

Deposit-Escrow EV Charging on Avalanche with zk-SNARK Billing Proofs: Prototype Scope and Security Limits

Mashaël Asiri

Technical and Engineering Unit, Applied College at Mahayil, King Khalid University, Saudi Arabia.

Email: abusharara@kku.edu.sa

Received: 13 June 2026 **Accepted:** 10 September 2026 **Published:** 21 October 2026

DOI: 10.15849/ijasca.206 **License:** Open access under CC BY 4.0. Authors retain copyright.

Publisher: Al-Zaytoonah University of Jordan (ZUJ), Amman, Jordan.

Cite this article: M. Asiri, "Deposit-escrow EV charging on Avalanche with zk-SNARK billing proofs: prototype scope and security limits," *Int. J. Advance Soft Compu. Appl.*, vol. 18, no. 3, pp. 357–373, 2026, doi: <https://doi.org/10.15849/ijasca.206>.

ABSTRACT

Electric-vehicle (EV) charging workflows expose billing and session information through centralized services or transparent ledgers. This study integrates deposit escrow, a Groth16 billing-arithmetic proof, and settlement on the Avalanche Fuji testnet. Its contribution is a proof-of-concept workflow linking authorization, payment, refund, and timeout handling, together with an analysis of the boundary between the implemented proof and a proposed station-authenticated protocol. Across five runs per setting, mean deposit and settlement confirmation times are 1.8 s and 2.1 s, respectively; proof generation averages 0.21 s on a desktop PC; and settlement consumes approximately 320,000 gas. These descriptive results apply to a simplified circuit that checks billing arithmetic with a placeholder binding. Station signatures are checked only by the off-chain harness, so station-authenticated billing and resistance to substituted energy witnesses remain unvalidated. Normal settlement provides meter-field omission, not consumption confidentiality: under the exact public linear tariff, energy E is recoverable from public payment C and positive tariff p as $E = C/p$. The results support feasibility of the escrow-and-arithmetic-proof workflow; authenticated witness binding, complete reproducibility artifacts, and stronger experimental validation remain prerequisites for a secure deployment.

Keywords: Electric vehicles; blockchain; Avalanche; deposit escrow; zero-knowledge proofs; billing arithmetic; privacy limitations.

1 Introduction

Electric-vehicle (EV) charging workflows routinely handle charging location and time, payment amounts, and energy-consumption information. Repeated session records can support behavioral profiling when charging operators or service providers can associate them with a vehicle. These concerns motivate protocols that specify not only what is omitted from a message, but also what an observer can infer from the information that remains public [1].

Blockchain-based charging systems seek to provide tamper-evident records, distributed coordination, and automated settlement. Prior studies address different layers, including security at the Internet-of-Energy/EV interface [2], charging-pile management [3], and privacy-oriented charging with fog computing [4]. These objectives should not be conflated with confidentiality of the final bill: integrity of a ledger and pseudonymous wallet addresses do not, by themselves, determine what consumption information an observer can recover.

Fully homomorphic encryption (FHE) and zero-knowledge proofs address different requirements. FHE supports computation on encrypted inputs [5], whereas a succinct zero-knowledge proof allows a verifier to check an encoded relation without receiving its private witness [6]. Their cost and privacy properties depend on the complete computation, trust model, and disclosure pattern. They therefore provide architectural context rather than security-equivalent performance baselines for this study.

We study a deposit-based workflow on the EVM-compatible Avalanche C-Chain [7]. A user locks a refundable deposit for a station, receives a metered session summary off-chain, and submits a settlement amount with a Groth16 proof. The contract checks the implemented proof relation and settles the escrow. The proposed complete protocol additionally requires station authentication of the proof witness; this requirement is not enforced by the evaluated circuit.

Privacy goals and limitations. The station necessarily observes the energy that it delivers. On the public ledger, the normal settlement omits E as an explicit transaction parameter, but discloses C , the tariff, station and account identifiers, deposits, and timing information. For the evaluated linear tariff with $p > 0$ and exact arithmetic, $C = E \cdot p$ uniquely determines E . We therefore use “meter-field omission” for the implemented disclosure property and reserve “consumption confidentiality” for a stronger property that this prototype does not achieve. No fine-grained time-series consumption profile or credential-privacy guarantee is evaluated.

Research question and novelty. The study asks how a deposit-escrow lifecycle can be integrated with a wallet-side billing-arithmetic proof and Avalanche settlement, and what this integration actually verifies and discloses. Its contribution is an application-level integration and evaluation, not a new cryptographic primitive or a complete confidential-billing protocol. Relative to the authentication, fog-computing, aggregation, and encrypted-billing objectives in related work [1, 4, 10, 14], the focus is a single-session escrow workflow with explicit separation of implemented functionality, target security requirements, and observable information.

Implemented workflow. The prototype connects deposit authorization, billing-arithmetic proof generation, on-chain verification, payment, refund, and timeout handling. The evaluation characterizes this specific integration rather than claiming the first use of escrow, blockchain, or zero-knowledge proofs in EV charging.

Protocol and disclosure analysis. We separate the implemented arithmetic relation from the proposed station-authenticated relation, explain the unresolved substituted-energy case, and derive the consumption information disclosed by public linear pricing.

Descriptive implementation evidence. Five-run timing summaries and approximate settlement gas quantify the simplified Fuji workflow. The plaintext comparison isolates an implementation-level gas difference without assuming equivalent authentication or confidentiality.

Paper organization. Section 2 reviews related work on blockchain-based EV charging and privacy-preserving billing. Section 3 presents the system model, threat assumptions, and protocol design. Section 4 describes the prototype implementation, and Section 5 reports experimental results. Section 6 discusses deployment considerations and limitations, and Section 7 concludes the paper.

2 Related Work

Blockchain applications in EV systems span infrastructure security, access control, payment automation, and privacy. We organize the literature by the property each approach addresses. Comparisons below are architectural unless explicitly identified as measurements of our own prototype; no cross-paper latency or throughput ranking is inferred from different experimental settings.

2.1 Blockchain for EV Charging Networks

Saha et al. study security at the Internet-of-Energy/EV interface using blockchain and lattice-based mechanisms [2]. Liu et al. examine blockchain-based management of charging piles in an electricity Internet-of-Things setting [3]. These approaches provide context for infrastructure security and coordination. The present study instead examines the relationship between escrow settlement and the statement enforced by a billing-arithmetic proof.

The Share&Charge project report provides historical context for practical blockchain-based charging development [8]. It is not used as technical or performance evidence. Avalanche is selected for its EVM execution environment and compatibility with Solidity contracts [7]; no head-to-head platform benchmark is performed.

Li, Han, and Tang combine blockchain and fog computing in a privacy-preserving charging scheme [4]. Their placement of computation at the fog layer differs from the wallet-side proof generation and on-chain verification studied here. Comparing trust, complexity, or confidentiality would require matched threat models and implementations.

