ELUDE: Secure EV Charging Point Prototype with OCPP 2.0.1 and TLS Authentication

Prof. D. B. Salunke[1] Prasad Tatar[2] Adesh Gaikwad[3] Priya Veer[4]
[1,2,3,4]Department of Electronics and Telecommunication Engineering
[1,2,3,4]Jayawantrao Sawant College of Engineering, Hadapsar, Pune, Maharashtra, India

Abstract — Electric Vehicle charging infrastructure re-quires reliable and secure communication between charging stations and backend management systems to ensure safe operation, accurate billing and continuous service availability. This paper presents a software-based implementation of a secure communication framework for EV charging stations using the Open Charge Point Protocol (OCPP) version 2.1. The proposed system establishes a protected data exchange channel between the charging station and a centralized backend server through authentication, message validation, and encrypted communication mechanisms. The architecture is designed to handle core OCPP functionalities such as device registration, transaction management, status notifications, and remote-control commands while maintaining compliance with protocol specifications. A modular server-side application processes incoming messages, verifies client identity, and manages session-level communication in real time. Security measures are integrated at multiple levels, including certificate-based authentication and secure WebSocket communication, to prevent unauthorized access and data tampering. Experimental testing demonstrates stable connectivity, ac-curate message handling, and consistent performance under continuous operation. The results indicate that the proposed implementation provides a dependable and scalable solution for managing EV charging stations while ensuring secure and standardized communication. This work highlights the practical applicability of OCPP 2.1 for modern charging infrastructure and contributes toward building interoperable and secure EV ecosystems.

Keywords: OCPP 2.1, Electric Vehicle Charging Station, Secure Communication, WebSocket Protocol, Backend Server, Authentication, TLS Encryption, Message Validation, Software Architecture and Interoperability]

I. INTRODUCTION

The rapid growth of Electric Vehicles (EVs) has led to a significant expansion of charging infrastructure across ur-ban and semi-urban regions. As charging networks increase in scale, the need for reliable and secure communication between charging stations and centralized backend systems becomes critical (Khy, 2025). These communication links are responsible for managing charging sessions, monitoring device status, processing billing information, and enabling remote control of charging operations. Any disruption or security breach in this communication channel can directly affect service availability, user trust, and system safety. To ensure interoperability among different manufacturers and service providers, the Open Charge PointProtocl (OCPP) has emerged as a widely adopted open standard for communi-cation between EV charging stations and backend management systems. The latest version, OCPP 2.1, introduces enhanced support for security features, device management, and smart charging operations (Kirchner, 2024). However, implementing a fully compliant and secure OCPP 2.1 communication framework remains a challenging task due to the complexity of protocol specifications and the need for robust authentication and data protection mechanisms (Hamdare et al., 2025; Eyada et al., 2020)

This paper presents a software-based implementation of a secure communication system using OCPP 2.1 that establishes a protected interaction between an EV charging station and a backend server. The proposed system focuses on maintaining message integrity, secure session management, and protocol-compliant data exchange over WebSocket communication. The design emphasizes modularity and scalability, allowing the system to support multiple charging stations and concurrent transactions while ensuring consistent performance (Patel et al., 2023).

In the proposed architecture, the charging station acts as a client that communicates with a centralized backend server using standardized OCPP messages such as boot notification, status notification, transaction events, and remote control commands. Security is enforced through certificate-based authentication and encrypted communication channels, preventing unauthorized access and minimizing the risk of data interception or tampering. The backend server validates each incoming request and maintains structured logs for monitoring and auditing purposes.

The main objective of this work is to demonstrate a practical and secure implementation of OCPP 2.1 for real-time EV charging management. By combining standardized protocol communication with integrated security mechanisms, the proposed system contributes toward building a dependable and interoperable EV charging ecosystem. This approach supports future expansion of charging networks and provides a foundation for secure integration with smart grid and energy management platforms.

II. ARCHITECTURE:

The architecture of the proposed system is based on the client-server communication model defined by the Open Charge Point Protocol (OCPP) 2.1, which standardizes inter-action between EV charging stations (clients) and a centralbackend management system (server). OCPP 2.1 enables interoperable and extensible messaging across diverse charging hardware and backend software, ensuring consistent operational behavior and secure communication.

