ML-based anomaly detection techniques and behavioral monitoring to complement the IAAM solution. Behavioral monitoring could increase resistance to unknown threats by identifying unusual patterns in OCPP traffic that may not fit predefined attack models. By dynamically detecting emerging threats in real time, this integration could significantly enhance the system's ability to respond to evolving attack tactics. Ultimately, this work contributes to improving the security of OCPP-based EVCS infrastructure, offering solutions that can be implemented without requiring costly hardware upgrades, thereby ensuring that the infrastructure remains secure and operational as the demand for electric mobility grows.

Acknowledgements The work is supported by the Computer Science and Informatics Research Centre (CIRC), Nottingham Trent University. Also, The authors extend their appreciation to the Deanship of Scientific Research at Zarqa University, Zarqa, Jordan, for funding this research work.

Author Contributions H. has developed, implemented the research, and wrote a draft of the research paper. O.K. has supervised the research project along with D.B., D.J., M.J. as Co-supervisors. M. A., and R.K. have validated the framework setup for the research. Y.C. and S.K. have validated the results and done a critical analysis. O.K., D.B., D.J., M.J., M. A., Y.C., R.K. and S.K. have edited the final version of the paper.

Funding The research work is funded by industry (JMVL Ltd) and Nottingham Trent University, UK.

Data Availibility Statement No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Muratori,M., Alexander,M., Arent,D., Bazilian,M., Cazzola,P., Dede,E. M., Farrell,J., Gearhart,C., Greene,D., Jenn,A. et al.: The rise of electric vehicles-2020 status and future expectations, Progress in Energy, vol. 3, no. 2, pp. 022002, 2021, IOP Publishing
2. Victormunoz. OCPP-1.6-Chargebox-Simulator: A Simple Chargepoint Simulator, Working with OCPP 1.6. Available online: https://github.com/victormunoz/OCPP-1.6-Chargebox-Simulator (accessed on 3 August 2023)
3. 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)
4. 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, 2024
5. Open Charge Alliance, Open charge point protocol 1.6, 2015, July 2016
6. Johnson,J., II,D. E., Fragkos,G., Zhang,J., Rohde,K. W., Salinas,S. C.: 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, USA, Tech. Rep., 2023
7. Hansen,M. E.: Implementation and test of the open charge point protocol in an autonomous charger for electric vehicles, DTU Wind-M-0756, 2024
8. Garofalaki,Z., Kosmanos,D., Moschoyiannis,S., Kallergis,D., Douligeris,C.: Electric vehicle charging: A survey on the security issues and challenges of the open charge point protocol (OCPP), IEEE Communications Surveys & Tutorials, vol. 24, no. 3, pp. 1504–1533, 2022, IEEE
9. Saposnik,L. R.: Hijacking EV charge points to cause DOS, Saiflow, 2024. [Online]. Available: https://www.saiflow.com/blog/hijacking-chargers-identifier-to-cause-dos/. Accessed: June 27, 2024
10. Alcaraz,C., Lopez,J., Wolthusen,S.: OCPP protocol: Security threats and challenges, IEEE Transactions on Smart Grid, vol. 8, no. 5, pp. 2452–2459, 2017, IEEE
11. Sarieddine,K., Sayed,M. A., Torabi,S., Attallah,R., Jafarigiv,D., Assi,C., Debbabi,M.: Uncovering covert attacks on EV charging infrastructure: How OCPP backend vulnerabilities could compromise your system, ACM, 2024
12. Rubio,J. E., Alcaraz,C., Lopez,J.: Addressing security in OCPP: Protection against man-in-the-middle attacks, in Proc. 9th IFIP Int. Conf. New Technol., Mobility and Security (NTMS), 2018, pp. 1–5, IEEE
13. Gebauer,L., Trsek,H., Lukas,G.: Evil SteVe: An approach to simplify penetration testing of OCPP charge points, in Proc. IEEE 27th Int. Conf. Emerging Technol. Factory Automation (ETFA), 2022, pp. 1–4, IEEE
14. CCC, Committee on Climate Change's (CCC), Net Zero: The UK's contribution to stopping global warming, May 2019, Online. Available: https://www.theccc.org.uk/wp-content/uploads/2019/05/Net-Zero-The-UKs-contribution-to-stopping-global-warming.pdf. Accessed: Jan. 22, 2023
15. 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. Sensors **23**(15), 6716 (2023)
16. Hussain, S., Kamal, A., Ahmad, S., Rasool, G., Iqbal, S.: Threat modelling methodologies: a survey. Sci. Int. (Lahore) **26**(4), 1607–1609 (2014)
17. R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, STRIDE-based threat modeling for cyber-physical systems, in Proc. IEEE PES Innov. Smart Grid Technol. Conf. Europe (ISGT-Europe), 2017, pp. 1–6
18. K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and others, Experimental security analysis of a modern automobile, in Proc. IEEE Symp. Security Privacy, 2010, pp. 447–462
19. Rouf,I., Miller,R., Mustafa, H., Taylor,T., Oh,S., Xu,W., Gruteser,M., Trappe,W., Seskar,I.: Security and privacy vulnerabil-