2.2 Privacy-Preserving EV Charging Protocols

Zhai et al.'s EPDB scheme uses decentralized identifiers and Pedersen commitments in a privacy-oriented electric-charging design [9]. Chen et al. address privacy-preserving authentication for EV battery swapping [10]. These works illustrate why authentication, identity-related privacy, and billing confidentiality should be assessed separately, with the last evaluated against information visible at settlement.

Lauser et al.'s Oblivious Plug&Charge uses oblivious RAM (ORAM) to aggregate consumption while limiting the association of individual charging sessions with vehicles [1]. Its aggregation mechanism and service-provider/operator disclosure model differ from a ledger that publishes a payment for every session.

Shafin and Reno investigate a hybrid blockchain payment system for charging on electrified roads [11]. Their work is relevant to charging-payment coordination, but its simulation and system assumptions do not provide a matched baseline for the Fuji measurements in this study.

The proposed complete protocol requires an authenticated commitment or signature over the same session and energy value used by the proof. A prover-controlled equality cannot establish that provenance. Section 6.2 discusses possible constructions; neither is implemented by the prototype's placeholder binding.

2.3 EV Energy Trading and Payment Systems

Khan and Byun study blockchain-based peer-to-peer energy trading and charging payments [12]. Sharma, Singh, and Lin consider secure strategies for a heterogeneous networked hub of EV charging stations [13]. These market and payment-coordination objectives complement the present focus on single-session escrow and verifiable billing arithmetic, but do not provide directly comparable computation or disclosure benchmarks.

The evaluated circuit is specialized to $C = E \cdot p$. The measured gas and proving time characterize that arithmetic task rather than a complete station-authenticated billing construction.

2.4 Comparison with FHE-Based Approaches

Aldweesh and Alangari describe EV charging authorization and billing using blockchain and homomorphic encryption [14], an approach distinct from the ORAM-based aggregation of Lauser et al. [1]. These studies provide architectural comparisons. A quantitative comparison would require common functionality, threat assumptions, hardware, and measurement boundaries; no matched FHE implementation is evaluated here.

Table 1 separates the objectives and available evidence for the billing approaches. A proof does not hide a public transfer amount, and encrypted computation alone does not specify what settlement reveals. Privacy comparisons therefore require the complete observer view, including outputs and metadata, rather than a single privacy ranking.

Table 1. Scope-aware comparison of billing approaches; not a performance ranking.

Dimension	Plaintext settlement baseline	Homomorphic-encryption approach	This prototype
Role in this paper	Locally implemented gas baseline.	Related-work context [14]; no matched implementation.	Measured escrow and arithmetic-proof workflow.
Computation	Meter value supplied directly for settlement.	Encrypted computation; disclosure depends on the complete protocol.	$C = E \cdot p$ with a placeholder binding.
Consumption disclosure	E is an explicit public input.	Requires protocol-specific assessment of outputs, keys, and recipients.	No explicit E field, but public C and p recover E exactly in the evaluated tariff.

Dimension	Plaintext settlement baseline	Homomorphic-encryption approach	This prototype
Station authentication	No equivalence claim beyond the reported baseline.	Not evaluated in this study.	Signature checked only in the off-chain harness; no end-to-end proof binding.
Quantitative evidence	Approx. 50,000 gas for settlement.	No comparable timing, throughput, or energy measurements here.	Five-run summaries; approx. 320,000 gas for settlement.
Deployment status	Reference implementation for comparison.	Outside this study's deployment evaluation.	Proof of concept, not production-validated.

The station-authenticated relation is a target specification, not a measured implementation in Table 1.

A fixed artificial delay does not constitute a benchmark for homomorphic-encryption computation. Such a delay captures neither cryptographic parameters nor computation, communication, or key-management costs and is excluded from the comparative evaluation.

Batching and aggregation may reduce per-session ledger exposure or amortize verification costs, but they are not implemented here. Their evaluation would require a specified protocol rather than an extrapolation from individual proof verification.

2.5 Other Related Technologies

Related work also considers driver-behavior-based EV insurance [15], federated learning for malicious-node detection in vehicular networks [16], and blockchain-based firmware updates for charging stations [17]. These approaches address complementary risks and are not experimentally integrated into this prototype.

In summary, this study contributes a bounded integration of deposit escrow and a billing-arithmetic proof. Its distinct focus is the relationship between settlement execution, the implemented proof statement, and the information exposed by the bill; comparative security or performance superiority is not established.

3 System Architecture and Protocols

This section separates the proposed complete protocol from the implementation evaluated in Section 4. Requirements involving verifier-enforced station attestation are not implemented properties. A private witness describes the proof interface, not necessarily information hidden from an observer of its public outputs.

3.1 Architecture Overview

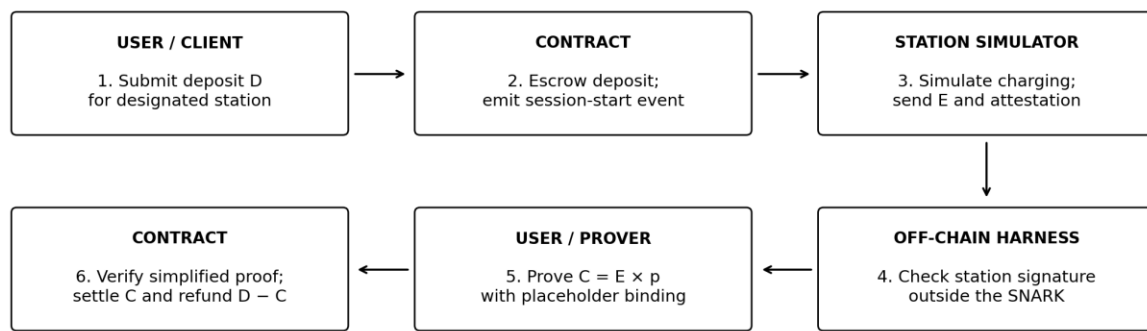
Figure 1 summarizes the prototype workflow and its unresolved authentication boundary. The system includes:

EV user U (driver). The user communicates with a station and a blockchain wallet. The prototype client generates the proof on a desktop PC; proof generation on a phone or vehicle unit is a deployment objective, not a measured capability.

Charging station S. In the system model, the station dispenses energy, measures the delivered amount, and holds an on-chain payout address and a signing key for session attestations. The evaluation represents this role with a station simulator rather than a physical charger.

Avalanche smart contract SC. The contract on Avalanche C-Chain maintains session state, escrows deposits, verifies proofs, and atomically settles payments.

Optional credential issuer I. The proposed complete protocol permits membership credentials and an authorization proof. The evaluation uses deposit-only authorization; credential privacy, revocation behavior, and membership-mode performance are outside its scope.



MISSING ENFORCEMENT: the verified proof does not authenticate E to the station.

PUBLIC DISCLOSURE: C and p reveal $E = C/p$ for the evaluated exact linear tariff.

Figure 1. Evaluated workflow and trust boundary. Station attestation is checked in the off-chain harness. The contract verifies the billing-arithmetic proof, not an authenticated link between its energy witness and the station's signed reading. The separate timeout path discloses E explicitly. Source: This study.