In the implemented system, the Charging Station (Client) represents the EV charging unit, which establishes a persistent connection with the Backend Server (Central System). The communication channel is maintained through WebSocket protocols (WSS) to permit real-time bidirectional exchange of JSON-formatted OCPP messages. Key operations such as boot notification, status updates,

transaction events, and remote commands are exchanged via this WebSocket channel, with TLS securing data in transit.
The system architecture comprises the following major components:

1) Charging Station Module: This module implements the OCPP 2.1 client interface and handles connection establishment, message framing, and protocol compliance. It is responsible for initiating a secure WebSocket connection to the backend server, sending operational messages (e.g., BootNo-tificationRequest, StatusNotificationRequest), and processing corresponding responses. In the project repository, this is realized through a client implementation that connects to a designated backend endpoint and manages session-level communication.
2) Secure Communication Layer: All messaging between the charging station and backend occurs over TLS-protected Web-Sockets to prevent unauthorized access, tampering, or interception. The use of certificate-based authentication ensures that both endpoints verify each other's identity before exchanging OCPP messages. This provides integrity and confidentiality for all exchanged data.
3) Backend OCPP Server: The backend component acts as the central management system. It listens for incoming WebSocket connections from charging stations and maintains persistent sessions for each connected station. The server validates in-coming requests, routes messages to appropriate handlers, and stores session data for further processing. As the authoritative entity, it responds with appropriate OCPP responses and issues control commands as required. Message Routing and Processing: Upon receiving OCPP requests from the charging station, the server parses the JSON payload and triggers corresponding application logic in the backend. For example, charging session initiation is processed by transaction management components, while status notifications update the station's operational state. Response messages adhere to OCPP 2.1 specifications, ensuring uniform error handling and structured replies.\Backend Services and Storage: The backend may integrate additional services such as database persistence, authentication management, and event logging. Persistent storage holds charging station metadata, session logs, and transaction records for reporting, analytics, or billing purposes. Authentication services verify certificate credentials, while event logs provide traceability for system operation.

The architectural flow begins when the charging station boots and sends a Boot Notification Request to the backend server. Upon successful authentication, a secure session is established. The charging station periodically sends status notifications and handles transaction events (e.g., StartTransaction, Stop Transaction) as EVs connect or disconnect. The backend processes these messages, updates internal state, and may issue remote commands such as Reset, ChangeAvailability, or Configuration updates. The continuous use of WebSockets facilitates real-time data exchange, while encryption ensures secure communication throughout the system lifecycle.

