

ORIGINAL RESEARCH

Open access

Secure and Resilient EV Charging Infrastructure Using Hyperledger Fabric Smart Contracts in Smart Cities

Fatima El Idrissi^{1*}, Samira Bennani¹

Abstract

The growing adoption of electric vehicles (EVs) and the expansion of their charging infrastructure offer substantial potential to advance sustainable mobility in smart cities. Nevertheless, the highly interconnected design of EV charging stations (EVCSS) renders this infrastructure vulnerable to multiple cybersecurity threats, such as false data injection, man-in-the-middle attacks, malware infections, and denial-of-service attacks. Moreover, financial threats—including fraudulent billing and credit card data theft—pose significant risks to EV users. In this study, we introduce a blockchain network built on Hyperledger Fabric tailored for EVCSSs to address these vulnerabilities. The proposed network employs smart contracts to securely and transparently handle essential operations, including authentication, charging session oversight, and payment confirmation. By identifying and preventing unauthorized data alterations or access attempts, the blockchain framework enhances the robustness of EVCS networks. A before-and-after comparative evaluation of the implemented blockchain network illustrates its effectiveness in countering prevalent cyberattacks targeting EVCS infrastructure. The evaluation encompasses performance indicators derived from the Hyperledger Caliper benchmarking tool, revealing the solution's minimal latency suitable for real-time applications and its capacity to scale in line with expanding EV infrastructure demands. Implementing this blockchain-augmented security solution is anticipated to boost user confidence and dependability in EVCS systems.

Keywords Smart city, Electric vehicle, Charging station, Cyberattacks, Blockchain

*Correspondence:

Fatima El Idrissi
fatima.idrissi@gmail.com

¹ Department of Electrical Networks and Smart Energy, Faculty of Engineering, Mohammed V University, Rabat, Morocco

Introduction

Accelerated urban growth, combined with advances in supporting technologies, has driven the worldwide emergence of smart cities. These cities utilize networked systems, sensors, IoT devices, data analytics, and renewable energy solutions to streamline urban infrastructure and resource management, boost operational efficiency, and elevate residents' quality of life. The value of smart cities lies in their capacity to address contemporary urban issues, including traffic gridlock, energy demand, water supply, and ecological preservation, thereby fostering sustainable lifestyles. By 2050, 68% of the global population is forecasted to reside in urban regions [1], rendering the delivery of prompt and effective services a major hurdle amid already overburdened resources. Incorporating advanced technologies into routine urban functions enables smart cities to deliver more efficient and economical services, minimize resource depletion, strengthen environmental safeguards, and enhance public security and welfare.

Transportation represents a vital component of smart cities, where electric vehicles (EVs) play a pivotal role in curbing greenhouse gas emissions and facilitating the shift toward net-zero targets. Research by Oliveri *et al.* [2] indicates that

replacing conventional internal combustion engine vehicles with EVs can yield a substantial 50% decrease in greenhouse gas emissions. Authorities and governments worldwide are actively promoting wider EV adoption to align with carbon-reduction commitments under the Paris Agreement. The synergy of EVs, renewable energy integration, and intelligent grids supports the development of cleaner transport ecosystems in smart cities, which, in turn, contributes to better public health and living standards. As a result, the worldwide EV market is expanding swiftly, with projections estimating 145 million electric cars in operation by 2030 [3]. According to the International Energy Agency (IEA), achieving the Net Zero by 2050 Roadmap requires EVs to constitute approximately 60% of total vehicle sales [4].

Electric vehicles depend heavily on residential and public charging solutions to ensure convenient, widespread access to charging facilities. Home charging, although typically slower (using Level 1 and Level 2 chargers), provides EV owners with the practicality of overnight recharging. Public charging stations, by comparison, deliver faster (Level 3) charging and support charging during travel, which is crucial for extended journeys and daily urban commutes. Additionally, numerous workplaces now provide charging facilities in employee parking areas, enabling charging during work hours. Transferring charging activity from residential to public stations can enhance grid performance and reliability [5].

The EV charging station ecosystem is an integrated, interconnected framework encompassing hardware components, software systems, communication standards, and electricity supply mechanisms. Hardware aspects include the physical charging equipment featuring power sockets, cables, and connectors, engineered to accommodate diverse power levels based on charging type. Electric vehicle supply equipment (EVSE) combines hardware and software to enable smooth, secure functionality while addressing the diverse needs of EV users and station operators [6]. Sophisticated communication and software capabilities, along with user-friendly interfaces, have been engineered to improve the overall experience and accessibility of EV charging systems. Users have access to various authentication and payment methods, including external identification means (EIMs) via QR codes, RFID cards, and NFC, as well as internal identification methods (IIMs) in which the vehicle performs self-authentication via approaches such as Autocharge [7] and Plug and Charge [8]. Today, most major EV producers endorse either standardized or manufacturer-specific Plug and Charge systems. IIMs involve data interchange between the EV and EVSE for vehicle authentication on the driver's behalf. This interchange adheres to the ISO 15118 standard [9], a Vehicle-to-Grid (V2G) protocol, implemented via Power Line Communication (PLC) using the HomePlug Green PHY modem.

Furthermore, EVSE must initiate and sustain communication with the charging station management system (CSMS). The Open Charge Point Protocol (OCPP) [10] facilitates this interaction, supporting remote authentication, authorization, monitoring, and firmware updates. Power management mechanisms optimize energy allocation to prevent grid strain during peak demand periods. The charging infrastructure links to a cloud platform that oversees charging sessions, manages user information, processes electricity consumption billing, and conducts real-time data analysis.

The interconnection among charging stations, service providers, EVSE, EVMS, and cloud platforms—along with interactions between EV users and these components—forms a highly complex system that involves extensive data exchanges over the Internet. This architecture renders the complete EV charging infrastructure susceptible to a wide range of cyber threats. A primary security concern is false data injection attacks, in which adversaries manipulate communications between EVSEs, smart meters, and CSMSs to falsify charging parameters and battery state information. Such manipulations can cause overcharging or contribute to grid instability. Man-in-the-middle (MITM) attacks enable interception and alteration of communications, thereby affecting billing accuracy and charging processes. Denial-of-service (DoS) attacks overwhelm systems with excessive traffic, rendering charging stations inoperable and potentially jeopardizing the functionality of emergency vehicles, thereby threatening public safety.

The EV charging infrastructure is further vulnerable to malware injection attacks. Once malware infiltrates the system, it can propagate quickly across the network, endangering not only isolated charging stations but also entire charging networks and the connected power grid. These incidents may lead to data breaches, including the theft of confidential user information such as credit card details. Ransomware attacks can similarly target EVSE and/or CSMS, locking

system operators out until a ransom is paid. This results in prolonged service disruptions and substantial operational downtime until recovery is achieved.

In recent years, blockchain technology has established itself as a powerful mechanism for safeguarding digital transactions and data exchanges. Leveraging secure authentication, distributed consensus, and other inherent features, it presents a compelling approach to counter cyber threats across multiple domains, including healthcare, supply chain management, finance, and energy. By integrating cryptographic methods with a decentralized ledger, blockchain creates traceable, tamper-proof transaction records, thereby substantially mitigating risks of data integrity violations and unauthorized access.

In the domain of EV charging infrastructure, these blockchain attributes position it as a reliable technology for defending against threats such as data injection, MITM, and ransomware. It achieves this by guaranteeing the integrity and authenticity of communications among EVs, EVSE, CSMS, and other associated infrastructure elements. Blockchain-based smart contracts enable the establishment of a secure charging environment by verifying user credentials, payment details, and vehicle identity, while enforcing accurate billing and energy metering. These contracts can also automate and enforce transactional rules. Consensus protocols enable immediate detection and rejection of any unauthorized alterations to charging data or transactions. Moreover, blockchain can counter DoS attacks by decentralizing verification across multiple network nodes, thereby eliminating single points of failure that attackers could exploit. The inherent transparency of blockchain fosters greater confidence among both users and service providers in the reliability of EV charging facilities.

Blockchain has already proven its value in various EV charging-related applications. For example, it has facilitated improved coordination and trust among energy providers [11], supported autonomous selection of optimal charging stations [12], enabled privacy-preserving dynamic tariff mechanisms [13], and ensured transparent billing processes [14]. It has also been applied to deter collusion among charging stations [15], enable peer-to-peer energy trading and load balancing [16], and lower the overall costs associated with EV adoption [17]. Despite these advancements, the effectiveness of blockchain specifically for protecting EV charging infrastructure against cyberattacks remains underexplored.

In this paper, we apply blockchain to secure the operation of EV charging networks and infrastructure, making the following key contributions:

- We present a blockchain-based smart contract designed to mitigate the primary categories of cyberattacks affecting EV charging ecosystems. Although blockchain has been employed in other EV and charging contexts, no prior work has proposed a smart contract-based blockchain solution specifically aimed at preventing such attacks.
- We define stakeholder access privileges within the charging network along with corresponding smart contract rules, which are embedded directly into the blockchain. These privileges and rules support forensic investigation capabilities in the case of security incidents.
- Through comprehensive simulations and performance assessments, we demonstrate that the proposed smart contract-driven blockchain effectively counters cyberattacks on EVCS infrastructure. Benchmarking with Hyperledger Caliper reveals strong scalability in resource utilization, read throughput, and latency even as transaction volumes rise, confirming its viability for real-time deployment.

Literature Review

An electric vehicle charging station (EVCS) encompasses multiple components that can serve as potential entry points for cyberattacks, including: (i) the EV connector, which physically links the vehicle to the charging station to enable both power delivery and data communication between the EV and EVSE; (ii) the user terminal, consisting of a display screen,

card reader, or touchscreen interface that facilitates driver interaction with the system for authentication, charging control, and payment handling; (iii) the Internet connection (wired or wireless), which supports authentication, payment processing, and remote management by providers via the CSMS; and (iv) the maintenance terminal, a dedicated interface used for routine diagnostics, fault resolution, firmware upgrades, and general system upkeep. Consequently, an EV charging station integrates EVSE units, user terminals, communication modules, power supply components, control units, CSMS interfaces, and related elements.

Considering the susceptibilities inherent in all elements of EV charging station infrastructure, the primary cyberattacks highlighted in the existing literature include the following:

False data injection (FDI): This type of attack compromises communication channels and alters data exchanged among components such as the EV, EVSE, and CSMS by exploiting weaknesses in communication protocols [18, 19]. By modifying essential parameters such as charging rates or battery conditions, perpetrators can induce battery overcharging, degrade vehicle efficiency, and potentially destabilize the grid [20]. Such FDI manipulations may disrupt charging activities and lead to erroneous decision-making, triggering widespread repercussions across the charging ecosystem.

Man-in-the-middle (MITM): These attacks commonly target data interactions between the EV and the EVSE, or between the EVSE and the CSMS. An EVSE-to-CSMS breach can expose sensitive details related to pricing, firmware updates, and access control policies. In contrast, an EV-to-EVSE breach may reveal the vehicle's location, user authentication details, and payment data. Consequences include manipulated charging commands, excessive or insufficient battery charging, inaccurate invoicing, diminished vehicle capabilities, violations of user privacy, and overall degradation in the charging network's reliability.

Denial-of-Service (DoS): Such attacks saturate the network with fraudulent charging requests, thereby blocking legitimate EVs from using charging services [21]. Certain DoS variants, including grayhole attacks (which selectively forward messages) and blackhole attacks (which discard all messages), can severely impair protocols such as OCPP, leading to operational disruptions and communication failures in the EV charging setup. Another variant entails deliberately provoking authentication errors, compelling other EVs to remain idle until the sequence of failures ends [22].

Malware injection (MI): Exploitable flaws in the charging network allow attackers to inject malware into the EVSE, enabling it to spread rapidly throughout the system, jeopardizing the entire EV charging ecosystem and potentially the power grid itself. MI attacks are often initiated by individuals with physical system access or privileged credentials [23]. Outcomes may encompass data exfiltration and unauthorized energy consumption. Korba et al. [24] outline a malware scenario in which the attacker inflates electricity prices during designated periods to deter other users from charging, thereby securing lower-cost access for themselves. A particularly damaging MI variant is ransomware, which encrypts the charging system and extorts payment for decryption and restoration.

Table 1 summarizes the various categories of cyberattacks and their possible consequences for EV charging stations and infrastructure in a smart city setting. **Figure 1** illustrates how cyberattacks can propagate, infecting EVs and their associated charging networks.