Communication channels. The protocol uses a local user–station channel for negotiation and delivery of the attestation, and a public blockchain for deposits and settlement. The prototype emulates local communication through a file. Secure transport is a deployment requirement; the study does not evaluate local-channel confidentiality, remote authentication, or network-level traffic analysis.

Metering assumption. The station observes E. Its signature can authenticate a stated reading, but cannot establish the physical accuracy of a dishonest or compromised meter. Ledger-observer disclosure is analyzed separately in Section 5.6.

Table 2. Notation used in the protocol.

Symbol	Meaning
SC	Smart contract on the Avalanche C-Chain; distinguished from cost C.
U, S, sid	User, designated charging station, and session identifier.
D	Escrowed deposit in the chosen payment token.
E, Emax	Metered energy in integer units and a specified maximum energy bound.
p, τ	Positive integer-scaled tariff and session-bound tariff identifier.
C	Public settlement cost; $C = E \cdot p$ for the evaluated linear tariff.
σS , pkS	Station attestation and the station's authenticated public key.
π , VK	Groth16 proof and the fixed verification key for the deployed circuit.
domain	Proposed chain/contract/protocol context for replay-resistant attestations.

Adversaries. We consider passive ledger observers seeking consumption or session linkage; malicious users seeking unpaid energy, underbilling, or session griefing; malicious stations seeking overcharging or deanonymization; and network adversaries delaying transactions or observing traffic. The limited functional checks in Section 5 do not demonstrate security against every adversary in this model.

Target properties and their status. Escrow accounting should pay the accepted amount once and refund the remainder. Station-authenticated billing additionally requires binding the amount to the designated station's session reading and tariff. The functional tests cover accounting and the deployed arithmetic relation, not that binding. Section 5.6 assesses consumption confidentiality and the limits of address rotation.

Conditional fairness. A deposit reserves at most D ; it does not establish how much energy was delivered or automatically compensate a station for delivery beyond that amount. A deployment must stop charging, request a top-up, or apply another explicit rule before the payable amount exceeds the deposit. Fairness further depends on metering integrity, the attestation mechanism, transaction availability, and a dispute policy for dishonest stations.

3.2 Protocol Phases

We describe the protocol in four phases. Table 2 summarizes key notation.

3.2.1 Authorization via Deposit and Optional Credentials

The user opens an escrow for a designated station. This demonstrates availability of the deposited amount, not a guarantee that the eventual bill is based on an authenticated reading.

$$\text{sid} \leftarrow \text{SC.startSession}(S, D)$$

The contract records an Active session and emits SessionStart after the deposit is accepted. The station begins only after observing a matching session. In the prototype, payment uses an ERC-20 test token, so funding and approval requirements belong to the token-transfer workflow rather than a native-AVAX value-bearing call.

In optional membership mode, the intended interface also accepts π_{auth} . Because that path is not evaluated, the protocol description does not establish its credential scheme, issuer trust, revocation policy, or proof cost. It is excluded from the reported authorization timings.

Session binding. The contract record binds sid to the depositor, station, deposit, and start timestamp. The proposed complete protocol must also bind the tariff and station key to that session and compare proof public inputs with stored values. Caller checks alone do not bind the energy witness to the session.

3.2.2 Charging and Meter Attestation

While charging, the station meters delivered energy and prepares an end-of-session attestation. The prototype's off-chain attestation is described as signing a digest of sid , E , and metadata. The proposed complete protocol requires an unambiguous message that also binds the station, user, and tariff:

$$m = H(\text{domain} \parallel \text{sid} \parallel U \parallel S \parallel \tau \parallel E), \quad \sigma_S = \text{Sign}_{\text{sk}_S}(m)$$

Here domain denotes a chain identifier, contract address, and protocol version, and τ identifies the fixed tariff for the session. This message is a target encoding, not the attestation format established for the prototype. All parties must use a canonical encoding, and the contract must authenticate the station key. The station sends the reading and attestation over the off-chain channel.

Enforcement boundary. An attestation prevents witness substitution only when settlement enforces use of the same station-endorsed reading. Checking it in an honest client or test harness does not protect against a user who modifies that software.

3.2.3 Settlement via On-Chain Proof Verification

For a linear tariff, the user computes the settlement amount:

$$C = E \cdot p, \quad p > 0$$

The proposed complete proof establishes knowledge of E and σ_S such that the station signature is valid for the session and the fixed tariff produces C . The evaluated proof checks only billing arithmetic with a placeholder binding. The user submits:

$$\text{SC.endSession}(\text{sid}, C, \pi)$$

If the deployed verifier accepts the proof and the contract checks $C \leq D$, the workflow transfers C to the station, refunds $D - C$ to the user, and marks the session Settled. This accounts for the accepted amount; it does not authenticate the energy used to calculate it.

Inputs and numerical representation. The proposed complete protocol must match its public statement to the stored session, station key, tariff, deposit, and settlement amount. E and σ_S can remain private witnesses. Section 3.3.1 specifies the numerical requirements for interpreting the relation as integer billing rather than finite-field arithmetic.

3.2.4 Timeout and Forced Settlement

$SC.forceEndSession(sid, E, \sigma_S)$

The proposed fallback allows the designated station to claim after a deadline, with signature verification outside the normal billing SNARK and plaintext settlement inputs supplied on-chain. The prototype reports a functional check of this path. Its liveness is conditional on the station possessing the required data and transactions being included; it is not an unconditional guarantee of fair payment against a malicious station or compromised meter.

The fallback explicitly discloses E and requires a defined outcome when the computed amount exceeds D , when no charging occurred, or when a reading is disputed. These policies and their edge-case coverage are not established by the normal-path measurements. A later timeout also does not repair an underbilled session that has already been accepted and closed.

3.3 Zero-Knowledge Statements and Circuit Design

Let VK denote the on-chain verification key for the compiled relation. A SNARK proves satisfaction of the relation encoded by that circuit; it does not supply predicates omitted from the circuit [6]. The target relation and the implemented scope are therefore specified separately.

Target station-authenticated relation—not implemented in the evaluated circuit. For session-bound public context $x = (\text{domain}, sid, U, S, pk_S, \tau, p, C, D, E_{\max})$, the proposed relation is:

$$\begin{aligned} \exists E, \sigma_S : \text{VerifySig}(pk_S, H(\text{domain} \parallel sid \parallel U \parallel S \parallel \tau \parallel E), \sigma_S) = 1 \\ \wedge C = E \cdot p \wedge 0 \leq E \leq E_{\max} \wedge 0 \leq C \leq D. \end{aligned} \quad (1)$$

The station key must be authenticated, and p , τ , and D must agree with contract state rather than be freely selectable proof inputs. Binding all public inputs to that state is part of the acceptance condition. Equation (1) also assumes a secure signature scheme, suitable setup and proof-system assumptions, and trustworthy metering; it is not a proof that a malicious station's signed measurement is physically accurate.

Implemented scope. The evaluated circuit enforces $C = E \cdot p$ with a lightweight placeholder equality and checks described as basic sanity checks. It does not verify a station signature, and no independently authenticated on-chain commitment is established for the witness. It therefore does not implement Equation (1). The exact compiled constraints and signal declarations are unavailable for independent inspection.