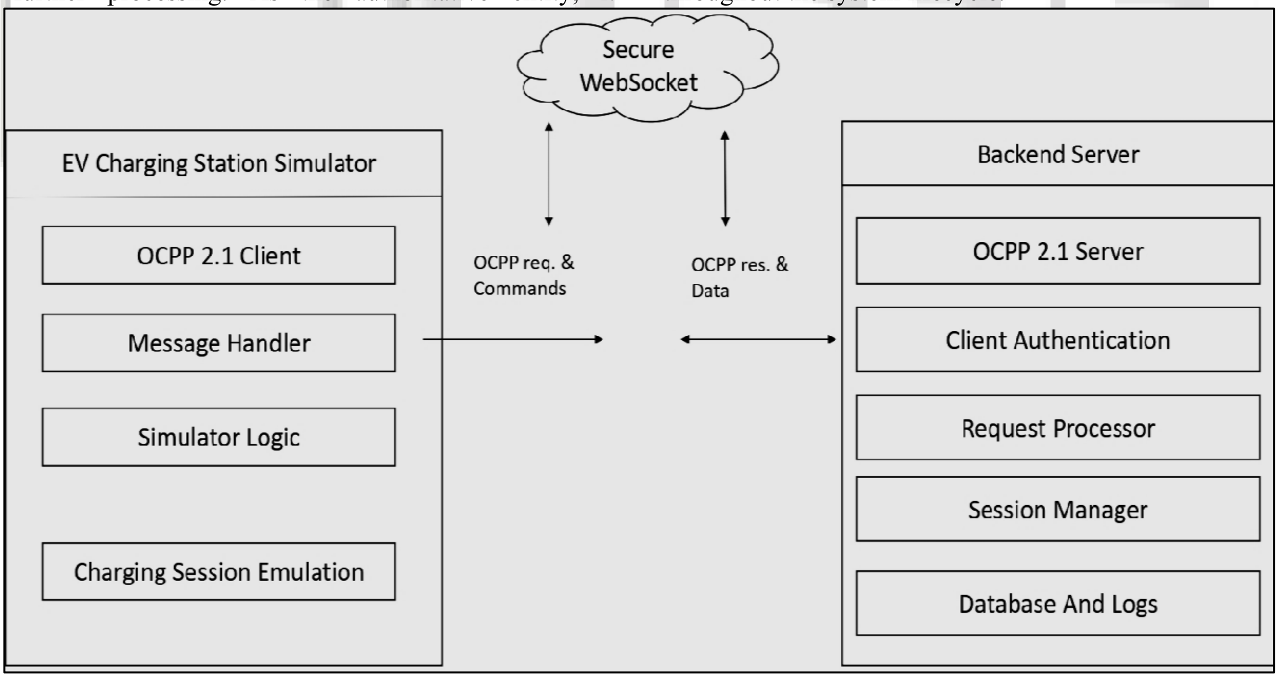

Fig. 1. Architecture of Communication System

The dashboard gives us an overview of how the system is doing. It keeps an eye on 6 chargers and the total energy they produce which is 131.00 kWh. The dashboard shows what is happening now. It looks at what's going on with the chargers and uses simple graphs to show how much energy is being used. This helps the people in charge see when the chargers are being used the most and compare how well each charger is working. They can see all of this information, about the chargers on the dashboard.

III. METHODOLOGY:

The Electric Vehicle charging station and the Central Management System use the OCPP 2.0.1/2.1 protocol to talk to each other. We want to make sure this conversation is secure. So, we made a software-based charging station simulator. This simulator acts like the thing. It helps us test the Electric Vehicle charging station and the Central Management System in a place.

First the simulator and the server say hello to each other. This is called Initialization and Mutual Authentication. The simulator and the server use Secure WebSocket protocols to talk. They show each special certificates to prove who they are. This way only the right people can talk to each other.

Next the simulator signs up with the Central Management System. This is called Registration and State Synchronization. The simulator sends a message to the Central Management System to set its clock and get ready to work. Then it sends heartbeats to the Central Management System to say it is working. This helps us see if the system can keep working and handle messages.

Finally, the simulator pretends to be an Electric Vehicle charging station, in the world. This is called Operational Simulation and Command Execution. It sends messages to the Central Management System to say when an Electric Vehicle is plugged in or out. The Central Management System gets these messages. Puts them in a database. The Central Management System can also tell the Electric Vehicle charging station what to do. We watch all of this to make sure it works.

We use a dashboard to see what is happening. We can see how long it takes for messages to get through and if they are correct. This helps us know if the Electric Vehicle charging station and the Central Management System are talking to each other correctly using the OCPP 2.0.1/2.1 protocol.

IV. IMPLEMENTATION:

A. *Software:*

The software system is implemented using a backend server and a charging station simulator that communicate using the Open Charge Point Protocol (OCPP) version 2.1 over secure WebSocket connections. The complete system is developed as a software-based platform to emulate real EV charging station behavior without physical hardware. It performs the following functions:

1) Charging Station Simulation: A software simulator is used to replicate the behavior of an EV charging station. It generates standard OCPP 2.1 messages such as Boot-Notification, StatusNotification, and TransactionEvent and sends them to the backend server for processing.
2) Secure Communication Layer: The simulator establishes a secure WebSocket (WSS) connection with the back-end server using TLS encryption and certificate-based authentication. This ensures confidentiality and integrity of all exchanged OCPP messages.
3) Message Processing Logic: The backend server validates incoming OCPP requests and processes them according to protocol specifications. It maintains session-level communication and ensures correct response generation for each request.
4) User Interface: A web-based dashboard is implemented to monitor real-time communication between the charging station simulator and the backend server. The dash-board displays connection status, transaction events, and system logs for analysis and debugging.
5) Authentication and Access Control: User authentication mechanisms are integrated into the backend system to restrict access to the monitoring dashboard. Only authorized users can view system status and control operations.
6) Logging and Error Handling: All message exchanges and events are recorded in structured log files and stored in a database for further evaluation. Error handling routines are included to manage invalid requests and unexpected communication failures.