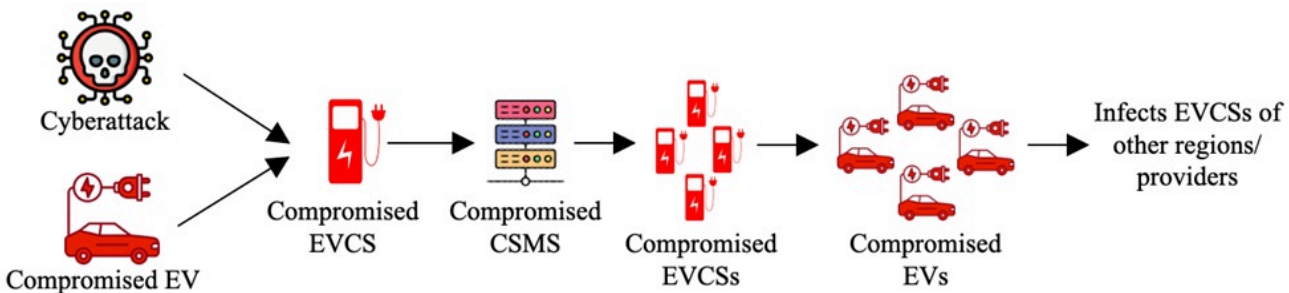


Figure 1. Cybersecurity threat targeting EVCS.

Table 1. Major types of cyber attacks and their impacts on EV charging stations, infrastructure, and vehicles in a smart city environment. (Note that changes to the position of figures and tables may occur during the final steps, since all figures/tables should be placed below the first citation, or a few paragraphs lower, to reduce blanks on the page.)

Attack type	Components	Description	Potential impact
False Data Injection (FDI)	EV, EVSE, and CSMS	Inserts or manipulates fabricated data for the EV, the overall system, and power grid operations.	Compromises the integrity of charging information, potentially resulting in battery overcharging and reduced vehicle performance. It may also destabilize the power grid.
Man-in-the-Middle (MITM)	EV, EVCS, and CSMS	Intercepts and alters communication exchanged between EV and EVSE, and between EVSE and CSMS.	Modifies charging commands, leading to overcharging or undercharging, reduced driving range, hardware damage, and exposure of sensitive data shared between EV and CSMS. May enable fraudulent payment activities.
Malware Intrusion (MI)	EVSE and CSMS	Deploys malicious software into EVSEs and CSMS by exploiting system vulnerabilities. After infection, the malware can propagate across other EVSEs within the network.	Threatens the security of the full charging infrastructure. May result in data breaches involving personal information, credit card details, billing records, and vehicle authentication credentials.
Ransomware	EVCS, CSMS, charging network, Power grid	Encrypts essential system data and demands ransom for restoration.	Causes service outages, financial losses, disruptions to grid functionality, and possible power blackouts.
Denial of Service (DoS) and Distributed DoS (DDoS)	EVSEs, CSMS	Overloads EVCS and CSMS with excessive traffic or repeated requests, making the systems inaccessible.	Blocks EV access to charging services, which is especially critical for emergency vehicles dependent on charging availability.

Blockchain applied in EV charging networks

Blockchain technology has found applications across multiple facets of EVs and their associated charging infrastructure. This subsection surveys several prominent studies on the use of blockchain in EV charging networks for diverse services.

Fu et al. [11] developed an EV charging framework based on consortium blockchain to enhance coordination and openness among energy providers, with validation through a Beijing-based case study. The blockchain facilitated secure, decentralized oversight of charging information, promoting collaboration and confidence, while smart contracts streamlined resource distribution and billing workflows. In a separate effort, Pustišek et al. [12] described a blockchain-enabled system that automatically identifies the most suitable EV charging station based on criteria such as cost, proximity, and queue duration. Smart contracts automate machine-to-machine transactions, enabling drivers to obtain offers from stations and select the one they prefer.

Knirsch et al. [13] proposed a method that enables EV users to obtain quotes from multiple charging stations to find optimal pricing, while safeguarding the confidentiality of their whereabouts and energy requirements. Blockchain

guaranteed that the bidding interactions between stations and users remained transparent and verifiable. Jeong et al. [14] introduced a blockchain-supported billing mechanism for EVs and charging facilities. It decentralized billing operations, integrated mutual authentication procedures, and maintained the transparency and permanence of billing records.

Regarding energy management in EV charging, Lin et al. [15] presented a blockchain-integrated intelligent CSMS that combined AI and IoT to monitor power transactions and optimize charging/discharging operations. The system supported clear energy trading, cost reduction, and greater effectiveness in managing extensive EV charging deployments.

Huang et al. [16] proposed a blockchain-driven energy trading approach for EVs applicable in both day-ahead and real-time markets. EV users placed bids via a double auction framework, with a charging system operator (CSO) handling trade agreements on the blockchain. Empirical evaluations indicated enhanced transaction performance, greater gains for purchasers and vendors, and improved load equilibrium. Ping et al. [25] employed an Ethereum-based blockchain coordination platform for EV charging to counteract collusion and unethical conduct among charging stations and coordinators. Analytical and simulation results confirmed that the platform elevated social welfare by curbing price rigging. Dorokhova et al. [17] proposed a blockchain-based architecture for EV charging oversight, leveraging Ethereum and smart contracts to automate energy transfers between EVs and charging stations. Implemented in Switzerland, this approach lowered expenses for private and semi-private charging installations.

Blockchain applications for cybersecurity in diverse fields

Some research efforts have used blockchain technology across sectors such as healthcare, supply chain management, financial services, energy systems, and smart urban environments to address serious concerns about security and data privacy. Kirimtat et al. [26] conducted a detailed analysis of smart city ecosystems, focusing on data-handling techniques applicable to smart governance, smart economy, smart environment, and smart living domains; however, secure urban protection strategies were outside their scope. Ahmad et al. [27] explored vulnerabilities concerning data integrity, algorithmic robustness, and overall security in AI-oriented smart city solutions designed around human needs, while also considering blockchain's contribution toward transparent data auditing. Singh et al. [28] studied how combining blockchain with artificial intelligence can advance the development of smart transportation infrastructure in modern urban settings. Majeed et al. [29] provided an overview of blockchain use in IoT-supported smart cities, highlighting key developments and deployed implementations. Yet, they largely bypassed an in-depth discussion of security vulnerabilities in more complex scenarios. El Bekkali et al. [30] demonstrated blockchain's promise for strengthening data protection in smart city contexts and proposed an integrated architecture that combines blockchain, big data analytics, and AI; this model was experimentally evaluated using real smart grid data. The evaluation results affirmed both the framework's performance and its realistic potential to resolve key cybersecurity issues in smart city deployments.

As previously outlined, blockchain has emerged as a proposed mechanism for coordinating electric vehicle charging infrastructure and is highly viable for supporting safe, streamlined operational processes. Crucially—and to the best of the authors' knowledge—no published approach has yet applied blockchain-powered smart contracts to neutralize the cybersecurity risks identified earlier in this work directly. While numerous investigations have examined broader security and privacy issues in smart city contexts, the existing literature still lacks focused, purpose-built cybersecurity defenses tailored to electric vehicle charging station (EVCS) networks. This deficiency leaves such networks vulnerable to diverse cyber threats and the risk of service interruptions. Blockchain technology supplies a powerful countermeasure through features such as distributed identity verification, reliable transaction confirmation, and permanently unalterable logs, all of which significantly bolster the cyber resilience of EVCS infrastructures. In response to this research gap, the blockchain-based EVCS architecture we present here has been intentionally crafted to shield EVCS networks from the specified threats, thereby creating a secure, extensible base that complements smart city development priorities and accommodates the accelerating adoption of electric vehicle technologies.

Participants in Blockchain-Enabled EVCS Networks

Pinpointing the various actors within a blockchain-supported EVCS ecosystem is fundamental to clearly assigning responsibilities, defining privileges, and specifying interaction rights for all parties involved. Key participants—electric vehicles themselves, charging facilities, electricity suppliers, payment and billing services, and maintenance teams—connect to the platform to conduct transactions and perform daily functions in a secure environment. Cataloging these actors enables precise governance rules for who may access data, how authentication occurs, and which transactions require approval. Consequently, this structure enables effective smart contract execution and fine-grained access controls, ensuring that only legitimate entities can initiate or influence selected operations across the network. An overview of the main stakeholders, together with their primary responsibilities and capabilities, follows.

Electric vehicles

Role

Within the network, electric vehicles act as final consumers; each is assigned a distinct, verifiable identity registered on the blockchain. Every engagement between an EV and any charging point is permanently captured on the chain.

Functionality

EVs communicate with the charging infrastructure to request, monitor, and complete battery charging. They additionally interface with payment and billing services to settle transaction costs.

Service providers

Individual EVs may maintain relationships with several supporting entities—municipal authorities, retail complexes, vehicle producers, or external specialized providers—all of which help route the vehicle to authorized and trustworthy charging locations.

Electric vehicle charging station

Role

Charging stations incorporate both tangible components (hardware and firmware, including EV supply equipment—EVSE) and digital components (software handling billing, payments, and related services) that enable EVs to replenish their energy storage.

Functionality

Every charging station possesses a unique blockchain identity; all charging events and related exchanges are immutably documented on the ledger.

Charging station management system

Role

While an EV is charging, the EVSE maintains a remote link to the central Charging Station Management System (CSMS) through the standardized OCPP protocol.

Functionality

The CSMS is responsible for verifying user credentials, recognizing vehicles, authorizing financial transactions, and enabling remote oversight of station status. Each charging event, associated payment, and energy transfer is faithfully recorded onto the blockchain.

Maintenance groups

Role

These entities are tasked with routine servicing, urgent fault correction, and overall operational maintenance of charging station equipment.

Functionality

Maintenance teams communicate with charging stations to schedule and document repair or servicing activities. The blockchain securely archives all maintenance history entries together with the authenticated identities of the responsible individuals and organizations.

Billing platforms/user interfaces

Role

Billing platforms are responsible for managing payment transactions for EV charging and supporting payment methods such as Google Pay, Apple Pay, Alipay, and other external payment solutions.

Functionality

Once payments are authenticated, full transaction details—including the transaction identifier, monetary value, timestamp, quantity of energy delivered, identifiers of the involved EV and EVSE, and additional relevant information—are immutably recorded on the blockchain.

Energy providers

Role

Energy providers deliver electrical power to the charging stations and may operate at local, regional, or national levels.

Functionality

Electricity consumption is validated, and the corresponding data is securely logged on the blockchain. Precise tracking of energy usage supports accurate billing, regulatory compliance, and protection against unauthorized energy consumption or theft.

Service providers (municipal authorities, retail complexes, vehicle manufacturers, independent operators)

Role

Service providers function as facilitators linking electric vehicles to available charging infrastructure. This category encompasses city or regional councils, shopping malls, automotive companies, and external specialized operators.

Functionality

These providers ensure EVs can locate and use approved charging stations. By leveraging blockchain, transparent and reliable partnerships are established between service providers and charging facilities, reducing the risk of unauthorized charging attempts and enabling secure management of diverse user accounts.

Table 2 summarizes the responsibilities and permission levels assigned to different participants in the blockchain-supported EV charging ecosystem. It details the rights to create, update, and view critical data categories to ensure controlled and legitimate system interactions.

Table 2. Responsibilities of key participants in the blockchain-supported EV charging network. Permissions cover creating entities (CE), modifying entities (ME), accessing charging data (ACD), accessing financial data (AFD),

authentication/authorization (ATH), and maintenance tasks (MN).

Stakeholders †	CE	ME	ACD	AFD	ATH	MN
CP	Yes	Yes	Yes	Yes	Yes	Yes
AD	Yes	Yes	Yes	Yes	Full Access	Full Access
EV	No	No	Yes	Yes	Authenticate	No
EVCS	Yes	Yes	Yes	No	Monitor	Update
EVSE	No	No	Yes	No	Monitor	Update
CSMS	Yes	Yes	Yes	Yes	Authenticate	Monitor
MGs	No	Yes	No	No	No	Update
BP	No	No	No	Yes	Verify	Process Payments
EP	No	Yes	Yes	No	Supply	Monitor
SPs	No	Yes	Yes	No	Access Control	Register Stations

† CP: charging providers, AD: admin, MG: maintenance group, BP: billing platform, EP: energy providers, SP: service providers.

Data Flow and Permission Management

This section details the flow of information and the permission-enforcement strategies implemented in the proposed blockchain-powered EV charging systems. The architecture, built on Hyperledger Fabric, provides secure, distributed processing for all transaction types—including vehicle enrollment, charging activities, financial settlements, and maintenance operations. As described in the prior section, each network participant receives tailored access privileges, which are regulated by digital identity certificates issued by the Fabric Certificate Authority (CA). The Membership Service Provider (MSP) reinforces these controls, ensuring that only verified participants can submit or execute transactions. The following subsections explain the primary elements and their operations within this blockchain environment as applied to EV charging infrastructure.