Why the missing predicate matters. A modified client may select E' different from the attested E , calculate $C' = p \cdot E'$, and recompute any prover-controlled placeholder values. The arithmetic relation may then hold without a valid association with the attested reading. This is a conditional analysis of the enforcement gap, not an observed exploit. Section 5.5 specifies the unresolved adversarial test.

3.3.1 Numerical Representation and Finite-Field Safety

Representation status. The evaluated relation is described in integer-scaled application units, but the energy unit, tariff scale, token decimals, permitted ranges, $E_{\max}$, bit widths, and implemented range constraints are not available. The following symbolic requirements define the intended exact-arithmetic interpretation; they are not a recovered configuration or evidence that the compiled prototype enforces them.

Scaling requirements. Let e be physical energy in kWh, s_E the number of integer energy units per kWh, t the tariff in tokens/kWh, and d the token decimals. One consistent target encoding sets $E = s_E \cdot e$,

expresses C and D in the smallest token units, and represents p as the number of smallest token units per energy unit. Exact billing then requires:

$$E = s_E \cdot e, \quad p = (10^d \cdot t) / s_E, \quad C = E \cdot p.$$

E and p must be integers under this encoding, with $p > 0$. If the selected resolution and tariff do not permit exact integer representation, the protocol must define a different scale or an explicit rounding relation and analyze that relation separately. Neither d nor s_E can be inferred from the illustrative tariff of 0.1 token/kWh.

Range and wraparound requirements. Let r be the scalar-field modulus used by the compiled circuit. A sufficient target policy constrains E , p , C , and D to non-negative integers with $0 \leq E \leq E_{\max}$, $1 \leq p \leq p_{\max}$, and $0 \leq C \leq D \leq D_{\max}$, where all maxima are below r and $E_{\max} \cdot p_{\max} < r$. Then $C = E \cdot p$ in the field implies the same equality over the permitted integers, because neither side can wrap modulo r .

Bit widths and enforcement. For each represented quantity X , its selected maximum $X_{\max}$ determines a minimum width $b_X = \text{ceil}(\log_2(X_{\max} + 1))$; for accepted bills, a target bound is $C_{\max} = \min(E_{\max} \cdot p_{\max}, D_{\max})$. A complete implementation must enforce a unique integer encoding, range constraints, and comparisons with the selected maxima, and must bind the public tariff and deposit to contract state. Witness-generation checks alone do not enforce circuit soundness. The actual values of b_E , b_p , b_C , b_D , $E_{\max}$, $p_{\max}$, and $D_{\max}$, together with the corresponding compiled constraints, remain unavailable for this prototype.

3.4 Smart Contract State and Interface

The contract maintains a mapping from session identifiers to session records:

$$\text{Sess}[\text{sid}] = (\text{user}, \text{station}, D, t_0, \text{state})$$

Core functions. `startSession` creates an active escrow; `endSession` verifies the deployed proof and performs payment and refund; `forceEndSession` provides the designated station's timeout path. The proposed complete protocol additionally requires an authenticated station-key registry and session-bound tariff context, or equivalent state.

Abuse resistance. One-time settlement and caller restrictions address replay of closed sessions and third-party access. They do not authenticate an arbitrary prover-supplied reading. Reentrancy protection, token-transfer failure handling, deposit caps, no-charge cancellation, and dispute rules require explicit implementation and tests. We do not treat the limited lifecycle tests as a full smart-contract security audit.

4 Implementation and Experimental Setup

The prototype connects deposit authorization, billing-arithmetic proof generation, on-chain verification, and settlement on Fuji. The following description identifies the implemented components, measurement boundaries, and information needed to reproduce the experiments. It does not treat workflow completion as proof of station-authenticated billing.

4.1 Prototype Components

The prototype consists of four main components:

Smart contract. The session manager and Groth16 verifier were implemented in Solidity and deployed to Avalanche Fuji using Hardhat. They support deposit, session-state, and settlement operations. No independent audit or production deployment is included in the evaluation.

Zero-knowledge circuits. The prototype uses Circom circuits compiled to a rank-1 constraint system (R1CS) and snarkJS to generate Groth16 keys and proofs [18]. Although an optional authorization circuit is described, the experiments use deposit-only authorization and evaluate only the billing-arithmetic circuit. The verification key is fixed in the Solidity verifier.

User client (prover). A lightweight Python client (using web3.py) orchestrates the user workflow: starting a session, receiving the station's end-of-session attestation, generating the billing proof π , and submitting endSession with (sid, C, π).

Station simulator. A Node.js process emulates a charging station by monitoring contract events. When a SessionStart event is observed for its station identifier, it simulates a charging interval and then produces an end-of-session attestation for the user. In our test harness, "local communication" is simulated by writing the attestation to a file read by the user client.

4.2 Smart Contract Design (Prototype)

The contract exposes the interfaces described in Section 3: startSession, endSession, and a timeout path forceEndSession. For each session, the contract records at minimum the depositor address, station identifier, deposit amount, and a start timestamp. The contract enforces a monotonic session state machine to prevent double settlement and to ensure that each deposit can be consumed at most once.

Asset used for deposits and settlement. Deposits and payments use a local ERC-20 test token; Fuji AVAX is used for transaction fees. This separates the experimental payment asset from the fee asset. Adapting the escrow to native AVAX would require value-bearing calls and refunds and is not part of the measured implementation.

Timeout path. A functional check closes an unpaid session through forceEndSession using plaintext settlement input. Separate latency and gas distributions, adverse-metering tests, and deposit-overflow coverage are unavailable for this path; normal-path endSession metrics do not characterize it.

4.3 Zero-Knowledge Proof Implementation

Proving system and curve. Billing-arithmetic proofs use Groth16 over BN254 with a fixed verification key. Groth16 is a preprocessing proof system [6]. The snarkJS workflow supports a Powers of Tau phase and circuit-specific setup, including contribution and verification operations [18]. The prototype's setup transcript, contribution history, and artifact identifiers are unavailable; the use of snarkJS alone does not establish how its trusted setup was conducted.

Billing circuit and signals. E is described as the private energy witness, while C and p are public billing values. The compiled public/private signal counts, declaration order, outputs, auxiliary witnesses, and the treatment of p as a signal or circuit constant are not specified. In particular, these conceptual roles must not be mistaken for a verified public-signal layout. The 0.21 s mean applies to the simplified billing-arithmetic circuit, not to station-signature verification.

Station attestation. The test harness verifies the station signature off-chain. The circuit's placeholder is described as a hash-based equality, but the reference value is not established as independently authenticated by the contract. It is therefore not a substitute for the station-attestation predicate in Equation (1).

Circuit and key artifacts. The constraint count, public/private signal counts, proving-key size, verification-key contents and hash, and R1CS and witness-generator identifiers are unavailable. These values cannot be inferred from the billing equation or timing summaries. A reproducibility release must associate the exact circuit with its setup artifacts and deployed verifier, including the setup procedure, contribution model, verification logs, and file hashes. Section 4.6 consolidates the remaining release requirements.