B. *Software Simulator Environment:*

Since the proposed system focuses on protocol-level communication and security validation, physical charging hard-ware is replaced with a software-based simulator. The simulator emulates essential hardware functionalities of an EV charging station.

1) Charging Station Simulator Module: Implements OCPP 2.1 client functionalities, generates charging session events and status notifications, simulates plug-in and plug-out operations of an EV, and communicates with the backend server over secure WebSocket channels.
2) Backend Server System: Acts as the Central Management System (CMS), processes OCPP 2.1 requests and responses, maintains session management and transaction records, and stores data in a database for logging and monitoring.
3) Network Environment: Secure WebSocket (WSS) communication over TCP/IP with TLS encryption for data protection, supporting multiple simulated charging stations simultaneously.
4) Development Platform: Implemented using Node.js or Python-based backend services, with a web-based dash-board developed using standard web technologies. The system is tested in a local software environment without physical charging hardware.

V. RESULTS AND DISSUASION:

After successful implementation of the charging station simulator and backend server, the proposed OCPP 2.1 communication framework was tested under controlled simulation conditions to evaluate its performance and reliability. The charging station simulator generated standard OCPP messages such as BootNotification, StatusNotification, and Transaction-Event, which were transmitted securely to the backend server over encrypted WebSocket connections.

The backend system accurately received and processed all incoming messages and displayed real-time communication logs and session status on a web-based monitoring dash-board. The system demonstrated consistent message delivery with minimal latency, enabling continuous interaction between the simulated charging station and the server. Transaction events were handled correctly, and the backend successfully responded with appropriate protocol-compliant messages.

During testing, the secure communication layer maintained stable connectivity without unauthorized access or message loss. The authentication mechanism functioned as expected, allowing only verified charging stations to establish com-munication sessions. The logging and monitoring features provided clear visibility of system activity, facilitating effective debugging and performance evaluation.

Overall, the proposed system exhibited reliable operation, correct protocol handling, and secure data exchange. The results confirm that the software-based implementation of OCPP 2.1 provides a dependable and scalable solution for managing EV charging station communication. Its real-time performance, security integration, and ease of deployment demonstrate its practical applicability for EV charging infras-tructure management in a controlled environment.