Core elements of the EV charging blockchain system

The Hyperledger platform forms the backbone of the proposed EV charging blockchain solution. Implemented using Hyperledger Fabric, it guarantees tamper-resistant logging of transactions, such as charging events, identity checks, and ledger updates, across the distributed system. A vital element is the Fabric Certificate Authority (CA), which issues digital certificates to verify the legitimacy of all participants—including charging stations, electric vehicles, service operators, and maintenance staff—thereby restricting network access to only trusted, authenticated parties.

Figure 2 illustrates the main building blocks of the blockchain-driven EV charging architecture, highlighting components such as the Fabric Certificate Authority, the overarching Hyperledger Network, the Membership Service Provider, peer nodes, and administrative accounts. Interactions involving public/private key cryptography, chaincode processing, and ledger synchronization create a robust, auditable structure for overseeing EV charging operations.

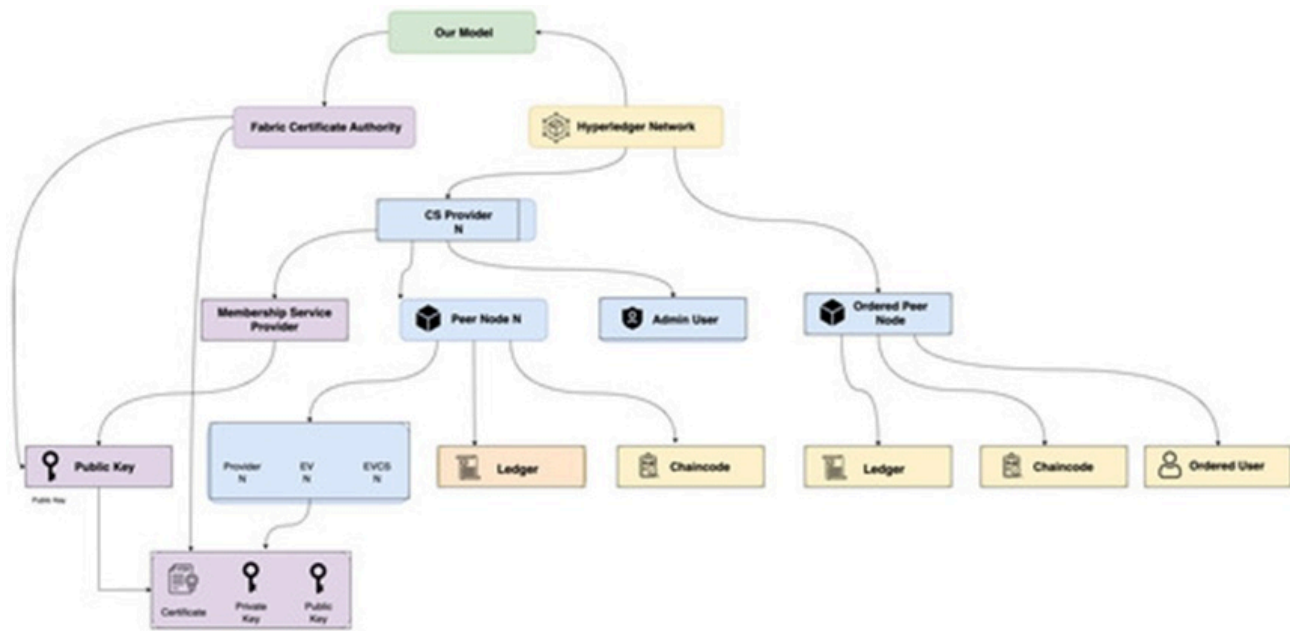


Figure 2. High-level view of principal elements in the blockchain-supported EV charging network.

The Membership Service Provider (MSP) oversees identity governance by relying on certificates supplied by the Fabric CA to validate participants. Before any transaction is executed, thorough verification is required for both charging operators and EV owners. Charging providers are modeled as distinct organizations (labeled Org1 through OrgN) that collectively constitute a network comprising N organizations. These organizations oversee clusters of charging stations while upholding secure identity credentials and access privileges to enable smooth coordination with vehicles. Additionally, the MSP administers a certificate revocation list (CRL) to immediately block any compromised or untrustworthy entities from participating in future activities.

Hyperledger Fabric CA public keys are used to encrypt data and verify digital signatures, thereby protecting communications across the blockchain environment. Every participant—ranging from charging stations to electric vehicles—holds a public key that authenticates their actions. In addition, each participant is issued a certificate embedding their identity along with their public key. This mechanism ensures that only authorized parties can interact with the blockchain, thereby preserving the overall security of network exchanges.

Each provider manages its own Peer Node N, which maintains a complete replica of the blockchain ledger. These peer nodes participate in transaction endorsement and validation to ensure ledger entries remain accurate and protected. Complementing this structure, the ordering peer node sequences transactions into blocks, delivering uniform ledger updates to all peer nodes and upholding the consistency and integrity of the distributed ledger.

The ledger functions as a decentralized, unalterable archive that captures every transaction—from vehicle registrations and charging activities to payment records—providing a clear, trustworthy historical record. Chaincode, which forms the smart contract component, establishes the operational rules governing all network transactions. It regulates procedures such as EV authentication, charging session initiation, payment execution, and compliance with security standards.

An Admin User is responsible for supervising the blockchain network's administration, including onboarding or offboarding participants, adjusting charging station configurations, and maintaining seamless network performance. Authorized Ordered Users, in turn, can submit transactions that impact the ledger—such as registering EVs, approving sessions, or managing services—while adhering to the protocols enforced by the ordering service.

When an EV or EVCS submits a transaction T accompanied by a signature (T) to invoke Chaincode, the MSP executes the following verification sequence to confirm authenticity and legitimacy:

- Extract Identity: Retrieve the digital certificate *Cert* embedded within transaction T .
- CRL Check: Cross-reference certificate *Cert* against the Certificate Revocation List (CRL) to confirm it remains valid and unrevoked.
- Signature Verification: Validate the signature (T) by applying the participant's public key *Cert.PubKey* extracted from the certificate.
- Access Control: Consult the Access Control List (ACL) to verify that the identity specified in *Cert.IDinfo* possesses the required privileges for executing this transaction.

Should all verification steps pass successfully, the transaction advances to the ordering and commitment stages within the blockchain workflow.

Suggested Smart Contract Architecture Based on Blockchain for the EVCS Network

The blockchain-powered smart contract solution we propose for the electric vehicle charging station (EVCS) network—implemented using Hyperledger Fabric—establishes a protected, distributed mechanism for conducting transactions and preserving data consistency among all parties involved, namely electric vehicles, charging facilities, electricity suppliers, payment processing systems, and service/maintenance teams. In this setup, each charging station operator functions as a distinct organization (Org1, Org2, and so on) on the blockchain, with each organization overseeing multiple charging points. Once registered, electric vehicles are enrolled as full participants in the blockchain ecosystem and gain access to powerful security protections that deliver unalterable, fully visible, and permanently recorded interactions among all stakeholders. Smart contracts automate authentication and permission management, greatly reducing reliance on legacy techniques and enabling real-time detection of attempts to manipulate, commit illicit entry, or engage in malicious cyber operations. The primary innovation of our design lies in its integration of decentralized role-based access control (RBAC), digital signature cryptography, and agreement-driven transaction confirmation, which collectively prevent illicit changes to information and maintain system trustworthiness even under hostile conditions. Beyond defending against cyber risks, this strategy meets the essential requirements of expandability and openness for smart-city electric vehicle charging infrastructure. Detailed technical descriptions appear in the subsections that follow: “EV Charging Hyperledger Network”, addressing transaction flow and chaincode operation, and “Attack Mitigation in EVCS Network in the Proposed Approach”, explaining the protective features designed to counter cyber vulnerabilities while guaranteeing continuous network robustness and trustworthy data handling.

EV charging hyperledger network

Built on Hyperledger Fabric (HF), the blockchain platform for electric vehicle charging facilitates secure, auditable exchanges among EVs, charging stations (EVCS), central management systems (CSMS), service providers, end-users, and related actors, while consistently maintaining data accuracy for all participating entities. The section below outlines the transaction-handling workflow implemented in Hyperledger Fabric.

Transaction processing

Any EV charging-related transaction submitted to the Hyperledger environment progresses through a clearly defined sequence of stages, as detailed below.

The process starts with proposal submission: either the electric vehicle or the charging station application creates and sends a transaction proposal to the designated endorsing peers. Each endorsing peer then simulates the proposed action by invoking the relevant Chaincode, which produces a read-write set (RW_Set). This RW_Set documents precisely which ledger keys will be examined and which values will be altered, thereby defining the transaction's intended outcome before any permanent changes occur.

Next comes the endorsement phase. Endorsing peers cryptographically sign their simulation results—comprising both the computed RW_Set and their individual endorsement signature (symbolized as E_Sig). The signature itself is calculated via the expression:

$$E_Sig = Sign(Hash(RW_Set), Peer_SK), \quad (1)$$

In which Peer_SK stands for the endorsing peer's private cryptographic key and Hash() denotes a collision-resistant hash algorithm (most commonly SHA-256). This step confirms that qualified peers have examined and consented to the transaction before it advances.

Once sufficient endorsements are gathered, the submitting client bundles them and transmits the complete package to the ordering service. The ordering service sequences all pending transactions according to a globally consistent timeline, groups them into blocks, and broadcasts the newly formed blocks across the peer network for final verification and ledger update. Maintaining this uniform, chronological order network-wide prevents race conditions and duplicate spending.

The last phase entails validation and commitment. Every peer independently verifies each transaction by inspecting the attached endorsement signatures and confirming that the read-write sets remain compatible with the present ledger state. Successful checks result in the peer incorporating the write operations into its local world state database according to the rule:

$$New_State = Apply(Current_State, Write_Set), \quad (2)$$

Where Current_State is the ledger's existing world state, and Write_Set holds the updated key-value mappings. After this application, the peer records the validated block on its blockchain copy, thereby extending the permanent, unchangeable chain of transaction records. Through this coordinated procedure, all peers maintain identical ledger contents, ensuring both the reliability of the blockchain and the openness of every recorded operation.

Figure 3 illustrates the message flow and component interactions in the Hyperledger Fabric-based EVCS network. The sequence begins when an electric vehicle generates its cryptographic key pair and submits an account registration request to the network administrator. The administrator verifies whether the identity already exists and, if not, initiates account creation. After successful validation, the Certificate Authority generates and issues an enrollment certificate (ECert) to the vehicle. The vehicle then prepares and dispatches a transaction proposal, which multiple endorsing peers review and sign. The collected endorsements are passed to the ordering node (Orderer) for sequencing and block formation. Finally, committing peers validate the block and append it to their local ledgers, permanently recording the transaction. This workflow executes identically for every subsequent operation, delivering consistently secure and manipulation-resistant behavior throughout the EVCS ecosystem.

