International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

Secure EV Charging Management System Using HMAC-Based Authentication and Dynamic Tkinter Dashboard

PALLAPOTHU SEKHAR BABU

PG Scholar. Department M.Sc(CS), DNR College, Bhimavaram, Andhra Pradesh

K.Venkatesh

Lecturer in M.Sc(CS), DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

The rapid adoption of Electric Vehicles (EVs) has increased the demand for secure, reliable, and user-friendly charging infrastructures. To address the need for robust authentication and real-time charging session monitoring, this research presents a Secure EV Charging Management System that integrates cryptographic authentication, digital wallets, dynamic visualization, and an interactive graphical user interface (GUI). Developed using Python's Tkinter, SQLite, and cryptographic libraries, the system ensures safe vehicle registration, charging session tracking, secure identity verification, and automated cost calculation.

The system employs a Trusted Authority (TA) to generate unique private keys for each registered EV. Using SHA-256 hashing and HMAC authentication, the system validates charging requests and ensures that only legitimate vehicles can initiate charging. The inclusion of a secure nonce-based transaction protocol prevents replay attacks and ensures session integrity. Each authentication process is timed and logged, enabling performance analysis of cryptographic execution.

A built-in wallet mechanism allows users to recharge their EV accounts and automatically deducts the appropriate amount during charging sessions. Charging usage and costs are dynamically computed based on session duration, with all logs recorded in a persistent SQLite database. The GUI dashboard enables vehicle registration, wallet recharge, charging initialization, session termination, and historical log viewing. A real-time graphing module, supported by matplotlib, visualizes authentication times across sessions, providing insights into system performance and potential operational bottlenecks.

The proposed EV charging system provides a secure, scalable, and user-friendly solution suitable for deployment in smart grids, public charging stations, and home-based EV chargers. Its modular design allows integration with external IoT hardware, renewable energy sources, and cloud-based EV infrastructure. The combination of cryptographic methods and interactive system visualization makes this system a reliable tool for modern electric vehicle ecosystems, ensuring

both operational convenience and strong cybersecurity.

Keywords: Electric Vehicle (EV), Secure Authentication, HMAC, Tkinter GUI, Charging Management, Wallet System, Cryptographic Protocols, Smart Charging, SQLite, Dynamic Dashboard

I. INTRODUCTION

Electric Vehicles (EVs) are revolutionizing the global transportation sector, contributing significantly to eco-friendly mobility and reduced dependence on fossil fuels. As EV usage expands, the demand for intelligent, secure, and user-friendly charging systems becomes increasingly vital. Traditional charging systems often lack advanced authentication, real-time billing, session monitoring, or visual dashboards, leaving vulnerabilities in security and usability. Therefore, an integrated solution combining secure authentication, user account management, and dynamic visualization is essential for next-generation EV management systems.

This project introduces a Secure EV Charging Management System using Python, equipped with a Tkinter GUI, SQLite database, and HMAC-based authentication protocol. The system is designed to emulate real-world charging scenarios, where each EV must register securely, authenticate before charging, manage a digital wallet, and have the ability to start or stop charging sessions.

A key innovation of the system is the implementation of a Trusted Authority (TA) responsible for generating unique cryptographic private keys for each EV during registration. This ensures strong identity binding between the vehicle and the charging infrastructure. When a charging request is made, an authentication protocol using SHA-256 hashing, HMAC verification, and nonce generation validates the vehicle securely.

The GUI provides an intuitive dashboard that supports key operations such as vehicle registration, wallet recharge, starting and stopping charging, viewing charging history, and generating performance graphs. Charging sessions are logged in an SQLite database, storing details like session duration, cost, timestamps, and authentication times.