ities of in-car wireless networks: A tire pressure monitoring system case study, in Proc. USENIX Security Symp. (USENIX Security 10), 2010
20. Checkoway,S., McCoy,D., Kantor,B., Anderson,D., Shacham,H., Savage,S., Koscher,K., Czeskis,A., Roesner,F., Kohno,T.: Comprehensive experimental analyses of automotive attack surfaces, in Proc. USENIX Security Symp. (USENIX Security 11), 2011
21. Woo, S., Jo, H.J., Lee, D.H.: A practical wireless attack on the connected car and security protocol for in-vehicle CAN. IEEE Trans. Intell. Transp. Syst. **16**(2), 993–1006 (2014)
22. Manderna, A., Kumar, S., Dohare, U., Aljaidi, M., Kaiwartya, O., Lloret, J.: Vehicular network intrusion detection using a cascaded deep learning approach with multi-variant metaheuristic. Sensors **23**(21), 8772 (2023)
23. Jafarnejad,S., Codeca,L., Bronzi,W., Frank,R., Engel,T.: A car hacking experiment: When connectivity meets vulnerability, in Proc. IEEE Globecom Workshops (GC Wkshps), 2015, pp. 1–6
24. Garcia,F. D., Oswald,D., Kasper,T., Pavlidès,P.: Lock it and still lose it-on the (in) security of automotive remote keyless entry systems, in Proc. USENIX Security Symp. (USENIX Security 16), 2016
25. Mazloom,S., Rezaeirad,M., Hunter,A., McCoy,D.: A security analysis of an in-vehicle infotainment and app platform, in 10th USENIX Workshop on Offensive Technologies (WOOT 16), 2016
26. Karthik,T., Brown,A., Awwad,S., McCoy,D., Bielawski,R. Mott,C., Lauzon,S., Weimerskirch,A., Cappos,J.: Uptane: Securing software updates for automobiles, in International Conference on Embedded Security in Car, 2016, pp. 1–11
27. Currie,R.: Hacking the CAN bus: Basic manipulation of a modern automobile through CAN bus reverse engineering, SANS Institute, 2017
28. Luo, Q., Liu, J.: Wireless telematics systems in emerging intelligent and connected vehicles: Threats and solutions. IEEE Wireless Commun. **25**(6), 113–119 (2018)
29. Jouvray, C., Pellischek, G., Tiguercha, M.: Impact of a smart grid to the electric vehicle ecosystem from a privacy and security perspective. World Electric Vehicle Journal **6**(4), 1115–1124 (2013)
30. Schneider Electric, EVLink Parking, 2018. [Online]. Available: https://us-cert.cisa.gov/ics/advisories/ICSA-19-031-01. Accessed: Sep. 8, 2024
31. Circontrol, CirCarLife, 2019. [Online]. Available: https://uscert.cisa.gov/ics/advisories/ICSA-18-305-03. Accessed: Sep. 18, 2024
32. Van Aubel,P., Poll,E., Rijneveld,J.: Non-repudiation and end-to-end security for electric-vehicle charging, in 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), 2019, pp. 1–5
33. Antoun, J., Kabir, M.E., Moussa, B., Atallah, R., Assi, C.: A detailed security assessment of the EV charging ecosystem. IEEE Network **34**(3), 200–207 (2020)
34. Acharya, S., Dvorkin, Y., Karri, R.: Public plug-in electric vehicles+ grid data: Is a new cyberattack vector viable? IEEE Transactions on Smart Grid **11**(6), 5099–5113 (2020)
35. Girdhar, M., Hong, J., Lee, H., Song, T.-J.: Hidden Markov models-based anomaly correlations for the cyber-physical security of EV charging stations. IEEE Transactions on Smart Grid **13**(5), 3903–3914 (2021)