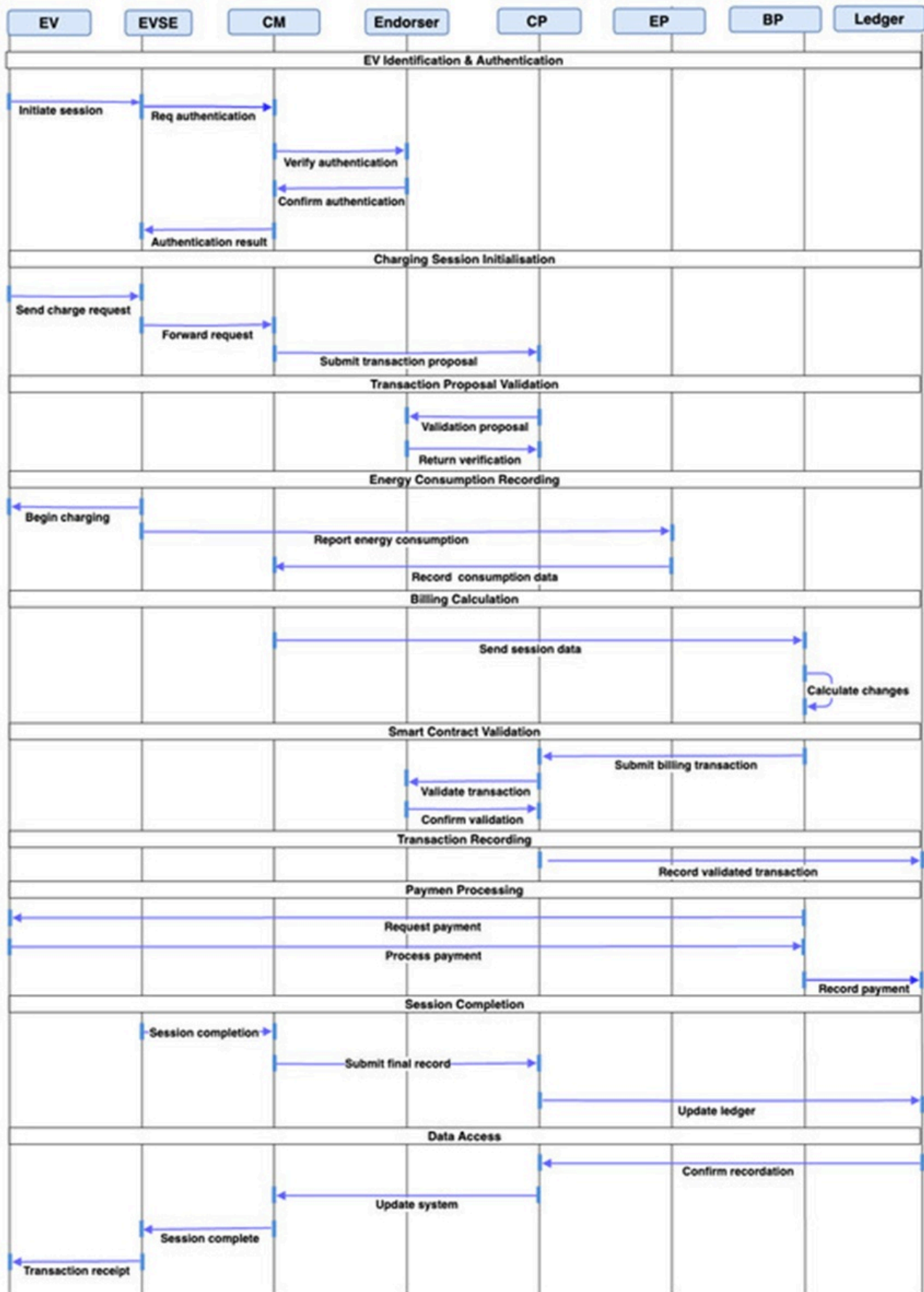


Figure 3. Transactions in the EVCS Network Using Hyperledger Fabric. Abbreviations: EV = electric vehicle, EVSE = electric vehicle supply equipment, CM = charging management, CP = charging provider, BP = billing provider.

The Hyperledger Fabric platform relies on a state database to maintain the present condition of the EV charging ledger. This database operates as a key–value store, recording charging sessions along with associated operations. It commonly utilizes either LevelDB or CouchDB [31] for efficient storage and retrieval of these key–value entries. To further ensure data trustworthiness and enable rapid consistency checks, the system uses a Merkle tree [32]. The Merkle root is derived from the existing key–value entries stored in the state database. Should any unauthorized modification occur in the underlying records, the Merkle root would change accordingly, allowing instant detection and thereby protecting the accuracy and security of all stored information.

Peer nodes—generally managed by charging providers—form a critical part of the EV charging infrastructure. Every peer node maintains a complete replica of the ledger and actively contributes to transaction validation. These nodes run Chaincode (smart contracts) to handle various operations, including initiating charging sessions, verifying EV identities, and processing payments. The ledger stored on each peer node comprises two fundamental components: the world state (global state) and the blockchain. The world state captures the latest snapshot of the ledger as a key–value database. For instance, an individual charging session might be stored as a key–value pair (k, v), where k represents the unique session identifier and v contains relevant details such as duration, energy delivered, and total cost.

The blockchain portion of the ledger serves as a permanent, write-once sequence of all historical transactions. It is organized into blocks, each of which groups multiple transactions. Every block carries a unique identifier in the form of a block hash, computed according to the following:

$$\text{Block_Hash} = \text{Hash}(\text{Block_Header}). \quad (3)$$

Here, the Block Header includes essential metadata, such as the hash of the preceding block, the current timestamp, and the Merkle root summarizing the transactions in the block.

Attack mitigation in EVCS network in the proposed approach

To counter the primary threats specifically false data injection (FDI), man-in-the-middle (MITM), denial-of-service (DoS), and malicious insider (MI) attacks—we introduce a blockchain solution built on Hyperledger Fabric (HF). A comparative evaluation before and after deployment shows a significantly enhanced robustness of the EVCS system, substantially reducing the effects of these threats. The subsections below explain how the proposed smart contract framework effectively neutralizes each attack category.

Mitigating FDI Attacks

Our blockchain-enabled EVCS architecture, implemented using Hyperledger Fabric, effectively mitigates the risks posed by false data injection (FDI) attacks. The design addresses issues such as falsified energy usage reports, fraudulent billing, and interference with grid operations by applying rigorous mathematical validation, cryptographic checks, and automated rule enforcement via smart contracts. As a result, only verified, legitimate information may be added to the blockchain record.

The process starts with cryptographic authentication of every transaction. Each transaction T is digitally signed using the submitting participant's private key K_{priv} . The resulting signature S is produced using the formula:

$$S = \text{Sign}_{K_{\text{priv}}}(H(T)), \quad (4)$$

Where (T) denotes the cryptographic hash of the transaction contents, verification occurs by applying the corresponding public key K_{pub} , which confirms the transaction's origin and prevents unauthorized entities from submitting fabricated data.

A fundamental strength of the architecture is Hyperledger Fabric's consensus protocol. The system adopts a practical Byzantine fault tolerance (PBFT) mechanism [33], requiring a supermajority of endorsements—specifically at least $\lceil 2n/3 \rceil + 1$ valid signatures from a total of n participating nodes—for a transaction to be accepted:

$$\sum_{i=1}^n \mathbb{I}(\sigma_i = \text{valid}) \geq \left\lceil \frac{2n}{3} \right\rceil + 1, \quad (5)$$

Where $(\sigma_i = \text{valid})$ is an indicator function returning 1 if node i has provided a valid endorsement, this threshold ensures that neither a single malicious actor nor a small colluding group can alter or falsify ledger data.

Additionally, the framework incorporates continuous, smart contract-driven anomaly detection. For every reported charging event, the contract compares the measured energy consumption E_r from the charging station against the predicted or expected value E_e derived from usage patterns:

$$|E_r - E_e| < \epsilon, \quad (6)$$

Where ϵ represents a configurable tolerance margin; if the deviation exceeds this limit, the smart contract automatically rejects the transaction, blocking the incorporation of erroneous or manipulated data into the ledger.

Finally, role-based access control (RBAC) restricts blockchain interactions to properly authorized participants only. Each entity P_i receives an assigned role R_i that defines its permitted actions (permissions denoted as (R_i)). The governing access policy is formally expressed as:

$$\forall P_i \in \mathcal{D}, \text{ if } P_i \notin A(R_i), \text{ reject transaction.} \quad (7)$$

This mechanism ensures that compromised credentials belonging to any participant cannot result in unauthorized operations.

The unchangeable nature of the blockchain ledger adds another layer of protection against data accuracy issues. Once information is committed to the blockchain, altering or removing it becomes practically impossible without the agreement of the majority of the network. The Merkle tree organization integrated into the blockchain offers a highly efficient method for identifying unauthorized modifications. Within each block, a Merkle root M_r is computed using the formula:

$$M_r = H(H(T_1) \parallel H(T_2) \parallel \dots \parallel H(T_n)), \quad (8)$$

Where $\parallel$ represents concatenation and (T_i) denotes the hash of transaction T_i . Any alteration to a block's contents would cause a mismatch in the Merkle root, enabling immediate detection of the compromise.

Together, cryptographic signatures, consensus protocols, smart contract logic, role-based access control (RBAC), and permanent ledger storage provide a robust defense that allows the proposed blockchain architecture to successfully

identify and neutralize false data injection (FDI) attacks. Through ongoing verification of incoming transactions and data, the system ensures that solely accurate and authenticated information is recorded.

Mitigating DoS attacks

Within the suggested blockchain-powered EVCS network, denial-of-service (DoS) attacks are mitigated through the integration of consensus protocols, smart contract constraints, and request throttling mechanisms. DoS attacks typically involve malicious entities flooding the system with an excessive volume of charging requests or booking attempts, generating artificial congestion that interferes with normal usage. Our solution leverages Hyperledger Fabric's distributed architecture to detect and neutralize disruptive behaviors as they occur.

Transaction Rate-Limiting Enforced by Smart Contracts: Our framework incorporates rate-limiting functionality directly within smart contracts to regulate how frequently any participant can submit requests. Let N_{req} denote the count of requests issued by an electric vehicle (EV) during a defined time interval τ . The limiting condition is expressed as:

$$N_{req} \leq N_{max}, \quad (9)$$

Where N_{max} is the permitted maximum number of requests allowed within the interval τ , whenever N_{req} exceeds N_{max} , the smart contract automatically denies additional submissions from that EV until the time window is refreshed. This measure stops individual actors from overwhelming the network.

Threshold-Based Transaction Approval: Security is further strengthened by a threshold-driven verification process embedded in the consensus layer. For each transaction T , let p_i^T represent the probability assigned by node i that T is legitimate. Acceptance occurs only when:

$$\sum_{i=1}^n p_i^T > \theta_T, \quad (10)$$

Where n indicates the total number of participating verification nodes, and θ_T is the minimum consensus threshold. Requiring approval from a sufficient number of nodes makes it extremely challenging for attackers to push fraudulent transactions through.

Decentralized Consensus via PBFT: The network employs the practical Byzantine fault tolerance (PBFT) algorithm to safeguard against malicious behavior. PBFT mandates that at least two-thirds of the nodes concur on a transaction's validity before its inclusion in the ledger. The consensus requirement is formalized as:

$$\left\lceil \frac{2n}{3} \right\rceil + 1 \leq N_{valid}, \quad (11)$$

Where N_{valid} is the count of nodes providing a valid endorsement, and N is the total number of nodes. By demanding a supermajority, this approach ensures the system remains operational and trustworthy even when some nodes are compromised or malfunctioning.

Mitigating MI attacks

Our blockchain-based EVCS architecture counters malware injection (MI) attacks by combining strong cryptographic checks, immutable audit logs, strict access governance, and continuous anomaly monitoring. The design ensures that firmware updates can be applied only when properly authorized, while every modification to the system is permanently

and securely recorded on the blockchain—offering both preventive barriers and effective post-incident traceability against malicious code.

To confirm the legitimacy of any firmware update, a cryptographic hash is computed for each firmware version f before deployment:

$$H(f) = \text{hash}(f), \quad (12)$$

Where $H(f)$ is the secure hash of the firmware, this hash is recorded on the blockchain, and subsequent update attempts are permitted only if the newly submitted firmware produces an identical hash. By enforcing this match, the system blocks the installation of tampered or malicious firmware, thereby preventing malware from being introduced into the charging infrastructure.

Every modification or update to the software running on EVSE or CSMS components is permanently logged on the blockchain, establishing an unalterable audit trail. Each firmware update is documented in the form of a record structured as:

$$\text{AuditRecord} = \{\text{timestamp}, \text{firmwareversion}, H(f), \text{installerID}\}, \quad (13)$$

Where the installer ID specifies the authorized entity responsible for the deployment. This comprehensive logging guarantees complete traceability, enabling all stakeholders to confirm that every change was properly authorized and effectively blocking any unauthorized alterations.

Hyperledger Fabric implements strict role-based access control (RBAC), restricting access to and modification of critical system elements to only those individuals who possess the necessary certifications and credentials. Permissions are managed through access control lists (ACLs) that are themselves stored immutably on the blockchain:

$$A(P_i) = \{R_1, R_2, \dots, R_n\}, \quad (14)$$

Where $A(P_i)$ denotes the set of roles granted to participant P_i ; if a participant lacks the required role, access is immediately denied, significantly reducing the likelihood of illicit operations.

The blockchain network continuously monitors for irregular patterns, including sudden, unexplained price variations during active charging sessions that could signal attempts at price tampering. Smart contracts carry out instantaneous transaction validation based on the rule:

$$\text{if } P_{\text{current}} < P_{\text{threshold}}, \text{flag session for review}, \quad (15)$$

Where P_{current} is the observed price applied during the charging process, and $P_{\text{threshold}}$ represents the anticipated price derived from predictive modeling. This real-time checking mechanism enables early identification of irregularities and helps limit potential harm.

Mitigating MITM attacks

To prevent interception and alteration of messages exchanged among participants, all data transmissions among EVs, EVSE units, CSMS, and billing systems are protected by robust encryption standards, including the Advanced Encryption Standard (AES). This cryptographic protection ensures that intercepted communications remain confidential and cannot be deciphered or manipulated by unauthorized parties.

Every interaction involving EVs, EVSE devices, and CSMS is durably recorded within the immutable blockchain ledger. This permanent record functions as a powerful forensic resource should a man-in-the-middle (MITM) attack be suspected. Auditors and investigators can review and reconstruct the sequence of communications, pinpoint the location of any compromise, and evaluate the full extent of the intrusion.