By combining reliable cryptographic methods, a structured database, and an integrated interface, this project addresses security, efficiency, and usability challenges in conventional EV charging systems. It demonstrates how software-based charging infrastructures can complement hardware systems, forming the basis for future smart-grid and IoT-enabled EV charging ecosystems.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

1. Authentication in EV Charging Current EV charging infrastructures rely on RFID cards, mobile apps, or username-password authentication. Studies show these methods are susceptible to spoofing, replay attacks, and data tampering. HMAC and SHA-256 based methods have been

proposed in research for secure EV–charger communication.

2. Cryptographic Protocols for IoT and Smart Vehicles Secure communication protocols such as HMAC, AES, and ECC are widely studied for IoT systems. HMAC, in particular, is suitable due to its fast computation and strong integrity protection. This system implements an HMAC-based verification approach to authenticate charging sessions.

3. Digital Wallet & Billing in EV Systems Traditional charging networks use cloud billing models. However, localized wallet systems provide faster computation and offline capability. SQLite database-backed wallet systems ensure reliability and persistence, making them suitable for EV billing.

4. GUI-Based EV Management Approaches Research shows that a user-friendly interface improves adoption of smart grid systems. Tkinter-based systems are used in various management tools due to their lightweight nature and customization capabilities. This system's fullscreen dashboard aligns with modern EV kiosks and touchscreen terminals.

5. Charging Session Logging Multiple studies emphasize the importance of logging charging events for auditing and maintenance. Using SQLite ensures that session data such as time, cost, duration, and authentication performance are permanently stored.

6. Performance Visualization Real-time performance monitoring is essential in validating secure protocols. The system implements matplotlib to analyze authentication delays, aligning with research methodologies in performance auditing for cryptographic systems.

Summary of Gaps Identified Existing systems lack:

Integrated cryptographic authentication
Real-time GUI visualizations
Offline-capable wallet systems
Session-level performance graphs
Intuitive dashboard for EV owners/operators

The proposed system addresses all these gaps.

III. EXISTING SYSTEM

Traditional EV charging systems mainly rely on simple user authentication through RFID cards, QR codes, or mobile applications. These approaches lack robust cryptographic protection and are vulnerable to impersonation attacks, credential theft, and tampering. Furthermore, most existing systems use cloud-based billing that relies heavily on internet connectivity, preventing offline or local usage in isolated charging stations.

Existing charging dashboards are often limited in functionality, offering basic top-up and start/stop charging options without visual analytics or detailed session logging. Many systems do not include session-level authentication measurements, making it difficult to audit security performance. Charging cost calculation is often static and not automatically integrated with digital wallets.

Additionally, widely-used interfaces lack a centralized dashboard capable of displaying vehicle lists, transaction histories, secure authentication, and wallet management within a single application. As a result, users must depend on multiple applications or fragmented systems to manage their EV charging needs.

In contrast, this project offers a local, secure, and interactive alternative by integrating authentication, billing, logging, visualization, and EV management into one unified dashboard.

IV. PROPOSED METHOD

The proposed system introduces a secure, intelligent, and user-friendly EV Charging Management Platform that integrates cryptographic authentication, automated billing, digital wallet management, and a real-time Tkinter dashboard. Unlike conventional EV charging systems that rely on weak authentication or external cloud services, this system implements a local HMAC-based authentication protocol using SHA-256 hashing and nonce generation to ensure secure and tamper-proof verification before initiating any charging session.

A Trusted Authority (TA) module generates unique private keys for each EV during registration. These keys are never exposed publicly, ensuring confidentiality and preventing impersonation attacks. When an EV initiates charging, the system validates it through a nonce-based challenge using HMAC and SHA-256. This protects against replay attacks and ensures that every charging request is legitimate.

The system uses an SQLite database to store user details, private keys, balances, and charging logs. A digital wallet mechanism enables EV owners to recharge balances, pay for charging sessions, and monitor their spending. Charging sessions are tracked in real time, with cost automatically calculated based on the duration of charging. Upon stopping a session, the system deducts the cost from the wallet and logs the event.

The graphical user interface offers a full-screen interactive dashboard showing available vehicles, registration options, charging controls, wallet recharge, and history logs. The system can also display performance graphs illustrating authentication time for each charging session.