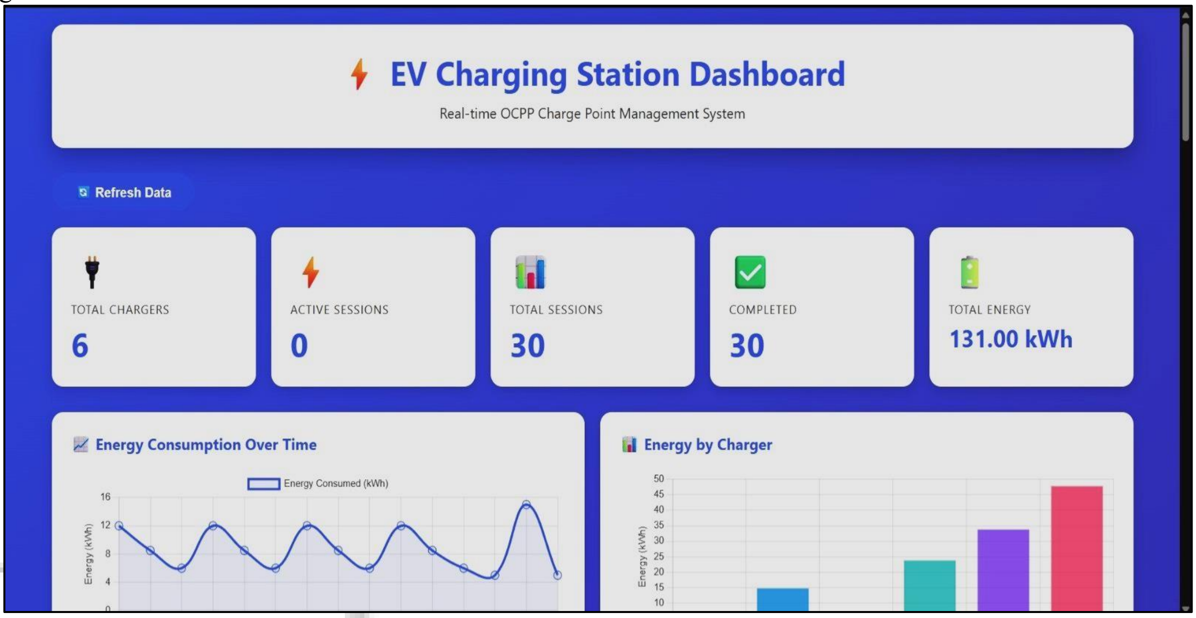

Fig. 2: Dash Board of Application

The pictures (Fig.2) show an Electric Vehicle Charging Station Management System. This system uses a set of rules called OCPP 2.1 protocol to talk to other parts of the system in real time. The diagram shows how the Electric Vehicle Charging Station Management System is connected to the internet. It uses something called WebSockets to talk to the server. The Electric Vehicle Charging Station Management System has a program that acts like a charger and this program talks to the server. The server does jobs, like making sure people are who they say they are and keeping track of what is happening. The server also keeps a record of everything that happens so the Electric Vehicle Charging Station Management System is safe and works properly.

Fig. 3: Energy Consumption

This line graph shows how much energy is used over time in kWh. It shows a pattern of electricity use that happens every day. The use of electricity goes up and down between about 10:30 AM and 1:00 PM. The graph shows that energy consumption keeps going down. It peaks at around 12 kWh. Then goes down to about 6 kWh. This means that sometimes people use a lot of energy and then use less. There is a spike in energy use at 12:26:06 PM. At this point energy consumption reaches 15 kWh. Then it sharply declines. This spike means that people suddenly started using power maybe to charge things. Energy Consumption Over Time has a wave pattern. The wave peaks, at 12 kWh. Energy Consumption Over Time goes down to 6 kWh. The Energy Consumption Over Time spike is 15 kWh.