36. Carryl, C., Ilyas, M., Mahgoub, I., Rathod, M.: The PEV security challenges to the smart grid: Analysis of threats and mitigation strategies, in. International Conference on Connected Vehicles and Expo (ICCVE) **2013**, 300–305 (2013)
37. Baker,R., Martinovic,I.: Losing the car keys: Wireless PHY-layer insecurity in EV charging, in 28th USENIX Security Symposium (USENIX Security 19), 2019, pp. 407–424
38. Sayed, M.A., Atallah, R., Assi, C., Debbabi, M.: Electric vehicle attack impact on power grid operation. International Journal of Electrical Power & Energy Systems **137**, 107784 (2022)
39. Khan, O.G.M., El-Saadany, E., Youssef, A., Shaaban, M.: Impact of electric vehicles botnets on the power grid, in. IEEE Electrical Power and Energy Conference (EPEC) **2019**, 1–5 (2019)
40. Mokarim,A., Gaggero,G. B., Marchese,M.: Evaluation of the impact of cyber-attacks against electric vehicle charging stations in a low voltage distribution grid, in 2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2023, pp. 1-7
41. Nasr,T., Torabi,S., Bou-Harb,E. Fachkha,C., Assi,C.: Power jacking your station: In-depth security analysis of electric vehicle charging station management systems, Computers Security, vol. 112, pp. 102511, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404821003357
42. Momson,G. S.: Threats and mitigation of DDoS cyberattacks against the U.S. power grid via EV charging, 2018
43. Deb,S., Tammi,K., Kalita,K. Mahanta,P.: Impact of electric vehicle charging station load on distribution network, Energies, vol. 11, no. 1, 2018, [online] Available: https://www.mdpi.com/1996-1073111/1/178
44. Hamdare,S.: ocpp-rpc: OCPP schema implementation using JSON in Node.js, GitHub, 2025. [Online]. Available: https://github.com/SHamdare/ocpp-rpc
45. Kavallieratos, G., Katsikas, S.: Managing cyber security risks of the cyber-enabled ship. Journal of Marine Science and Engineering **8**(10), 768 (2020)
46. Alcaraz,C., Cumplido,J., Trivino,A.: OCPP in the spotlight: Threats and countermeasures for electric vehicle charging infrastructures 4.0, International Journal of Information Security, **22**(5), 1395–1421, (2023)
47. Alsheavi, A.N., Hawbani, A., Othman, W., Wang, X., Qaid, G., Zhao, L., Al-Dubai, A., Zhi, L., Ismail, A., Jhaveri, R., Alsamhi, S.: IoT Authentication Protocols: Challenges, and Comparative Analysis. ACM Computing Surveys **57**(5), 1–43 (2025)
48. Hiller,J., Henze,M., Zimmermann,T., Hohlfeld,O., Wehrle,K.: The case for session sharing: relieving clients from TLS handshake overheads, in 2019 IEEE 44th LCN Symposium on Emerging Topics in Networking (LCN Symposium), pp. 83-91, IEEE, Oct. 2019
49. Thakor, V.A., Razzaque, M.A., Khandaker, M.R.: Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities. IEEE Access **9**, 28177–28193 (2021)
50. Melki, R., Noura, H.N., Chehab, A.: Lightweight multi-factor mutual authentication protocol for IoT devices. International Journal of Information Security **19**(6), 679–694 (2020)
51. Verma, P., Tapaswi, S., Godfrey, W.W.: An adaptive threshold-based attribute selection to classify requests under DDoS attack in cloud-based systems. Arabian Journal for Science and Engineering **45**, 2813–2834 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.