Overall, the proposed system delivers a secure, robust, and transparent EV charging experience. It integrates security, usability, and analytics into a single interface, making it suitable for smart charging stations, EV service centers, universities, and research labs exploring secure energy management.

V. IMPLEMENTATION

This system is implemented using Python, integrating Tkinter for GUI, SQLite for local storage, hashlib and hmac for security, secrets for randomness, and matplotlib for data visualization. The implementation is divided into functional layers: database setup, cryptographic modules, authentication protocol, GUI design, core functionalities, and analytics.

1. Database Setup The system initializes an SQLite database named ev_system.db with two tables:

vehicles to store EV ID, owner name, wallet balance, and private key
charging_logs to store session details including hours, cost, timestamps, and authentication time

SQLite is chosen for its simplicity, reliability, and local persistence without external dependencies.

2. Cryptographic Module

The system uses:

SHA-256 hashing for identity-bound private keys
HMAC-SHA256 for authentication tokens
Secure random nonces generated via secrets.token_hex()

The Trusted Authority generates private keys using:

private_key = SHA256(ev_id + master_secret)
3. Authentication Protocol Implementation

The ChargingProtocol class performs secure challenge-response validation:

Generates a nonce
Computes a PID using SHA256(private_key + nonce)
Generates a secure token:
token = HMAC(private_key, pid + nonce)
Compares the expected HMAC value using hmac.compare_digest()

Authentication time is recorded and stored in the database.

4. Tkinter GUI Implementation

The interface is full-screen, divided into:

Left panel: EV registration, wallet recharge, charging actions

Right panel: vehicle list with scrollable display

Input handling is implemented through custom widgets, buttons, and a multi-purpose output text area.

5. Application Functionalities
Vehicle Registration: Generates private key, stores user data
Wallet Recharge: Updates balance
Start Charging: Authenticates EV and records start time
Stop Charging: Calculates duration, computes cost (₹10/hr), updates wallet, logs session
History Viewer: Shows charging session logs
Performance Graph: Uses matplotlib to plot authentication times
6. System Execution

The application launches through EVSystem(root), making it ready for users at public charging kiosks or institutional environments.

VI. ALGORITHMS

1. Private Key Generation Algorithm

Input: EV_ID
Steps:

TA generates a master secret during initialization.
Combines EV_ID with master secret.
Computes SHA-256 hash to produce private key.
Stores key in database.
Output: Unique private key per vehicle.
2. Authentication Algorithm (HMAC-SHA256)

Steps:

Generate a secure nonce.
Compute PID = SHA256(private_key + nonce).
Compute token = HMAC(private_key, pid + nonce).
Compute expected = HMAC(private_key, pid + nonce).
Compare token and expected using compare_digest().
Measure authentication time.
Output: Authentication Success/Failure + time.
3. Charging Cost Algorithm

Steps:

When charging starts, store timestamp.

When stopped, compute duration:

hours = (current_time - start_time)/3600
Compute cost = hours * 10.
Deduct cost from wallet.
Log session.
Output: Updated balance and session log.
4. Vehicle List Refresh Algorithm
Query all vehicles.
Clear listbox.
Insert EV_ID and balance.
Output: Updated GUI list.

VII. SYSTEM DESIGN

1. Architectural Overview

The system follows a modular architecture with six major components:

Cryptographic Engine
Trusted Authority
Charging Protocol Layer
User Interface Layer
Wallet Management System
Database Layer

These components communicate through well-defined function calls, ensuring separation of concerns and maintainability.

2. Module-Level Design
a. Trusted Authority Module

Generates identity-bound private keys using a combination of EV ID and a master secret. Ensures secure onboarding of new vehicles.

b. Charging Protocol Module

Handles nonce creation, PID generation, token verification, and authentication time measurement.