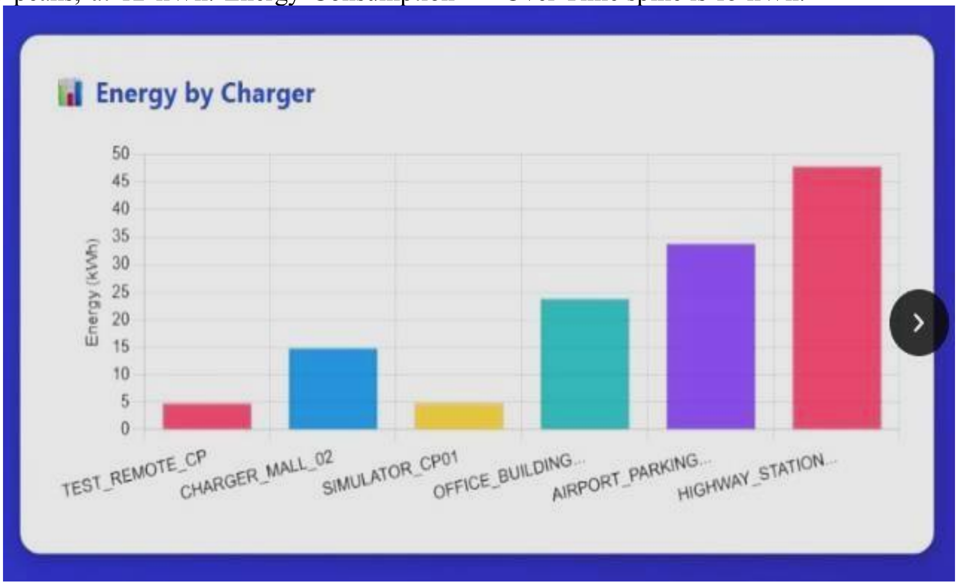

Fig. 4: Energy Consumption by Each Charger

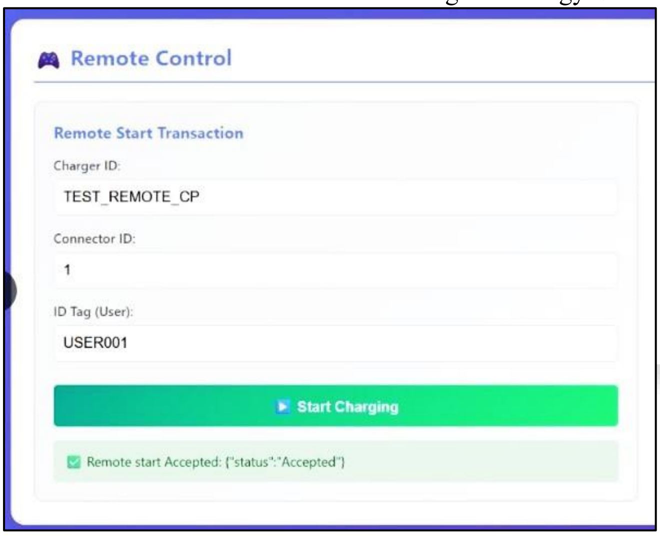

Fig. 5: Remote Control

These images show how well each EV charging station works and how control we have over the whole network. The Energy by Charger chart shows how electricity each place uses. The Highway Station uses the electricity at almost 48 kWh. On the hand places like Test Remote CP and Simulator CP01 do not use much electricity, less than 5 kWh. We can control the charging stations remotely. To start charging operators enter a Charger ID and User ID. Then they can start a Remote Start Transaction. If it works, we see a message that says the command was successful. This means we can control and approve charging sessions from one server. The EV charging network works well. We can manage it easily. The images show that we have control, over the network and can see how electricity each station uses.

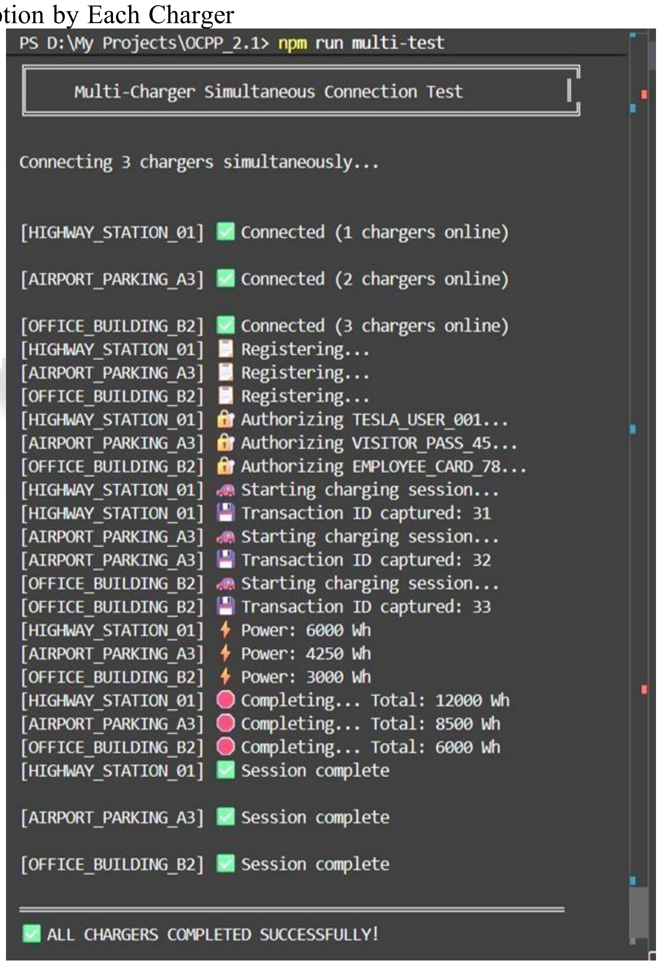

Fig. 6: Session Logs

This picture shows a computer terminal output. It is a test of EV chargers connecting at the same time. The test uses a Node.js script. The test is for three kinds of EV chargers. They are at a Highway Station, Airport Parking and Office Building. They all connect to the OCPP backend. The log shows each step of the process. This includes when they first connect. It also shows when they register. When users authorize with different IDs. For example, "TESLA_USER_001". It also tracks transactions. In the end it shows power delivery numbers. Confirms all sessions were completed successfully. This shows the system can handle connections and process data at the same time. The EV chargers work well with the OCPP backend. The test is a success. It tests the EV chargers and the OCPP backend. The EV chargers are, at places. They all connect to the OCPP backend at the time.