4.4 Experimental Setup

Network. All contract deployments and measurements were performed on the Avalanche Fuji testnet, accessed via a public RPC endpoint. We treat a transaction as confirmed when the receipt is returned successfully and the corresponding contract event (SessionStart or SessionEnd) is observed by the test harness.

Hardware and software environment. Client and station components ran on a PC with an Intel Core i7-9700 CPU at 3.0 GHz and 16 GB RAM. Proof generation used snarkJS. The operating system and exact versions of Circom, snarkJS, Solidity, Hardhat, web3.py, Python, and Node.js are unavailable, as are dependency lockfiles and compiler optimization settings. These desktop measurements do not establish smartphone or vehicle-unit performance.

Parameterization. The experiment uses a flat tariff and samples E within a range intended to keep honest-session C within D. The value 0.1 token/kWh is an illustrative tariff, not a fully documented run configuration. Exact deposits, sampled energy values and distribution, tariff scaling, token decimals, Emax, and any random seed remain unspecified. Consequently, the summaries do not define a reproducible parameter sweep.

4.5 Measurement Methodology and Baselines

Timing boundaries. Deposit latency starts at startSession submission and ends when the harness has observed a successful receipt and the matching SessionStart event. Settlement latency uses the corresponding endSession and SessionEnd boundaries. These application-observed intervals include RPC and event-observation delays; they are not measurements of consensus finality. Proof generation is reported as client-side wall-clock time. Inclusion of witness computation, key loading, and process startup is unspecified, as are the exact timestamp placement, timer implementation, and RPC polling configuration.

Gas. endSession gas is obtained from transaction receipts and covers the complete settlement transaction, including proof verification, state updates, and transfers. No execution trace isolates verifier-only gas. Gas is independent of an AVAX-to-fiat conversion.

Plaintext baseline. The baseline submits E directly and omits the SNARK. Its approximately 50,000-gas settlement is an implementation-level reference, not a security-equivalent comparator: the two paths differ in verification work and data representation, while the proof-bearing path lacks station-authenticated witness binding and reveals E through public linear pricing. The available baseline description does not establish matched attestation checks or identical transfer and state-update logic. Accordingly, the gas difference cannot be interpreted as the isolated price of confidentiality, authentication, or SNARK verification. No matched baseline latency distribution or FHE implementation is available.

Repetitions and uncertainty. Five runs per setting underlie the latency and proof-time summaries. Means and standard deviations describe this small sample only. Individual observations, experimental dates and duration, warm-up treatment, failed-run handling, and the sample-versus-population standard-deviation convention are unavailable. The summaries therefore do not support tail-latency, reliability, statistical-superiority, or load-stability conclusions. Additional repetitions and the per-run records are needed to assess variability independently.

Throughput status. No controlled completed-session throughput measurement is available. The unvalidated submission estimate in Section 5.4 is excluded from the performance summary and is not interpreted as system or network capacity.

4.6 Reproducibility and Artifact Availability

A complete replication package is not available with this study. Reproduction requires the source-code revision; contract, verifier, and token addresses; deployment transactions; token decimals; circuit source, R1CS statistics, and ordered public-signal mapping; setup and key artifacts; dependency versions; compiler flags; and the exact experimental configuration. The circuit must be checked against the scaling and range requirements in Section 3.3.1 rather than assuming that those requirements were implemented. Without these materials, the timing summaries and compiled security boundary cannot be independently reproduced.

Per-run evidence. A replication release should provide run and session identifiers, parameter values, transaction hashes and receipts, submission/receipt/event timestamps, proof-stage timings, gas used, and

success or failure outcomes. Experimental dates, warm-up exclusion, timeout and retry rules, background conditions, and the statistical summary script must accompany those records. The aggregate summaries do not identify the original individual observations.

5 Results and Evaluation

The results characterize the simplified Fuji prototype using five-run timing summaries, approximate settlement gas, and functional checks. Charging is simulated and local communication is file-based. The measurements therefore concern the implemented software workflow rather than physical charger integration or production performance.

5.1 Key Metrics

Table 3 summarizes the available timing and gas measurements. Proof generation covers billing arithmetic and the placeholder binding only. Station energy use and user-device energy consumption were not measured.

Table 3. Reported descriptive performance of the simplified Fuji prototype.

Metric	Reported value	Interpretation and boundary
Deposit confirmation latency	1.8 s (SD 0.3 s)	Five runs; submission to receipt/event observation.
Billing proof generation	0.21 s (SD 0.05 s)	Five runs on the stated desktop PC; arithmetic plus placeholder only.
Settlement confirmation latency	2.1 s (SD 0.4 s)	Five runs; endSession submission to receipt/event observation.
endSession gas	Approx. 320,000	Full proof-bearing settlement transaction; dispersion not reported.
Plaintext-baseline settlement gas	Approx. 50,000	No SNARK; not a security-equivalent baseline; dispersion not reported.

SD denotes the reported standard deviation; the sample-versus-population convention is unspecified. Individual measurements are unavailable for independent recomputation. Throughput and device-energy consumption are not measured quantities in this table.

5.2 Latency

Measured delays. Mean confirmation times are 1.8 s for startSession and 2.1 s for endSession. Figure 2 presents these application-level observations with the reported standard deviations; it does not compare blockchain confirmation with cryptographic computation in another system.

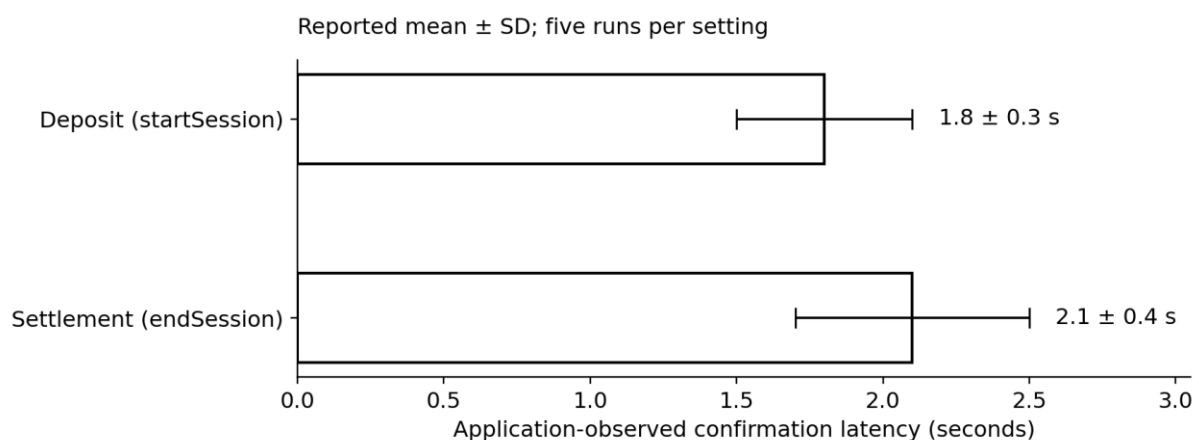


Figure 2. Fuji confirmation latency for deposit and settlement transactions. Error bars show the reported standard deviation over five runs per setting, not confidence intervals. Source: This study; summary values in Table 3.