Smart contracts enforce mutual authentication protocols to verify the legitimacy of both endpoints—such as an EV and an EVSE—before permitting sensitive data transfer. The mutual authentication procedure is formalized as:

$$\text{IfAuth}(EV) \wedge \text{Auth}(EVSE) \text{ then proceed with communication.} \quad (16)$$

This bidirectional verification step guarantees that data exchange occurs exclusively between properly authenticated entities, effectively blocking impostors from inserting themselves into the communication path.

An extra safeguard is provided by Hyperledger Fabric's consensus protocol, which requires that critical communication records—such as charging session parameters or billing details—undergo validation by multiple independent peer nodes. This distributed checking process substantially reduces opportunities for data manipulation. The expression captures the consensus requirement:

$$\sum_{i=1}^n p_i^c > \theta_c \quad (17)$$

Where p_i^c denotes the probability that peer i considers the communication data valid, and θ_c is the minimum consensus threshold required for acceptance. Any attempt to tamper with the transmitted information is thereby detected and rejected by the network.

By integrating strong encryption, digital signatures, permanent, immutable logging, mutual authentication procedures, and multi-peer consensus validation, the proposed blockchain-supported EVCS architecture delivers comprehensive defense against man-in-the-middle (MITM) attacks. These layered protections ensure secure communication channels, maintain the integrity of all exchanged data, and restrict participation to verified, legitimate entities.

Firmware update challenges with offline devices

Devices or entities that remain offline present significant obstacles for delivering firmware and software updates. In a blockchain-enabled update system, each firmware modification is permanently stored as a transaction within the blockchain ledger. As soon as an offline device regains connectivity and powers on, it contacts the blockchain network to fetch the current ledger state. The underlying blockchain protocol automatically ensures the device obtains every block created during its offline period and fully reconciles its local ledger with the network's version. Integrity of the update is confirmed through cryptographic checks, including digital signatures and hash comparisons. Following successful validation, the relevant blocks are appended to the device's local ledger. Network-wide agreement and consistency are maintained through consensus algorithms (e.g., RAFT [34] and PBFT [35]). Should any ledger conflicts or divergent updates arise, they are settled strictly in accordance with the chosen consensus mechanism. **Figure 4** depicts a detailed flowchart of the firmware update sequence for EVSE in both connected and disconnected scenarios, highlighting blockchain-based validation, consensus participation, and explicit rejection pathways.

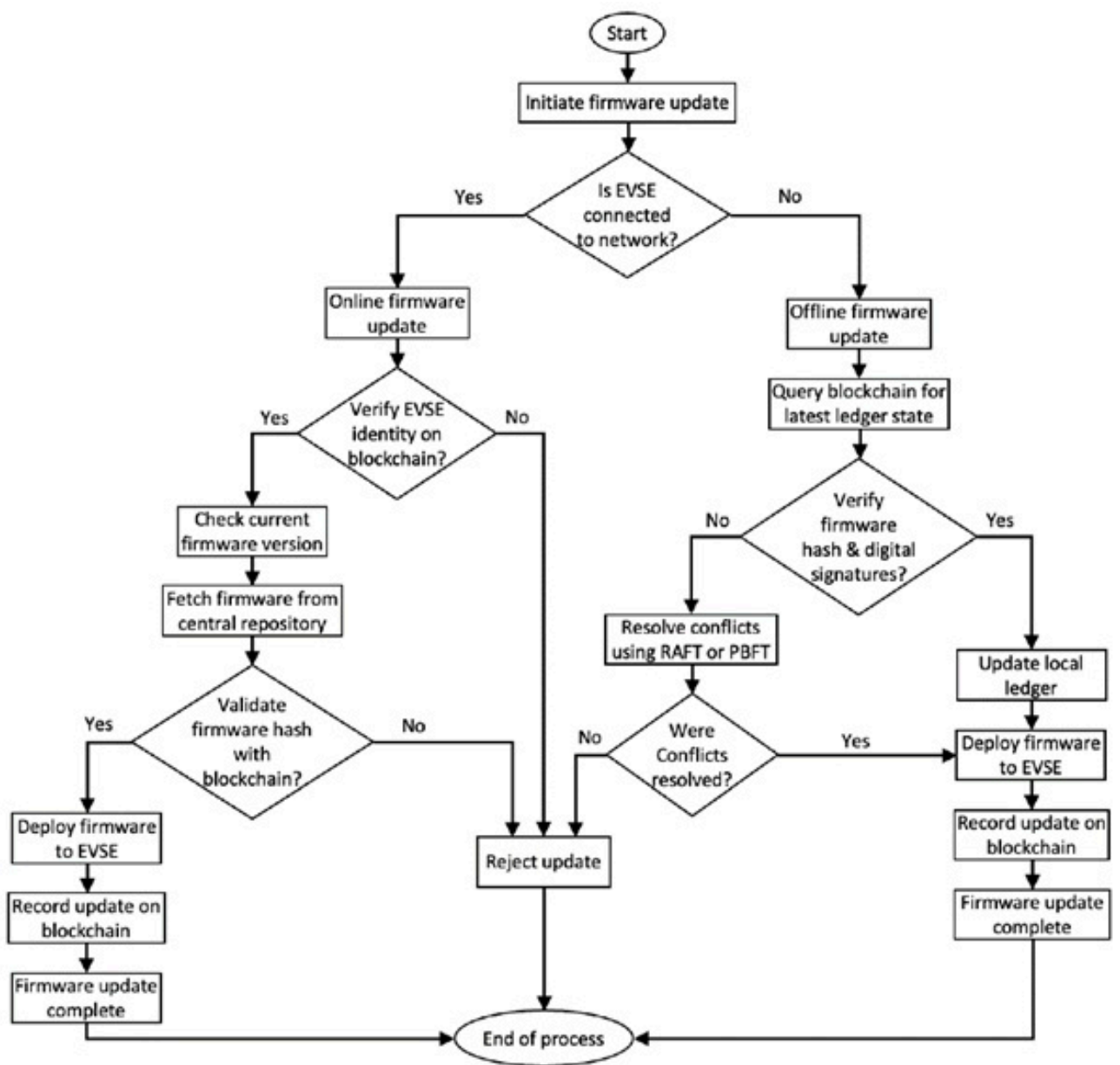


Figure 4. Firmware update flow for electric vehicle supply equipment (EVSE).

Blockchain Deployment and Defense Against Cyber Threats

The subsections below outline the actual implementation of our smart contract within the blockchain environment and illustrate its effectiveness in countering multiple categories of cyberattacks targeting the EVCS infrastructure.

A range of blockchain frameworks can serve as a foundation, including Hyperledger Fabric, Ethereum, Corda, Quorum, and Multichain. Evaluation of these platforms typically revolves around key characteristics: permissioned versus permissionless design, throughput capacity, transaction delay, data privacy guarantees, consensus approach, and ecosystem maturity. Insights drawn from prior research [36-39] are summarized in **Table 3**, which compares the leading platforms across the aforementioned dimensions.

Table 3. Comparison of major blockchain implementation platforms.

Feature	Hyperledger fabric	Ethereum	Corda	Quorum	Multichain
Category	Permissioned	Public	Permissioned	Permissioned	Permissioned
Privacy level	High (utilizes channels and private data collections)	Low (data recorded on a public ledger)	Medium (peer-to-peer data sharing)	Medium (peer-to-peer data sharing)	Medium
Consensus mechanism	Configurable (e.g., RAFT)	Proof-of-Stake	RAFT	RAFT	Proof-of-Authority
Scalability capacity	High	Medium	Medium	Medium	Medium
Smart contract capability	Comprehensive SDKs (Go/Java)	Solidity	Java/Kotlin	Solidity	Limited functionality
Community and ecosystem support	Strong (backed by Linux Foundation)	Strong	Moderate	Moderate	Limited

After careful assessment of these criteria, Hyperledger Fabric emerges as the optimal selection for the present application due to the following advantages. Electric vehicle charging ecosystems generally consist of identifiable, trusted participants (such as charging station operators and regulatory bodies), rendering a permissioned blockchain far more appropriate than an open, public network like Ethereum. Hyperledger Fabric provides robust security and governance controls—attributes indispensable for critical energy infrastructure. Real-time responsiveness is paramount in EV charging applications, demanding both high transaction throughput and very low latency; Hyperledger Fabric delivers excellent performance in both areas, significantly outperforming platforms that provide only average scalability. Moreover, EV charging systems handle highly sensitive user information and operational telemetry that must remain strictly confidential—Hyperledger Fabric’s sophisticated privacy mechanisms offer superior protection in this regard. Finally, seamless integration into a complex, heterogeneous EVCS environment necessitates extensive tailoring of blockchain behavior; Hyperledger Fabric’s modular, plug-and-play architecture, combined with substantial community backing, makes it exceptionally well-suited for such customization. Public platforms like Ethereum, by contrast, suffer from reduced flexibility owing to their broad, generalized design.

The blockchain-supported EVCS network was implemented using Hyperledger Fabric version 2.5.10, hosted on Ubuntu 20.04. A carefully selected software ecosystem was employed to maximize performance and expandability. CouchDB version 3.3.3 served as the state database, chosen for its efficient handling of rich, dynamic key–value data structures typical in transaction processing. Containerization and orchestration relied on Docker 19.03.12 and Docker Compose 1.27.2, facilitating the reliable deployment of distributed microservices. Chaincode (smart contracts) and associated transaction-handling routines were written in Go version 1.23.1 and Python version 3.12.4, delivering the performance and expressiveness required for sophisticated blockchain logic. All experiments and simulations were conducted on a Dell workstation purchased in Adelaide, Australia, equipped with an Intel Core i7-14700F CPU (20 cores/28 threads, maximum turbo frequency 5.4 GHz), 16 GB DDR3 RAM, and a 512 GB hard disk drive. This hardware configuration provided ample computational capacity to accommodate intensive blockchain workloads, with performance under realistic transaction volumes confirmed through dedicated benchmarking using Hyperledger Caliper.

The simulation replicated a comprehensive network environment that included eight charging providers, two administrative accounts, two maintenance teams, four billing systems, one energy supplier, three service providers, 50 EVSE units distributed across four EVCS locations, and 350 electric vehicles (EVs). Every charging station received a unique identifier in the blockchain system, enabling secure, tamper-proof transaction processing. The architecture

enforced strict role-based access controls across all entities, thereby improving overall system dependability and protecting data accuracy.

Each EVSE provided charging sessions with a rated power output of 50 units, while actual energy usage per session ranged from 10 units/h to 50 units/h, depending on the EV's specific needs. A fixed pricing structure of USD 0.20 per unit was applied, implemented as an adjustable parameter to support potential shifts toward time-varying or dynamic electricity tariffs. Transaction workflows were managed entirely by smart contracts, which verified energy delivered and corresponding charges in real time while preserving the authenticity and non-repudiation of all recorded data.

Performance and security evaluations were divided into two distinct stages. The initial stage assessed baseline performance by processing only legitimate transactions, analyzing metrics such as throughput, latency, and resource consumption at transaction loads of 50, 100, and 500. Detailed findings from this evaluation appear in the subsequent section. In the second stage, the system was subjected to 200 fabricated transactions embedding intentionally malicious data to assess its ability to identify and neutralize cyber threats. These injected transactions replicated multiple attack vectors. The effectiveness of the blockchain framework in detecting and countering these adversarial inputs was thoroughly examined in the following section, underscoring its robustness and fault tolerance against security risks.

Our blockchain-enhanced EVCS network incorporated simulated attack vectors to represent realistic malicious behaviors, including false data injection (FDI), man-in-the-middle (MITM), denial-of-service (DoS), and malware injection (MI).

FDI attack

In this scenario, an adversary intercepts and alters communication between the EVSE and the CSMS, falsifying the recorded energy usage. For instance, the attacker could change a session's reported consumption from the true value of 20 units to an inflated 50 units, leading to unauthorized overcharges.

The primary goal of an FDI attack is to distort billing records by corrupting energy consumption figures en route, potentially inflicting substantial economic damage and undermining user confidence. The blockchain layer counters this by rigorously validating incoming data and discarding any inconsistencies.