The technical documentation and the project images confirms that the OCPP 2.1 framework is fully operational, with the architectural Secure WebSocket layer effectively facilitating the transmission of BootNotification and StatusNotification messages as described. This seamless data flow is visually validated by the web-based dashboard, where "consistent message delivery" is reflected in the real-time tracking of 30 completed sessions and the live energy consumption analytics. Furthermore, the terminal logs showing the capture of specific Transaction IDs (31, 32, 33) provide the empirical evidence needed to support the text's claim that transaction events are handled correctly, proving that the backend system is not only receiving data but also responding with protocol-compliant accuracy.

VI. DISSUASION:

The ELUDE prototype shows that moving to the OCPP 2.0.1/2.1 standard is very important for keeping EV infrastructure safe. This study used a software simulator to test state transitions from the first registration to real-time events and it worked well. The ELUDE prototype can handle the features of the OCPP 2.0.1/2.1 standard, like better device management in a modular backend architecture.

A big finding of this study is that the Zero-Trust communication model is effective. The ELUDE prototype uses encrypted connections. Checks the identity of devices, which stops common attacks like Man-in-the-Middle attacks and unauthorized access. This keeps information like commands and billing data safe. The study found that the energy consumption data was consistent which shows that the system is working well.

The ELUDE prototype also has a real-time monitoring dashboard, which shows how well the system is working. The dashboard can send commands like resetting the system with little delay. This makes the ELUDE prototype a good choice for cities where the grid needs to be stable. The ELUDE prototype can work with different types of chargers as long as they follow the OCPP 2.0.1/2.1 standard. The study was done in a controlled environment. The results are a good foundation for future work with Smart Grid technologies and the ISO 15118 standard. This will help create an electric vehicle ecosystem that is stronger and can grow. The ELUDE prototype and the OCPP 2.0.1/2.1 standard are important, for the future of vehicles.

VII. CONCLUSION:

The proposed OCPP 2.1 communication system presents a reliable and secure solution for managing interaction between EV charging stations and a centralized backend server. By implementing standardized protocol messaging and encrypted communication channels, the system ensures safe data ex-change and consistent operational control within EV charging infrastructure. The use of a software-based charging station simulator enables flexible testing and validation of protocol compliance without reliance on physical hardware.

The developed system successfully handles essential charg-ing operations such as station registration, transaction management, and status monitoring through a real-time web-based dashboard. Security mechanisms, including certificate-based authentication and secure WebSocket communication, protect against unauthorized access and data manipulation. These features contribute to maintaining system integrity and operational reliability. Experimental evaluation demonstrates that the proposed framework maintains stable connectivity, accurate message processing, and dependable performance under simulated op-erating conditions. Its modular architecture and software-driven design allow easy integration with future charging stations and backend services. Overall, this work highlights the practical applicability of OCPP 2.1 for building interoperable and secure EV charging management systems. The proposed approach supports the development of scalable EV infrastructure and provides a foundation for further enhancements such as smart charging, energy management integration, and large-scale deployment in real-world environments.

REFERENCES:

[1] Eyada, M. M., Saber, W., El Genidy, M. M. and Amer, F. (2020). Performance evaluation of IoT data management using MongoDB versus MySQL databases in different cloud environments. IEEE access, 8, 110656-110668.
[2] Patel, C., Bashir, A. K., AlZubi, A. A. and Jhaveri, R. (2023). EBAKE-SE: A novel ECC-based authenticated key exchange between industrial IoT devices using secure element. Digital Communications and Networks, 9(2), 358-366.
[3] Khy, T. (2025). International Journal of Advanced Research in Science, Communication and Technology (IJARSCT. Published in World Journal of Advanced Research and Reviews.
[4] Eyada, M. M., Saber, W., El Genidy, M. M. and Amer, F. (2020). Performance evaluation of IoT data management using MongoDB versus MySQL databases in different cloud environments. IEEE access, 8, 110656-110668.
[5] Hamdare, S., Brown, D. J., Jha, D. N., Aljaidi, M., Cao, Y., Kumar, S., ... and Kaiwartya, O. (2025). Cyber defense in OCPP for EV charging security risks. International Journal of Information Security, 24(3), 134.
[6] Kirchner, S. R. (2024). Ocpp interoperability: A unified future of charging. World electric vehicle journal, 15(5), 191.