Proof time versus confirmation time. The simplified circuit's mean proof-generation time of 0.21 s is below the two confirmation means on the specified desktop. This descriptive comparison does not predict the cost of full station-attestation verification or performance on a smartphone or vehicle unit.

The confirmation observations inform prototype workflow design, but total session latency also includes metering, local communication, and any token approval or setup operations. No user study or service-level assessment establishes acceptable commercial performance.

5.3 Gas Usage and Transaction Cost

Settlement gas. endSession uses approximately 320,000 gas, compared with approximately 50,000 gas for the plaintext baseline. The rounded transaction totals imply about 270,000 additional gas and a ratio of 6.4×. Figure 3 presents this implementation-level difference. As explained in Section 4.5, it neither isolates verifier gas nor compares systems with equivalent functionality and security.

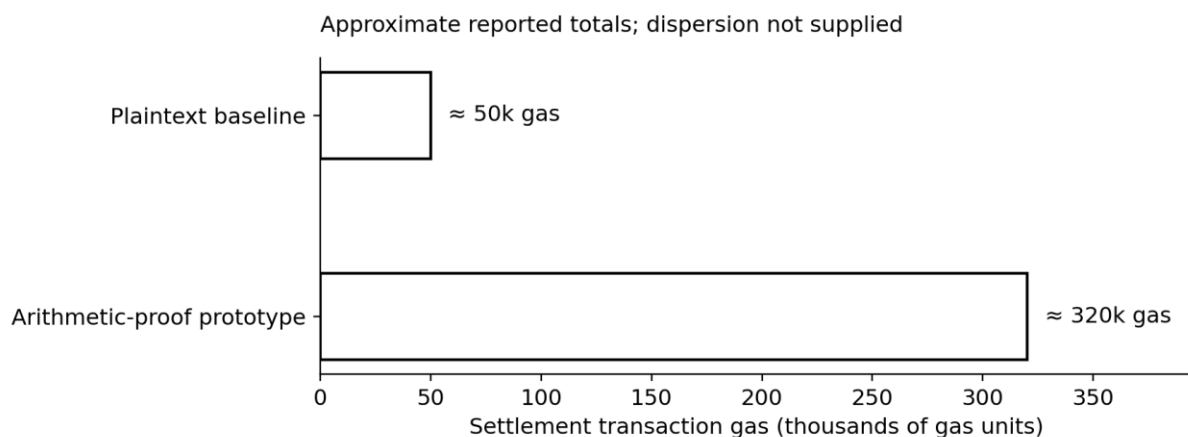


Figure 3. Approximate settlement gas for the plaintext baseline and simplified proof-bearing prototype. Values are rounded transaction totals without reported dispersion; the implementations are not security-equivalent. No authorization-gas result is available. Source: This study; summary values in Table 3.

Illustrative fee calculation. Fees vary with gas price; gas use is therefore the primary cost metric. For a hypothetical fixed price of 25 nAVAX/gas, the additional 270,000 gas would cost 0.00675 AVAX. This is an arithmetic illustration, not a measured fee or a claim about the current network gas price. The ERC-20 token used for deposits and payment is distinct from the Fuji AVAX used for fees.

$$\text{additional fee} = 270,000 \times 25 \times 10^{-9} \text{ AVAX} = 0.00675 \text{ AVAX}$$

5.4 Throughput and Scalability Implications

Unvalidated harness estimate. An approximate submission estimate of 140 transactions/minute is associated with a harness described as sequential. Its concurrency, observation interval, overlap between calls, and counts of confirmed transactions and successfully settled sessions are unspecified. It is therefore not a measurement of completed-session throughput, sustainable system throughput, or Avalanche capacity.

For a strictly sequential schedule that waits for both mean confirmations, $60/(1.8 + 2.1)$ is approximately 15.4 sessions/minute before proof generation, simulated charging, and other overheads. This conditional arithmetic check is not a throughput measurement; it illustrates why a submission count cannot be converted into a completed-session rate without a defined schedule and observation window.

Load-evaluation boundary. The experiment does not vary offered load or characterize concurrent users, queues, failure rates, or confirmation-time stability. Network capacity and sustained application throughput therefore remain unmeasured. Platform documentation describes architectural options rather than application throughput demonstrated here [7].

5.5 Correctness and Security Checks

Functional checks exercise the escrow lifecycle and deployed proof verifier. Table 4 separates their observed outcomes from unresolved security properties. In particular, the checks do not include a malicious client generating a new proof for substituted energy.

Table 4. Functional-test evidence and unresolved security requirements.

Check or property	Evidence reported	Scope or remaining requirement
Invalid proof or altered public C	endSession reverted.	Checks consistency with the deployed relation; not authenticated meter provenance.
Second settlement of a closed sid	Rejected by session-state logic.	Does not by itself establish proof binding across two active sessions.
Caller restrictions	Depositor settlement and designated-station timeout checks reported.	Does not establish honesty of either authorized party.
Timeout closure	Unpaid session closed in a functional test.	No comprehensive malicious-meter, over-deposit, or outage evaluation.
Valid new proof using substituted E'	Not reported.	Required to assess underbilling with a modified client.
Full attestation and tariff binding	Not enforced end-to-end in the evaluated circuit.	Requires an authenticated signature/commitment path and public-input/state checks.
Consumption confidentiality	Fails analytically for public exact linear pricing.	Public C and p reveal E, regardless of proof validity.

The observed reverts establish behavior only for the tested inputs and contract paths. Rejecting an invalid proof or altering C without generating a new proof does not test whether the contract accepts a different energy witness. Station-authenticated billing therefore remains unvalidated.

Unresolved substituted-energy experiment. For an active session with an accepted station attestation for E, a modified depositor client would retain the session, tariff, station, and deposit context but choose a permitted $E' < E$ and calculate $C' = p \cdot E'$. It would recompute any prover-controlled placeholder values, generate a new proof π' , and submit $\text{endSession}(\text{sid}, C', \pi')$. A matched honest settlement should be run in a separate session with equivalent parameters. The test must preserve the attestation, witnesses, public signals, proofs, transaction receipts, and state changes so that a rejection can be distinguished from an unrelated caller, balance, or session-state failure.

Acceptance of the substituted witness would demonstrate underbilling for the tested configuration. Rejection would require identifying the enforced predicate before concluding that station authentication is present. This experiment has not been performed in the available evaluation; the analysis is not evidence of an observed attack or an experimentally validated defense.

5.6 Privacy Assessment

Direct disclosure. Inspection of normal-path transaction arguments and events found no explicit E parameter; the transaction publishes C with π . This meter-field omission is a property of the interface, not consumption confidentiality.