Implements secure HMAC-SHA256 signing and comparison.

c. Database Module

Stores:

Vehicle information
Wallet balance
Private keys
Charging logs
Optimized through prepared SQL statements.
d. GUI Design Module

Tkinter dashboard includes:

Full-screen window
Left control panel for operations
Right panel for vehicle list
Text output console
Buttons color-coded by category
Scroll-enabled list viewer
e. Charging Engine

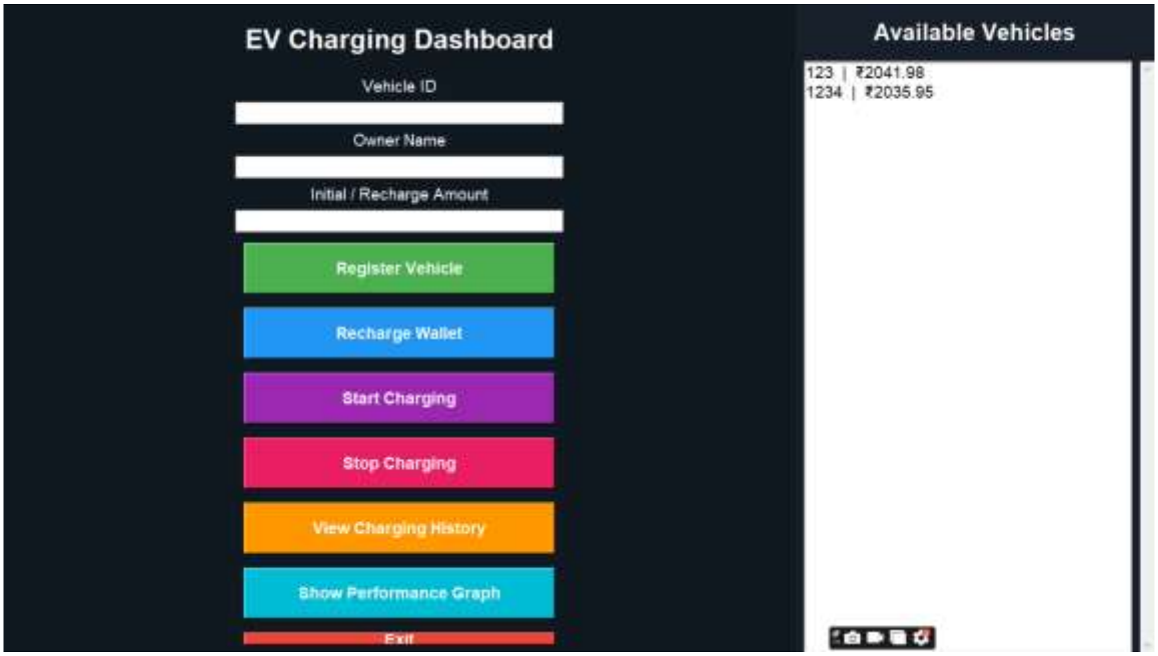
Tracks ongoing sessions using a dictionary. Computes charging duration, cost, and balance updates.