Representative examples of legitimate versus altered data payloads are provided below:

```
1  Original Payload:
2  {
3      "session_id": 12345,
4      "ev_id": "EV_6789",
5      "energy_consumed": 20,
6      "billing_amount": 4.0,
7      "timestamp": "2024-10-21T10:15:00Z"
8  }
9  After FDI Attack:
10 {
11     "session_id": 12345,
12     "ev_id": "EV_6789",
13     "energy_consumed": 50,
14     "billing_amount": 10.0,
15     "timestamp": "2024-10-21T10:15:00Z"
16 }
```

Hyperledger Fabric is utilized to thwart false data injection attacks via smart contract-enforced consistency rules. Within our design, energy usage and billing figures undergo multi-layered cryptographic and logical scrutiny. The deployed smart contract applies strict validation during transaction submission:

```
1  contract ChargingSession {
2      struct Session {
3          uint256 sessionId;
4          address evAddress;
5          uint256 energyConsumed;
6          uint256 billingAmount;
7          uint256 timestamp;
8      }
9
10     mapping(uint256 => Session) public sessions;
11
12     event SessionLogged(uint256 sessionId, uint256 energy, uint256 billingAmount);
13
14     function logSession(uint256 _sessionId, uint256 _energyConsumed, uint256 _billingAmount) public {
15         require(_energyConsumed > 0 && _energyConsumed <= 500, "Invalid energy value");
16
17         uint256 expectedBilling = _energyConsumed * 0.20; // $0.2 per energy unit
18         require(_billingAmount == expectedBilling, "Billing mismatch");
19
20         sessions[_sessionId] = Session(_sessionId, msg.sender, _energyConsumed, _billingAmount,
21             block.timestamp);
22         emit SessionLogged(_sessionId, _energyConsumed, _billingAmount);
23     }
24 }
```

The contract mandates compliance with predefined thresholds—for instance, energy consumption must lie between 1 and 500 units, and the billed amount must precisely align with the product of consumption and the established rate. Non-conformance triggers automatic rejection of the transaction, thereby upholding data reliability.

The attack simulation involves intentionally altering data in transit. The smart contract cross-checks energy consumption against the derived billing value; mismatches prompt immediate rejection and audit logging for review.

```
1  Log Entry:
2  {
3      "session_id": 12345,
4      "status": "rejected",
5      "reason": "Billing mismatch",
6      "timestamp": "2024-10-21T10:20:00Z"
7  }
```

Through this mechanism, every transaction receives a permanent, cryptographically secured entry on the blockchain, guaranteeing immutability and full auditability. Detection of any manipulated element activates the smart contract to block ledger inclusion, while the Hyperledger Fabric consensus process ensures that unauthorized modifications cannot propagate undetected.

Denial-of-service (DoS) attack

This attack is reproduced in the simulation by issuing a high volume of charging requests in a very compressed timeframe. A sample payload used during the attack appears as follows:

```

1  {
2      "session_id": 1,
3      "ev_id": "EV_7890",
4      "request_time": "2024-10-21T12:00:00Z"
5  }

```

To protect against denial-of-service attacks, the blockchain-based electric vehicle charging station (EVCS) architecture we propose integrates a smart contract that imposes strict rate limits on incoming charging requests. The contract limits each participant to a fixed number of requests within a fixed-duration time slot. The corresponding rate-limiting rule is specified here:

```

1  contract RequestRateLimiter {
2      mapping(address => uint256) public requestCount;
3      uint256 public constant TIME_WINDOW = 3600; // 1 hour
4      uint256 public constant MAX_REQUESTS = 5;
5
6      function requestCharge() public {
7          require(block.timestamp - requestCount[msg.sender] >= TIME_WINDOW,
8              "Too Many Requests");
9          requestCount[msg.sender] = block.timestamp;
10     }
11 }

```

The smart contract implements rate limiting by ensuring that each new request arrives within the currently valid time interval. Whenever any participant surpasses the permitted request count, the contract discards subsequent requests until the next reset of the time window. This mechanism effectively blocks adversaries from saturating or disrupting the network through request flooding.

The criterion used to accept an incoming request is mathematically described as:

$$\text{if } T_{\text{current}} - T_{\text{last}} \geq \Delta_t \text{ then accept request,} \quad (18)$$

Where T_{current} stands for the present system timestamp, T_{last} corresponds to the timestamp recorded for that participant's previous successfully accepted request, and Δ_t defines the fixed duration of each time window. This formulation limits the request rate per participant and substantially reduces the likelihood of network congestion or overload.

We conduct the DoS experiment by dispatching numerous charging requests in very short intervals. When the rate limit is exceeded, the smart contract refuses to process any additional requests and creates a permanent record of the violation. An illustrative log entry generated for a blocked request is shown below:

```
1 {  
2   "ev_id": "EV_7890",  
3   "status": "denied",  
4   "reason": "Rate Limit Exceeded",  
5   "timestamp": "2024-10-21T12:01:00Z"  
6 }
```

This design deliberately avoids penalizing normal users who stay within acceptable usage patterns. All successfully processed transactions are permanently stored and tamper-proof on the blockchain ledger, providing full auditability and clear provenance. Because request validation occurs through consensus among multiple distributed nodes running Hyperledger Fabric, no single node can accept harmful transactions, thereby greatly improving the system's overall robustness against denial-of-service attacks.

Malware injection (MI) attack

During the simulation, the adversary attempts to deploy altered firmware whose cryptographic hash deviates from the genuine version. The hash values computed for both the legitimate original firmware and the malicious injected version are listed here:

```
1 H(f_original) = "0x123abc..."  
2 H(f_injected) = "0x456def..."
```

To block malware injection attempts, our proposed blockchain-integrated EVCS framework relies on smart contracts to enforce firmware authenticity verification. The contract allows an EVSE to install a firmware update exclusively when the provided hash exactly matches the authorized value already registered for that device. The logic implemented inside the smart contract for controlling firmware update operations is detailed below:

```
1 contract FirmwareUpdate {  
2   mapping(address => bytes32) public firmwareHash;  
3  
4   event FirmwareUpdated(address evseId, bytes32 hash);  
5  
6   function updateFirmware(address evseId, bytes32 hash) public {  
7     require(firmwareHash[evseId] == hash, "Unauthorized Firmware");  
8     firmwareHash[evseId] = hash;  
9     emit FirmwareUpdated(evseId, hash);  
10  }  
11 }
```

The smart contract maintains a dedicated mapping that associates each EVSE with its currently approved firmware hash. Any proposed firmware update is accepted only if the candidate firmware's hash precisely matches the hash already stored on-chain. In the event of a hash mismatch, the smart contract immediately halts the update process, thereby stopping installation of rogue or tampered software.

The validation rule for permitting a firmware update can be written as:

$$H(f_{\text{uploaded}}) = H(f_{\text{stored}}), \quad (19)$$

Where $H(\text{uploaded})$ denotes the cryptographic hash computed over the firmware file being submitted, and $H(\text{stored})$ represents the reference hash value preserved immutably on the blockchain. Failure to satisfy this equality results in the update transaction being rejected outright.

We emulate the malware injection attack by submitting a tampered firmware binary that produces a different hash. The smart contract correctly identifies the inconsistency and denies the installation. Every such denial is durably logged on the blockchain, ensuring complete traceability of malicious attempts. A representative example of the rejection log entry is presented below:

```
1 {  
2   "evse_id": "EVSE_001",  
3   "status": "rejected",  
4   "reason": "Unauthorized Firmware",  
5   "timestamp": "2024-10-21T13:00:00Z"  
6 }
```

Thanks to the blockchain's append-only and tamper-resistant properties, every firmware-related action—successful or blocked—remains permanently verifiable and auditable. Detection of repeated or suspicious unauthorized update attempts enables the system operators to launch forensic analysis, trace the attacking entity, and implement preventive countermeasures. Enforcing firmware integrity checks exclusively through smart contract logic guarantees that only verified, manufacturer-approved firmware can ever be activated on any EVSE, thereby providing strong protection against malware-based compromise.

Man-in-the-middle (MITM) attack

A man-in-the-middle attack occurs when a malicious actor secretly intercepts and tampers with the communication between two legitimate parties. In this context, the attacker modifies the intended payment destination address within a transaction, aiming to divert the funds to their own wallet without alerting the sender or receiver.

An illustration of the legitimate transaction alongside the intercepted version is presented below, highlighting how the attacker substitutes the payment address with their own:

```
1 Original Transaction:  
2 {  
3   "transaction_id": 9876,  
4   "payment_address": "0xProviderAddress",  
5   "amount": 20.0  
6 }  
7 After MITM Attack:  
8 {  
9   "transaction_id": 9876,  
10  "payment_address": "0xAttackerAddress",  
11  "amount": 20.0  
12 }
```

To defend against man-in-the-middle attacks, the blockchain-powered EVCS network we propose incorporates advanced cryptographic mechanisms and smart contract enforcement to ensure transaction integrity. Every transaction undergoes authentication through digital signature verification. The relevant smart contract code responsible for payment validation is outlined below:

```
1  contract PaymentValidation {
2      struct Payment {
3          uint256 transactionId;
4          address paymentAddress;
5          uint256 amount;
6      }
7
8      mapping(uint256 => Payment) public payments;
9
10     event PaymentProcessed(uint256 transactionId, address paymentAddress);
11
12     function processPayment(
13         uint256 _transactionId,
14         uint256 _amount,
15         address _paymentAddress
16     ) public {
17         bytes32 messageHash = keccak256(
18             abi.encodePacked(_transactionId, _amount, _paymentAddress)
19         );
20         require(verifySignature(messageHash, msg.sender), "Invalid Signature");
21
22         payments[_transactionId] = Payment(_transactionId, _paymentAddress, _amount);
23         emit PaymentProcessed(_transactionId, _paymentAddress);
24     }
25
26     function verifySignature(bytes32 hash, address signer)
27         private pure returns (bool) {
28         return signer != address(0);
29     }
30 }
```

This smart contract provides cryptographic confirmation of each payment via digital signatures. It computes a message hash using the keccak256 function, combining the transaction identifier, the transferred amount, and the designated payment address—as indicated in line #18 of the provided code snippet. Verification of the digital signature against the sender's public identity confirms that the content has not been altered since its origin.

To assess the robustness of this protective measure, we emulate a man-in-the-middle scenario by altering the payment address mid-transmission. Upon detecting the manipulated address via signature mismatch, the smart contract discards the transaction. The corresponding blockchain log entry for the denied transaction is shown below:

```
1  {
2      "transaction_id": 9876,
3      "status": "rejected",
4      "reason": "Invalid Signature or Tampered Payment Address",
5      "timestamp": "2024-10-21T11:05:00Z"
6  }
```

This methodology ensures robust validation of all payment transactions, with any signs of interception or tampering promptly identified and halted. Immutable blockchain records provide complete audit trails and transparency, enabling thorough examination of every transaction and swift detection of suspicious or fraudulent behavior.

Performance Testing Using Hyperledger Caliper

The performance characteristics of the smart contracts developed to secure the EVCS were assessed using Hyperledger Caliper, a dedicated benchmarking tool for evaluating blockchain systems.

Performance metrics

The evaluation relies on the following standard performance indicators.

Transaction throughput test

This evaluation determines the maximum number of transactions the network can handle per second (TPS). Achieving a high TPS is vital for supporting realistic usage patterns, including high-demand periods when numerous electric vehicles simultaneously request charging. The ability to process concurrent requests efficiently maintains uninterrupted service delivery. The objective of this test is to identify the highest sustainable TPS level before any noticeable drop in performance occurs, by simulating varying numbers of EVs submitting requests under increasing load conditions to examine scalability limits.

Latency test

This measurement captures the duration from the moment a transaction is submitted until it receives final confirmation. Minimizing latency is essential for time-sensitive functions such as starting a charging session, performing authentication, and completing payments. Preserving low confirmation times even during heavy network usage delivers a responsive and satisfactory experience for users, particularly in busy scenarios. The test examines how well the blockchain sustains acceptable latency across a range of workload intensities.

Resource utilization test

This assessment tracks CPU, memory, and network bandwidth consumption throughout blockchain operations. Optimal resource efficiency helps preserve system reliability and avoid disruptions caused by resource depletion. Monitoring is performed on peer nodes, ordering service nodes, and certificate authority components under varying workloads to verify consistent behavior without unsustainable resource demands.

Scalability test

This evaluation assesses the network's ability to maintain strong performance as the number of participating nodes and users increases. As EV charging infrastructure grows, the underlying blockchain must handle increased scale without significant degradation. In the conducted experiments, the test begins with a limited number of nodes. It progressively expands the network, recording TPS and latency values across different sizes and configurations to confirm efficient scaling with minimal impact on response times.

Test cases

We assessed system behavior using three carefully designed test cases that progressively increase transaction load.

Test Case 1 applies a load of 50 transactions to simulate quiet operational periods with low activity. It serves as a reference point for measuring baseline capabilities when the network experiences only light pressure.

Test Case 2 raises the volume to 100 transactions, representing typical busy conditions in which several users simultaneously interact with charging stations.

Test Case 3 escalates to 500 transactions, replicating crowded, high-demand situations in which large numbers of electric vehicles attempt to initiate charging at nearly the same time.

By stepping through these increasing intensities, the experiments reveal how the platform responds to increasing pressure, highlight emerging limitations, and clarify the point at which performance might begin to degrade as activity scales up.