Deterministic consumption recovery. Under the evaluated exact linear tariff and $p > 0$, the public values determine:

$$E = C / p. \quad (2)$$

Equation (2) is exact under the stated model. Zero knowledge does not conceal information already implied by the public statement and visible payment. A protocol with rounding or a non-injective tariff would need a separate analysis of the readings consistent with each disclosed payment; neither mechanism is evaluated here.

Other observables. Station and wallet identifiers, timing, deposit D, payment C, and refunds remain visible. Address rotation does not hide those amounts or prevent funding-trace and external-observation linkage. No linkability experiment is reported. The study also neither models nor tests protection of a fine-grained consumption time series or authorization credentials.

Timeout disclosure. The fallback explicitly supplies E or equivalent plaintext settlement information. This path requires separate privacy and dispute analysis in any future design that masks normal payments. In the evaluated linear-tariff prototype, normal settlement already reveals E through Equation (2).

5.7 Limitations

The evaluation is limited to five-run summaries on a shared testnet. Raw observations, complete circuit and setup artifacts, exact software and numerical configurations, and a controlled load experiment are unavailable. It also lacks mobile, physical-station, and instrumented energy measurements. The unresolved substituted-energy test and deterministic consumption disclosure are analyzed in Sections 5.5 and 5.6, respectively.

These limitations restrict the supported result to execution of the deposit-escrow and billing-arithmetic-proof workflow at the stated descriptive costs. Stronger conclusions require additional implementation artifacts and experiments.

6 Discussion

This section discusses requirements for a stronger implementation and possible deployment extensions. Implemented features and measurements are confined to Sections 4 and 5.

6.1 Practical Deployment Considerations

Metering integrity. The station's signature authenticates its statement, not the physical accuracy of delivered energy. A production design therefore needs trustworthy metering, protected keys, software-maintenance controls, and an auditable dispute process. Firmware-update mechanisms are relevant complementary work [17], but they are not integrated or tested here.

Wallet and key management. A deployment would require secure key storage, deposit funding, proof generation, recovery, and transaction-failure handling. Device-specific benchmarks are needed; address rotation should be evaluated against funding traces and other public metadata.

Fees and payment assets. The experiment separates ERC-20 settlement from Fuji AVAX fees. Alternative assets, sponsored transactions, or relayers would change the operational and trust model and require new measurements.

Tariff publication and numerical consistency. The station attestation, proof inputs, and contract state must use the same fixed session tariff and the numerical rules specified in Section 3.3.1. A richer tariff changes circuit requirements but does not improve confidentiality when its public output remains invertible.

Integration with charging infrastructure. OCPP provides a relevant interface between charging stations and management systems [19]. An event-to-station gateway is a possible integration route, but this study uses a station simulator and file-based communication, not a tested OCPP connector. Start/stop safety, lost events, retries, and physical metering would need explicit integration tests.

Failures, disputes, and abuse. Session cancellation, interrupted charging, delayed transactions, insufficient deposits, failed transfers, and disputed station readings require defined contract and operational policies. Caller restrictions and a timeout are not a complete fairness mechanism. Rate limits and admission rules may reduce resource abuse, but have not been evaluated as part of the reported workload.

6.2 Closing the Prototype-to-Protocol Gap

Two implementation routes could bind the proof's energy witness to the station. Both are unimplemented and require a complete security argument and fresh benchmarks. Their effect on billing integrity is separate from the public-payment leakage analyzed in Section 5.6.

In-circuit attestation verification. The circuit could verify a suitable station signature over the complete session-bound message in Section 3. The contract must select an authenticated station key and match the proof's session, tariff, and amount to stored state. The final benchmark must report constraint count, proving time and memory, setup artifacts, public-input layout, and transaction gas for that circuit—not reuse the arithmetic-only measurements.

Authenticated hiding commitment. Alternatively, the station could publish or authorize a randomized commitment to the session summary, and the SNARK could prove an opening consistent with $C = E \cdot p$. The commitment's provenance must be authenticated by the contract, its session and tariff must be bound, and its opening randomness must remain private with adequate entropy. A deterministic hash of a small-domain energy value is not automatically hiding, and a commitment chosen only by the user does not authenticate station metering.

A commitment-posting transaction may add an on-chain step and change latency, gas, and transaction counts per session. These costs must be measured for the complete authenticated construction rather than inferred from the placeholder circuit.

Adversarial validation. The substituted-energy test specified in Section 5.5 should be accompanied by tests for unauthorized station keys, different active-session identifiers, altered tariffs, reused commitments, invalid ranges, arithmetic wraparound, repeated settlement, and timeout boundaries. A station that signs a false physical reading requires a separate metering and dispute model, not only cryptographic rejection tests.

6.3 Extending Privacy Beyond Meter-Field Omission

Consumption confidentiality requires changing the public disclosure identified in Section 5.6, not merely strengthening the proof. In particular, authenticating a hidden witness does not hide an energy value uniquely determined by the public bill.

Shielded settlement could conceal transferred amounts; aggregation or appropriately designed off-chain settlement could reduce per-session disclosure. These are possible architectural changes, not integrated features. Each requires an explicit observer model, an analysis of deposits, refunds, timing, and withdrawals, and measurement of the added complexity. Merely adopting tiered pricing, rotating addresses, or posting a hidden commitment does not automatically prevent inference.

The timeout path and any regulatory or dispute disclosure should be assessed separately. A privacy claim must state which parties learn aggregate and per-session consumption, under which failures, and with what auxiliary information. No differential-privacy, anonymity-set, or quantitative inference-resistance result is asserted in the current study.

6.4 Scalability Options: Application-Specific Chains and Batching

Avalanche documentation describes application-specific L1 architectures [7]; batching or aggregation could also change verification amortization. These options are unmeasured alternatives, not a way to extrapolate capacity from the existing harness. Moving to another chain or combining proofs changes infrastructure, data availability, and possibly the trust and fee model.

A future load experiment should report offered, submitted, confirmed, and successfully settled rates separately; concurrency and arrival patterns; warm-up and measurement windows; queue sizes; failures; and latency distributions under increasing load. Testing should continue long enough to distinguish transient bursts from a stable service rate. No numerical capacity prediction is made here.

6.5 Economic and Regulatory Considerations

Drivers and operators would need to assess deposit liquidity, fee burden, wallet usability, dispute resolution, and station integration. Cross-operator roaming, commercial cost savings, and user acceptance of the public disclosure were not evaluated.

A production system would require jurisdiction-specific assessment of payment, consumer-protection, data-protection, and energy-reporting obligations. The availability of a meter reading at the station does not by itself establish regulatory compliance, and blockchain records do not remove obligations related to incorrect billing or disclosure. This study makes no legal-compliance claim.

6.6 Broader Applicability

Deposit escrow with verifiable usage arithmetic may be relevant to other metered services, including mobility and IoT subscriptions. Generalization remains conditional on authentic usage measurement, an appropriate settlement/dispute process, and an explicit privacy analysis. Public unit pricing creates the same inference problem outside EV charging whenever the disclosed amount uniquely determines usage.