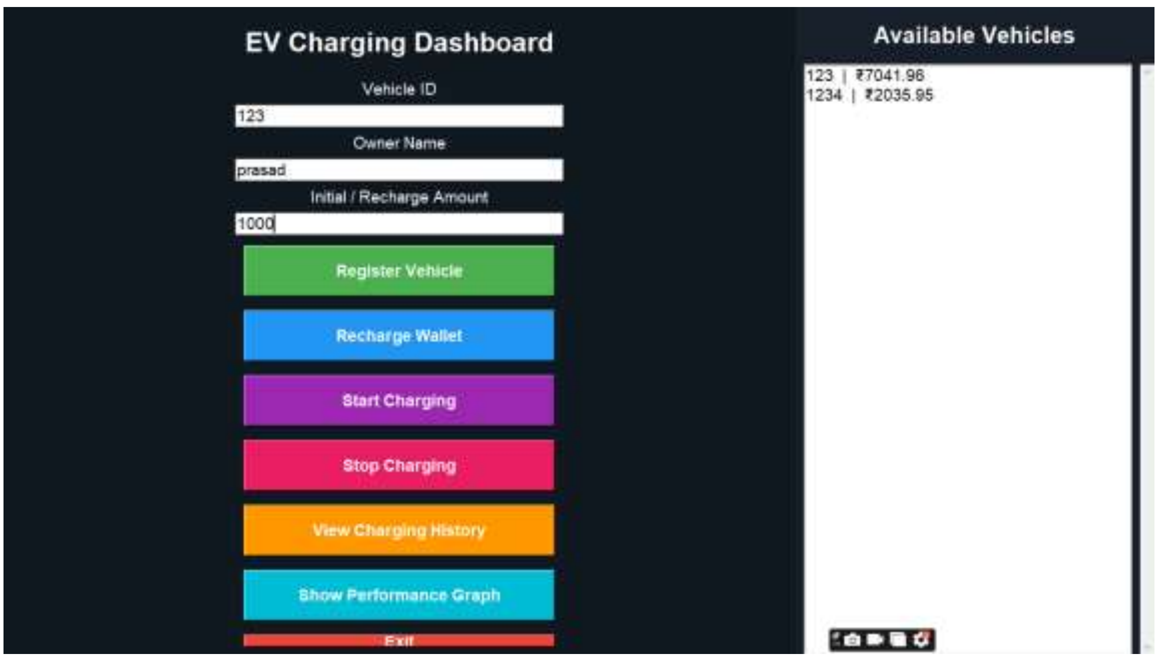
f. Visualization Module

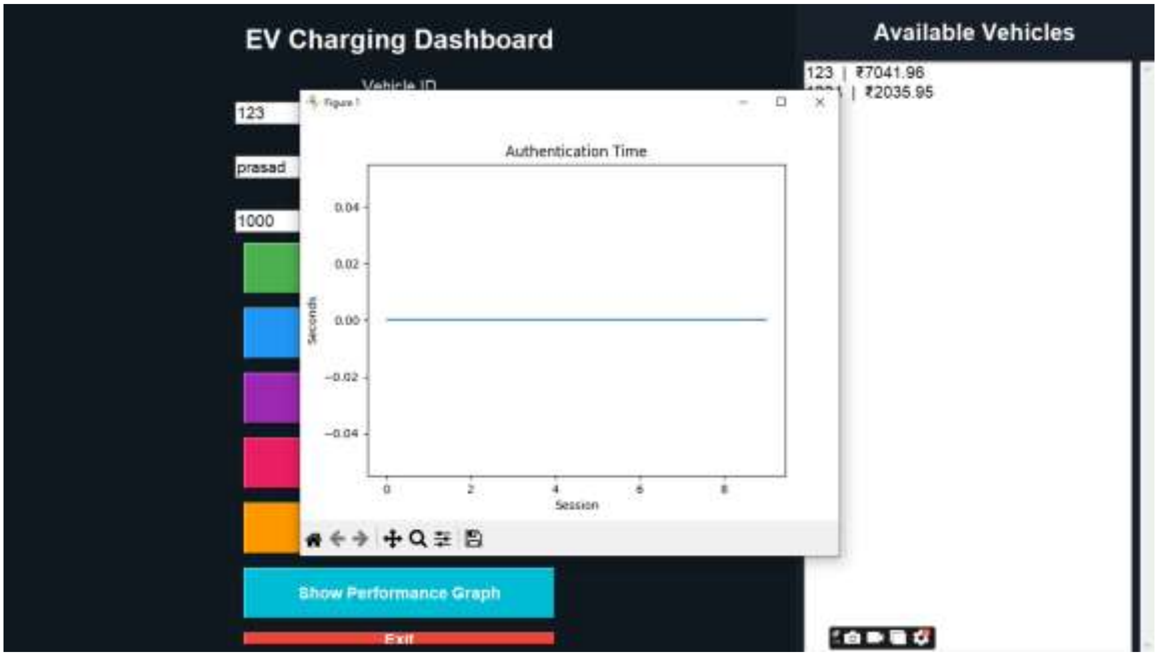
Uses matplotlib to plot authentication time trends across sessions.