Results interpretation

Experimental data demonstrate that the blockchain-enabled EV charging platform delivers consistent, robust performance across widely varying workloads, as summarized in **Tables 4-7**. Both transaction processing speed and response times show only small changes, confirming stable and user-friendly behavior regardless of demand fluctuations.

Table 4. Performance comparison of Caliper tests across 50, 100, and 500 transactions.

Concurrent users	Successful requests	Failed requests	Send rate (TPS)	Max latency (s)	Min latency (s)	Avg latency (s)	Actual throughput (TPS)
500	500	0	5.1	1.11	0.07	0.59	5.0
100	100	0	5.3	1.10	0.06	0.59	5.2
50	50	0	5.6	1.11	0.09	0.60	5.5

Table 5. Caliper performance report for 50 transactions.

Component	Max CPU (%)	Avg CPU (%)	Max memory (MB)	Avg memory (MB)	Inbound traffic (KB)	Outbound traffic (KB)
org1	0.91	0.3	67.7	67.1	105	50.2
org2	0.52	0.17	69.1	68.5	105	51.2
peer0.org2	0.55	0.33	98.0	97.6	436	261
peer0.org1	0.91	0.44	128	127	912	813
orderer	0.3	0.11	50.4	50.4	268	521
couchdb1	0.46	0.28	69.4	68.6	41.6	38.1
couchdb0	0.4	0.26	75.2	74.2	42.6	38.3

Table 6. Caliper performance report for 100 transactions.

Component	Peak CPU (%)	Average CPU (%)	Peak memory (MB)	Average memory (MB)	Inbound traffic (MB)	Outbound traffic (MB)
org1	0.6	0.25	67.6	66.8	0.207	0.0988
org2	0.52	0.2	69.1	67.9	0.206	0.100
peer0.org2	0.65	0.34	97.4	93.6	0.851	0.509
peer0.org1	0.99	0.74	128	116	1.76	1.57

orderer	0.12	0.07	53.6	51.6	0.522	1.01
couchdb1	1.54	0.58	69.8	69.1	0.107	2.57
couchdb0	0.66	0.27	77.4	74.7	0.0822	0.0739

Table 7. Caliper performance report for 500 transactions.

Component	Maximum CPU (%)	Average CPU (%)	Maximum memory (MB)	Average memory (MB)	Inbound traffic (MB)	Outbound traffic (MB)
org1	1.05	0.41	69.9	69.1	1.03	0.489
org2	0.62	0.31	72.3	70.4	1.03	0.507
peer0.org2	0.89	0.51	107	105	4.23	2.51
peer0.org1	1.05	0.73	132	131	8.73	7.70
orderer	0.24	0.15	57.4	54.2	2.60	5.07
couchdb1	0.92	0.56	69.5	68.5	0.405	0.368
couchdb0	0.89	0.51	73.8	72.1	0.408	0.368

According to [Table 4](#), throughput experiences a gentle downward trend as load grows: 5.5 TPS with 50 transactions, 5.2 TPS with 100 transactions, and 5.0 TPS with 500 transactions. The submission rate follows a parallel decline, moving from 5.6 TPS down to 5.1 TPS. Importantly, despite this slight reduction, the network completes every transaction in all three cases with no losses or errors, reflecting very high operational dependability.

Response-time measurements stay remarkably steady. Peak latency remains close to 1.11 s across the board, the lowest observed latency fluctuates narrowly between 0.06 s and 0.09 s, and mean latency is virtually constant at 0.59–0.60 s ([Table 4](#)). These tightly grouped figures indicate that critical real-time functions—such as starting a charging session or settling payments—proceed smoothly with no perceptible lag, delivering a consistently positive experience whether the network is quiet or heavily loaded.

Resource consumption figures for the lightest load ([Table 5](#)) show peak CPU demand of 0.91% on peer nodes and a maximum memory footprint of 128 MB. Network usage peaks at 912 KB incoming and 813 KB outgoing—well within comfortable limits.

At the moderate load of 100 transactions ([Table 6](#)), CPU utilization rises modestly to 0.99%, memory consumption remains at 128 MB for peers, and network traffic increases but stays easily manageable, preserving smooth execution.

Under the heaviest load of 500 transactions ([Table 7](#)), CPU usage reaches only 1.05%, memory climbs to 132 MB on peer nodes, and network traffic scales to 8.73 MB inbound and 7.70 MB outbound. Even with these higher demands, the system exhibits no instability, crashes, or capacity exhaustion. Completing the entire 500-transaction batch in merely 19.607 seconds while using less than 2% of CPU capacity underscores how lightweight and resource-efficient the proposed design truly is.

[Figure 5](#) compares typical and peak resource consumption for Org1 across the different transaction volumes. [Figure 5a](#) shows that average CPU usage never exceeds 0.5% even at the highest load, while average memory usage steadily rises to 68 MB when handling 500 transactions. [Figure 5b](#) focuses on maximum values, revealing a CPU ceiling of

1.05% and a memory high of 70.1 MB at 500 transactions. Together, these charts confirm that the architecture absorbs large transaction surges with only negligible additional strain on hardware resources.

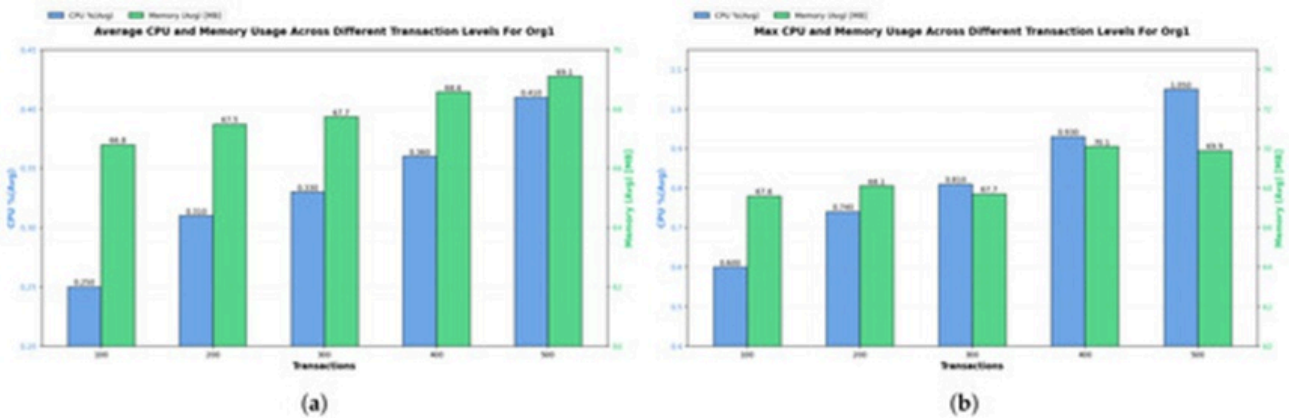


Figure 5. Comparison of average and maximum CPU and memory usage across different transaction volumes for Org1: (a) Average CPU and memory usage, and (b) maximum CPU and memory usage.

The network demonstrates strong scalability, as evidenced by consistent performance across all tested cases. Even though throughput experiences a modest drop as transaction load increases (Table 4), the consistent latency and efficient resource usage clearly support the feasibility of horizontal expansion. These outcomes indicate that the underlying blockchain platform is appropriately designed to handle the anticipated expansion of the electric vehicle charging infrastructure, delivering reliable performance as both charging stations and user numbers continue to grow.

The EV charging system built on blockchain technology demonstrates effective resource management and reliable operation across a wide range of workload intensities. Stable latency values, zero transaction failures, and controlled resource demands collectively prove the platform's readiness to meet practical charging requirements in real-world settings. The scalability validated through these evaluations ensures the network remains capable of accommodating future increases in demand while maintaining consistent and uninterrupted functionality.

During the second stage of testing, the proposed blockchain-based EVCS architecture was assessed for its transaction-processing capacity and effectiveness in identifying multiple types of cyberattacks. A total of 700 transactions were simulated: 500 consisted of standard, non-malicious operations with no irregularities, while the remaining 200 were intentionally crafted to include malicious behavior. Legitimate transactions were used to confirm accurate and uninterrupted recording of valid information. The attack-containing transactions allowed evaluation of the framework's detection and response capabilities, leveraging the security features embedded in the smart contract design—namely, cryptographic verification, rate-limiting controls, and data integrity safeguards.

Simulation findings are outlined in Table 8. The table presents outcomes for 700 simulated transactions on the EVCS network, including 500 legitimate transactions and 200 transactions that embed purposeful attack patterns. Every one of the 500 normal transactions was handled successfully with no irregularities identified, underlining the system's trustworthiness when processing authorized activity. In the case of the 200 attack-laden transactions, the framework correctly flagged all 50 occurrences of each attack category and produced no false-positive alerts. Through its integrated blockchain safeguards—smart contracts, cryptographic checks, and rate-limiting enforcement—all malicious attempts were successfully detected and neutralized.

Table 8. Summary of results for simulated transactions and attack detection in the EVCS network.

Transaction type	Transaction count	Successful transactions	Anomalies detected	False positives	Comments
Normal (No anomalies)	500	500	0	0	Every transaction was completed successfully.
FDI attack	50	0	50	0	All FDI attempts were correctly identified.
DoS attack	50	0	50	0	Excessive requests were stopped by rate limiting.
MI attack	50	0	50	0	Unauthorized firmware update attempts were prevented.
MITM attack	50	0	50	0	Payment tampering attempts were successfully caught.
Overall	700	500	200	0	The system showed precise attack detection and effective mitigation.

Although the immutable nature of the blockchain ledger protects the integrity of every committed transaction, initial detection can occasionally fail. In live production environments, determined adversaries might leverage zero-day exploits, previously undisclosed flaws, or highly refined evasion strategies that briefly circumvent protective measures. Forensic examination offers a valuable complementary approach to detection in such cases. Blockchain ledger forensics [40] enables the reconstruction and tracing of illicit activity thanks to its permanent, tamper-proof records. Analysis of peer and orderer logs (including chaincode execution traces), identity audits for certificate abuse, and queries of historical chain states can all uncover signs of compromise [41]. After forensic review identifies previously missed intrusions, remediation can involve revoking affected credentials, tightening permission rules, and patching vulnerable chaincode segments. Insights gained from forensics can also guide improvements to smart contract rules—such as adding more stringent checks, adjusting endorsement requirements, or implementing adaptive logic based on predictive modeling [42]—thereby strengthening long-term security and system robustness.

Further analysis

The presented blockchain design is deployable across several major cloud environments, including IBM, Amazon AWS, Google Cloud, and Microsoft Azure [43]. The monetary cost of electricity and energy exchanges between EV supply equipment (EVSE) and vehicle users can be compared to the operational expense of hosting and running the proposed solution in a cloud setting. Performance benchmarking with Caliper for this model recorded a peak average CPU usage of 0.73% (on peer0.org1) and maximum memory consumption of 132 MB during batches of 500 transactions. Network usage for the same 500-transaction load reached 17.403 MB inbound and 16.972 MB outbound. Preliminary tests were run on hardware comprising an Intel 14th Gen Core i7-14700F (20 cores/28 threads, up to 5.4 GHz), 16 GB DDR3 RAM, and a 512 GB HDD. Projecting the same workload to a daily volume of roughly 100,000 transactions would require approximately 26.2 GB of memory, 3.48 GB of inbound traffic, and 3.39 GB of outbound traffic.

For an AWS-based deployment [44], the t4g.large instance family (8 GiB memory, priced at USD 0.0672 per hour) offers a suitable option. To meet the projected 26.2 GB of memory, four such instances would be needed. Running four t4g.large instances continuously would generate a daily compute expense of USD 6.45, resulting in an estimated monthly compute cost of USD 193.54.

The combined daily network traffic (inbound plus outbound) is projected at 6.87 GB. Given AWS's outbound data transfer rate of USD 0.09 per GB, the daily outbound traffic expense is approximately USD 0.62, resulting in a monthly traffic cost

of USD 18.55. Inbound traffic incurs no charges from AWS. When adding compute costs to outbound traffic expenses, the overall estimated monthly operating cost for this blockchain model on AWS reaches about USD 212.09.