7 Conclusion and Future Work

This study integrates deposit authorization, a Groth16 billing-arithmetic proof, and on-chain payment and refund on Avalanche, with a timeout path for unpaid sessions. Its contribution is an application-level proof of concept and a distinction between executed settlement, the verified arithmetic relation, and the requirements of a station-authenticated protocol.

Across five runs per setting, mean deposit and settlement confirmation times are 1.8 s and 2.1 s, and desktop proof generation averages 0.21 s. Proof-bearing settlement uses approximately 320,000 gas. These descriptive costs apply only to the simplified circuit and do not establish sustainable throughput or performance at the security level of the proposed complete protocol.

Two security and privacy boundaries remain decisive. The evaluated SNARK does not authenticate its energy witness to the station, and the public linear tariff reveals consumption as $E = C/p$ despite meter-field omission. Resolving witness authentication would address the first boundary but would not remove the second.

The next implementation must enforce station-authenticated witness binding and fully specified integer ranges, scaling, and tariff rules. Validation requires the substituted-energy experiment, reproducible circuit and setup artifacts, raw per-run records, and a larger controlled workload. Shielded or aggregated settlement, mobile proving, and physical-station integration remain separate extensions requiring their own security, privacy, and cost evaluations.

DECLARATIONS

Funding. This research received no external funding.

Conflicts of interest. The author declares that there are no conflicts of interest.

Ethics approval. Not applicable. This study did not involve human participants, animals, or any activities requiring ethical approval.

Informed consent. Not applicable. This study did not involve human participants.

Data and code availability. No datasets or source code are publicly available in connection with this study.

Author contributions. The sole author was responsible for the conceptualization, methodology, software development, investigation, analysis, validation, and writing of the manuscript. The author reviewed and approved the final version of the manuscript.

Use of artificial intelligence. The author used ChatGPT (OpenAI) solely for grammatical correction, language editing, and improving the clarity and readability of the manuscript. The AI tool was not used to generate research findings, conduct scientific analysis, or draw conclusions. The author reviewed and verified the final content and takes full responsibility for the manuscript.

Acknowledgements. None.

REFERENCES

- [1] T. Lauser, N. Yuca, D. Kern, N. Matyunin, S. Katzenbeisser, and C. Krauß, Oblivious Plug&Charge: A privacy-preserving EV charging scheme based on ORAM, in *Proc. 3rd USENIX Symposium on Vehicle Security and Privacy (VehicleSec 25)*, 2025, pp. 221–232. Available: <https://www.usenix.org/conference/vehiclesec25/presentation/lauser>.
- [2] R. Saha et al., The blockchain solution for the security of Internet of Energy and electric vehicle interface, *IEEE Trans. Veh. Technol.*, 70(8) (2021), 7495–7508, doi: 10.1109/TVT.2021.3094907.
- [3] H. Liu, Z. Lv, Z. Song, S. Zhou, W. Liu, and M. Fang, Application of blockchain technology in electric vehicle charging piles based on electricity Internet of Things, *Wireless Commun. Mobile Comput.*, 2022 (2022), Art. no. 8533219, doi: 10.1155/2022/8533219.
- [4] H. Li, D. Han, and M. Tang, A privacy-preserving charging scheme for electric vehicles using blockchain and fog computing, *IEEE Syst. J.*, 15(3) (2021), 3189–3200, doi: 10.1109/JSYST.2020.3009447.
- [5] C. Gentry and S. Halevi, Implementing Gentry's fully-homomorphic encryption scheme, in *Advances in Cryptology—EUROCRYPT 2011, LNCS*, vol. 6632 (2011), pp. 129–148, doi: 10.1007/978-3-642-20465-4_9.
- [6] J. Groth, On the size of pairing-based non-interactive arguments, *Cryptology ePrint Archive*, Paper 2016/260, 2016. Available: <https://eprint.iacr.org/2016/260>.
- [7] Avalanche, Primary Network and Avalanche L1s, Builder Hub, official documentation. Available: <https://build.avax.network/docs/primary-network> and <https://build.avax.network/docs/avalanche-l1s>. Accessed: Sep. 10, 2026.
- [8] M. Roon, Share&Charge platform—Phase 1, Share&Charge, Jan. 22, 2019. Available: <https://medium.com/share-charge/share-charge-platform-phase-1-a65c7f8fc27f>.
- [9] D. Zhai et al., EPDB: An efficient and privacy-preserving electric charging scheme in Internet of Robotic Things, *IEEE Internet Things J.*, 11(20) (2024), 32464–32477, doi: 10.1109/JIOT.2024.3426536.
- [10] C.-M. Chen, Q. Miao, S. Kumari, M. K. Khan, and J. J. P. C. Rodrigues, A privacy-preserving authentication protocol for electric vehicle battery swapping based on intelligent blockchain, *IEEE Internet Things J.*, 11(10) (2024), 17538–17551, doi: 10.1109/JIOT.2024.3360280.
- [11] K. M. Shafin and S. Reno, A hybrid blockchain-enabled payment system for efficient electronic vehicle charging payments on electrified roads, *J. King Saud Univ.—Comput. Inf. Sci.*, 37 (2025), Art. no. 16, doi: 10.1007/s44443-025-00013-6.
- [12] P. W. Khan and Y.-C. Byun, Blockchain-based peer-to-peer energy trading and charging payment system for electric vehicles, *Sustainability*, 13(14) (2021), Art. no. 7962, doi: 10.3390/su13147962.
- [13] D. D. Sharma, S. N. Singh, and J. Lin, Blockchain-enabled secure and authentic Nash equilibrium strategies for heterogeneous networked hub of electric vehicle charging stations, *Blockchain: Research and Applications*, (2024), Art. no. 100223, doi: 10.1016/j.bcr.2024.100223.
- [14] A. Aldweesh and S. Alangari, Privacy-preserving EV charging authorization and billing via blockchain and homomorphic encryption, *World Electr. Veh. J.*, 16(8) (2025), Art. no. 468, doi: 10.3390/wevj16080468.
- [15] B. L. Sahu et al., Blockchain and machine learning integrated secure driver behavior centric electric vehicle insurance model, *IEEE Trans. Intell. Transp. Syst.*, 25(11) (2024), 19073–19087, doi: 10.1109/TITS.2024.3439822.
- [16] S. R. Bandarapu et al., Blockchain-based federated learning framework for malicious node detection in internet of vehicles (IoV) networks using fog and cloud computing, *J. King Saud Univ.—Comput. Inf. Sci.*, 37 (2025), Art. no. 120, doi: 10.1007/s44443-025-00134-y.
- [17] A. Aldweesh, Blockchain-based secure firmware updates for electric vehicle charging stations in Web of Things environments, *World Electr. Veh. J.*, 16(4) (2025), Art. no. 226, doi: 10.3390/wevj16040226.
- [18] iden3, snarkjs: zkSNARK implementation in JavaScript and WASM, project documentation. Available: <https://github.com/iden3/snarkjs>. Accessed: Sep. 10, 2026.
- [19] Open Charge Alliance, Open Charge Point Protocol, official protocol overview. Available: <https://openchargealliance.org/protocols/open-charge-point-protocol/>. Accessed: Sep. 10, 2026.