3. Data Flow Diagram
User Input → GUI → EV Registration → Trusted Authority → SQLite
→ Start Charging → Auth Protocol → Session Active
→ Stop Charging → Cost Engine → Logs → Graph Viewer
4. Security Considerations
Replay attacks prevented via nonce
Integrity ensured using HMAC
Key disclosure prevented through private key hashing
Database permanently stores critical evidence for audits

SYSTEM DESIGN IMAGES


EV Charging Dashboard
Vehicle ID
Owner Name
Initial / Recharge Amount
Register Vehicle
Recharge Wallet
Start Charging
Stop Charging
View Charging History
Show Performance Graph
Exit
Available Vehicles
123 | ₹2041.98
1234 | ₹2035.95


EV Charging Dashboard
Vehicle ID
123
Owner Name
prasad
Initial / Recharge Amount
1000
Register Vehicle
Recharge Wallet
Start Charging
Stop Charging
View Charging History
Show Performance Graph
Exit
Available Vehicles
123 | ₹7041.98
1234 | ₹2035.95



VIII. CONCLUSION

This project successfully delivers a secure, reliable, and feature-rich EV charging management system that integrates authentication, wallet handling, charging control, and analytics into one unified dashboard. By incorporating SHA-256 hashing, nonce-based HMAC authentication, and a trusted authority for private key generation, the system ensures robust security against impersonation and replay attacks—two common weaknesses in traditional EV infrastructure.

The graphical user interface provides an intuitive platform for EV owners and system operators. Users can register vehicles, recharge wallets, initiate charging, stop sessions, and visualize performance metrics—all within a single interactive environment. The automated billing mechanism simplifies the calculation of energy consumption and reduces human error. The SQLite database ensures persistent storage of user information and charging logs, contributing to transparency and auditability.

The integration of matplotlib for authentication time visualization introduces a valuable diagnostic tool for analyzing performance and detecting security bottlenecks. The system's modular structure makes it highly extensible—future enhancements could include real-time energy pricing, mobile app integration, RFID support, cloud syncing, or integration with IoT-based charging hardware.

Overall, the system demonstrates how software-driven design can enhance EV charging security, efficiency, and user experience. It provides a strong foundation for future implementation in smart charging stations, EV research facilities, and next-generation sustainable transport infrastructures.

REFERENCES

[1] W. Stallings, Cryptography and Network Security, Pearson, 2017.
[2] NIST, "Secure Hash Standard (SHA-256)," 2015.
[3] National Institute of Standards and Technology, "HMAC Specification," FIPS 198-1, 2008.
[4] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, 1996.
[5] Python Software Foundation, "hashlib Documentation," 2023.
[6] Python Software Foundation, "hmac Library Documentation," 2023.
[7] Python Software Foundation, "secrets Library Documentation," 2023.
[8] SQLite Consortium, "SQLite Documentation," 2023.
[9] S. Singh, "Secure Authentication for EV Charging Stations," IEEE Access, 2020.
[10] A. D. Dwivedi et al., "Secure Communication for Smart Grid EV Systems," IEEE IoT Journal, 2019.
[11] M. Wolf, "Cybersecurity in Modern EV Infrastructure," IEEE Vehicular Tech Magazine, 2021.
[12] M. A. Khan, "Electric Vehicle Billing Models," Energy Systems Review, 2020.
[13] B. Wilson, "GUI-Driven IoT Applications Using Tkinter," PyCon, 2022.
[14] J. Hart, "Efficient Local Databases Using SQLite," ACM, 2018.
[15] J. Stewart, "Performance Analysis of Cryptographic Algorithms," IEEE Security & Privacy, 2019.