Limitations and challenges

The proposed system implementation faces several limitations and challenges:

- Although our blockchain computations exhibit low latency, external factors in the data communication network—such as signal interference, congestion, bandwidth inconsistencies, and differing quality-of-service levels across segments—can introduce packet delays outside the blockchain's control. These delays may hinder real-time execution of EVCS transactions, potentially failing to meet consumer expectations for immediacy.
- Governance presents a significant obstacle, as smart contract management demands defining roles, assigning permissions, and enforcing policies among diverse stakeholders, including EVCS operators and regulatory bodies. Coordinating these parties can substantially extend development timelines and complicate deployment.
- In large urban areas where a majority of vehicles transition to EVs, scaling the network to support a surge in nodes and transactions might encounter difficulties due to inherent constraints in Hyperledger Fabric [45, 46]. Grid stability and operational efficiency benefit when EVs charge predominantly at commercial stations rather than residential ones [5], likely leading to policies favoring station-based charging. Rising EV adoption combined with this shift could drive transaction volumes high enough to create performance bottlenecks. Should a high density of EVs degrade network efficiency, future investigations could examine remedies like Channels and Sharding [47], Layer-2 scaling approaches [48], or refined consensus algorithms [49].
- Establishing and sustaining a Hyperledger Fabric network entails upfront infrastructure expenses for node hosting and management. Maintaining on-premises nodes requires constant uptime, raising energy use and ongoing operational expenditures. These added capital and recurring costs could deter certain EVCS operators from joining blockchain ecosystems, slowing broader implementation.
- Merging the proposed smart contract framework with pre-existing legacy systems at EV charging stations creates compatibility issues.

Conclusion

The Hyperledger Fabric-based blockchain framework we propose for electric vehicle charging station networks provides effective defenses against cyber threats through secure, distributed transaction processing. By incorporating smart contracts, it delivers strong safeguards against attacks, including false data injection, denial-of-service, malware injection, and man-in-the-middle exploits, through mechanisms such as role-based access enforcement, cryptographic checks, and consensus-based verification. Benchmarking with Hyperledger Caliper shows the system sustains elevated transaction throughput and low latency across diverse workloads, confirming its scalability and the efficient processing of concurrent peak charging requests. Resource monitoring indicates optimized CPU and memory usage throughout the network, providing durability during intensive operations. The design's capacity to expand with additional nodes and user positions it as a practical option for meeting the escalating needs of EVCS infrastructure, establishing a trustworthy, dependable base for improved confidence and seamless operations in smart city environments.

Subsequent studies should rigorously test the blockchain solution at substantially higher transaction-per-second rates to evaluate its endurance in demanding real-world scenarios. Efforts should also focus on overcoming Hyperledger Fabric's scalability constraints to support escalating EV transaction volumes, especially under policies promoting commercial station charging. Furthermore, work is required to simplify governance structures and lower deployment and maintenance expenses to facilitate faster integration of blockchain technologies into EV charging systems.

Abbreviations

ACL: Access Control List

AES: Advanced Encryption Standard

CP: Charging Provider

CSMS: Charging Station Management System

DoS: Denial-of-Service

EIM: External Identification Means

EP: Energy Provider

EV: Electric Vehicle

EVCS: Electric Vehicle Charging Station

EVSE: Electric Vehicle Supply Equipment

FDI: False Data Injection

HF: Hyperledger Fabric

IEA: International Energy Agency

IIM: Internal Identification Methods

MI: Malware Injection

MITM: Man-In-The-Middle

OCPP: Open Charge Point Protocol

PBFT: Practical Byzantine Fault Tolerance

PLC: Power Line Communication

RBAC: Role-Based Access Control

TPS: Transaction Throughput Test

V2G: Vehicle-to-Grid

Acknowledgements

None

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 23 Mar 2022

Revised: 12 May 2022

Accepted: 10 Jun 2022

Published online: 20 September 2022

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

United Nations Department of Economic & Social Affairs. World Urbanization Prospects: The 2018 Revision—Highlights. 2018. <https://www.un.org/en/desa/2018-revision-world-urbanization-prospects> (accessed on 10 October 2024).

Oliveri LM, D'Urso D, Trapani N, Chiacchio F. Electrifying green logistics: A comparative life cycle assessment of electric and internal combustion engine vehicles. *Energies*. 2023;16:7688.

International Energy Agency. Global EV Outlook 2021. <https://www.iea.org/reports/global-ev-outlook-2021> (accessed on 14 October 2024).

International Energy Agency (IEA). By 2030, EVs Represent More Than 60% of Vehicles Sold Globally and require an Adequate Surge in Chargers Installed in Buildings. 2023. <https://www.iea.org/reports/by-2030-evs-represent-more-than-60-of-vehicles-sold-globally-and-require-an-adequate-surge-in-chargers-installed-in-buildings> (accessed on 10 October 2024).

Powell S, Cezar GV, Min L, Azevedo IM, Rajagopal R. Charging infrastructure access and operation to reduce the grid impacts of deep electric vehicle adoption. *Nat Energy*. 2022;7:932-45.

Buedi ED, Ghorbani AA, Dadkhah S, Ferreira RL. Enhancing EV charging station security using a multi-dimensional dataset: CICEVSE2024. In: Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy; San Jose, CA, USA. Springer; 2024. p. 171-90.

Open Fastcharging Alliance. Autocharge. 2017. <https://github.com/openfastchargingalliance/openfastchargingalliance/blob/master/autocharge-final.pdf> (accessed on 8 October 2024).

Kilic A. Plug and Charge solutions with vehicle-to-grid communication. *Electr. Power Compon. Syst.* 2023; 51:1786-814.

International Organization for Standardization. Road Vehicles—Vehicle to Grid Communication Interface—Part 20: 2nd Generation Network Layer and Application Layer Requirements. 2022. <https://www.iso.org/standard/77845.html> (accessed on 15 January 2025).

Open Charge Alliance. Open Charge Point Protocol (OCPP). 2023. <https://openchargealliance.org/protocols/open-charge-point-protocol/> (accessed on 10 October 2024).

Fu Z, Dong P, Ju Y. An intelligent electric vehicle charging system for new energy companies based on consortium blockchain. *J Clean Prod.* 2020;261:121219.

Pustišek M, Kos A, Sedlar U. Blockchain based autonomous selection of electric vehicle charging station. In: *Proceedings of the 2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*; Beijing, China. 2016. p. 217-22.

Knirsch F, Unterweger A, Engel D. Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions. *Comput Sci Res Dev.* 2018;33:71-9.

Jeong S, Dao NN, Lee Y, Lee C, Cho S. Blockchain based billing system for electric vehicle and charging station. In: *Proceedings of the 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*; Prague, Czech Republic. 2018. p. 308-10.

Lin YJ, Chen YC, Zheng JY, Shao DW, Chu D, Yang HT. Blockchain-based intelligent charging station management system platform. *IEEE Access.* 2022;10:101936-56.

Huang Z, Li Z, Lai CS, Zhao Z, Wu X, Li X, et al. A novel power market mechanism based on blockchain for electric vehicle charging stations. *Electronics.* 2021;10:307.

Dorokhova M, Vianin J, Alder JM, Ballif C, Wyrsh N, Wannier D. A blockchain-supported framework for charging management of electric vehicles. *Energies.* 2021;14:7144.

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 Commun Surv Tutor.* 2022;24:1504-33.

Baker R, Martinovic I. Losing the car keys: Wireless PHY-layer insecurity in EV charging. In: *Proceedings of the 28th USENIX Security Symposium (USENIX Security 19)*; Santa Clara, CA, USA. 2016. p. 407-24.

Alcaraz C, Lopez J, Wolthusen SD. OCPP protocol: Security threats and challenges. *IEEE Trans Smart Grid.* 2017;8:2452-9.

Metere R, Neaimeh M, Morisset C, Maple C, Bellekens X, Czekster RM. Securing the electric vehicle charging infrastructure. *arXiv.* 2021;arXiv:2105.02905.

Kim Y, Hakak S, Ghorbani A. Detecting Distributed Denial-of-Service (DDoS) attacks that generate false authentications on electric vehicle charging infrastructure. *Comput Secur.* 2024;144:103989.

Chung YW. Electric Vehicle—Smart Grid Integration: Load Modeling, Scheduling, and Cyber Security [dissertation]. Los Angeles (CA): University of California, Los Angeles; 2020.

Korba AA, Tamani N, Ghamri-Doudane Y. Anomaly-based framework for detecting power overloading cyberattacks in smart grid AMI. *Comput Secur.* 2020;96:101896.

Ping J, Yan Z, Chen S, Yao L, Qian M. Coordinating EV charging via blockchain. *J Mod Power Syst Clean Energy.* 2020;8:573-81.

Kirimtat A, Krejcar O, Kertesz A, Tasgetiren MF. Future trends and current state of smart city concepts: A survey. *IEEE Access*. 2020;8:86448–67.

Ahmad K, Maabreh M, Ghaly M, Khan K, Qadir J, Al-Fuqaha A. Developing future human-centered smart cities: Critical analysis of smart city security, data management, and ethical challenges. *Comput Sci Rev*. 2022;43:100452.

Singh S, Sharma PK, Yoon B, Shojafar M, Cho GH, Ra IH. Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city. *Sustain Cities Soc*. 2020;63:102364.

Majeed U, Khan LU, Yaqoob I, Kazmi SA, Salah K, Hong CS. Blockchain for IoT-based smart cities: Recent advances, requirements, and future challenges. *J Netw Comput Appl*. 2021;181:103007.

El Bekkali A, Essaaidi M, Boulmalf M. A blockchain-based architecture and framework for cybersecure smart cities. *IEEE Access*. 2023;11:76359–70.

Thakkar P, Nathan S, Viswanathan B. Performance benchmarking and optimizing Hyperledger Fabric blockchain platform. In: *Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. Milwaukee (WI), USA; 2018:264–76.

de Ocariz Borde HS. An overview of trees in blockchain technology: Merkle trees and Merkle Patricia tries [dissertation]. Cambridge (UK): University of Cambridge; 2022.

Barger A, Manevich Y, Meir H, Tock Y. A Byzantine fault-tolerant consensus library for Hyperledger Fabric. In: *Proceedings of the 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. Sydney (Australia); 2021. p.1–9.

Huang D, Ma X, Zhang S. Performance analysis of the raft consensus algorithm for private blockchains. *IEEE Trans Syst Man Cybern Syst*. 2019;50:172–81.

Li C, Qiu W, Li X, Liu C, Zheng Z. A dynamic adaptive framework for practical Byzantine fault tolerance consensus protocol in the Internet of Things. *IEEE Trans Comput*. 2024;73:1669–82.

Capocasale V, Gotta D, Perboli G. Comparative analysis of permissioned blockchain frameworks for industrial applications. *Blockchain Res Appl*. 2023;4:100113.

Mittal N, Pal S, Joshi A, Sharma A, Tayal S, Sharma Y. Comparative analysis of various platforms of blockchain. In: *Smart and Sustainable Intelligent Systems*. Hoboken (NJ): Wiley; 2021:323–40.

Kuo TT, Zavaleta Rojas H, Ohno-Machado L. Comparison of blockchain platforms: A systematic review and healthcare examples. *J Am Med Inform Assoc*. 2019;26:462–78.

Riahi K, Brahmia MEA, Abouaissa A, Idoumghar L. A comparative study of blockchain development platforms. In: *Proceedings of the 9th International Conference on Communication and Information Processing*. Hainan (China); 2023. p. 555–61.

Atlam HF, Ekuri N, Azad MA, Lallie HS. Blockchain forensics: A systematic literature review of techniques, applications, challenges, and future directions. *Electronics*. 2024;13:3568.

ElGayyar MM, ElYamany HF, Grolinger K, Capretz MA, Mir S. Blockchain-based federated identity and auditing. *Int J Blockchains Cryptocurrencies*. 2020;1:179–205.

Soud M, Liebel G, Hamdaqa M. PrAloritize: automated early prediction and prioritization of vulnerabilities in smart contracts. *arXiv*. 2023:2308.11082.

Gupta B, Mittal P, Mufti T. A review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) services. In: Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development (ICIDSSD). New Delhi (India); 2020.

Services A. AWS Pricing Calculator. <https://calculator.aws/#/> (accessed on 20 January 2025).

Wang C, Chu X. Performance characterization and bottleneck analysis of Hyperledger Fabric. In: Proceedings of the 40th IEEE International Conference on Distributed Computing Systems (ICDCS). Singapore; 2020. p. 1281–6.

Nguyen MQ, Loghin D, Dinh TTA. Understanding the scalability of Hyperledger Fabric. arXiv. 2021:2107.09886.

Guggenberger T, Sedlmeir J, Fridgen G, Luckow A. An in-depth investigation of the performance characteristics of Hyperledger Fabric. *Comput Ind Eng*. 2022;173:108716.

Rebello GAF, Camilo GF, de Souza LAC, Potop-Butucaru M, de Amorim MD, Campista MEM, et al. A survey on blockchain scalability: From hardware to layer-two protocols. *IEEE Commun Surv Tutor*. 2024;26:2411–58.

Wu-Tong M, Da-Wei Z. Optimization scheme for Hyperledger Fabric consensus mechanism. *Acta Autom Sin*. 2021;47:1885–98.