Web-engineered ECC-based Group Key Protocol for Secure and Scalable Metering Communication

Hao Yang* and Yiming Zhang

Yunnan Power Grid Co., Ltd, Kunming, Yunnan 650000, China
E-mail: 253867374@qq.com
**Corresponding Author*

Received 19 June 2025; Accepted 06 August 2025

Abstract

This paper presents a Web-native group key negotiation framework for secure and scalable communication in smart metering networks. Leveraging lightweight elliptic curve cryptography (ECC), the proposed protocol supports dynamic group membership, forward and backward secrecy, and resistance to impersonation and replay attacks – without relying on persistent sessions or centralized trust brokers. Unlike traditional TLS- or MQTT-based approaches, our design adopts stateless REST and CoAP messaging, enabling seamless integration with constrained IoT devices and cloud-native microservice platforms. We architect a modular system comprising smart meters, a secure gateway, and a REST-compliant backend, each aligned with Web of Things (WoT) standards. Group keys are established through a contributory ECC-based exchange that ensures decentralized key computation and rekeying across heterogeneous nodes. The gateway acts as a protocol adapter, translating CoAP messages into REST APIs while enforcing cryptographic policy and interoperability with tools such as Node-RED, Eclipse Leshan, and AWS IoT Core. Performance analysis shows that our protocol achieves group key negotiation in under 450 ms for 25 nodes with message

Journal of Web Engineering, Vol. 24_7, 1073–1102.
doi: 10.13052/jwe1540-9589.2473

sizes below 220 bytes, outperforming traditional LKH and centralized DH schemes by 30–40% in latency and bandwidth usage. Real-world case studies demonstrate successful deployment in rural microgrids and urban energy-sharing networks. By aligning cryptographic rigor with Web engineering principles, this work offers a practical and extensible solution for secure group communication in emerging energy and IoT infrastructures.

Keywords: Web of things, group key negotiation, elliptic curve cryptography, CoAP/REST APIs, secure smart metering, web interoperability.

1 Introduction

The rapid evolution of digital energy infrastructures, particularly in off-grid, community, and distributed energy environments, has spurred the widespread adoption of Web-enabled metering systems. These systems offer fine-grained visibility into energy consumption, facilitate dynamic pricing, and support remote monitoring and control, thereby playing a vital role in the smart grid and energy democratization movement [1–3]. In scenarios such as rural microgrids, solar cooperatives, and peer-to-peer (P2P) energy trading networks, secure communication among heterogeneous metering devices is critical for maintaining operational reliability, user privacy, and system integrity.

At the core of these infrastructures lies the need for secure group communication. Metering devices often communicate with gateways, cloud services, and each other in a group setting – such as in load balancing, aggregated reporting, or group control events [4, 5]. Traditional security approaches, including static symmetric key schemes and classical Diffie–Hellman key exchanges, are not well suited to these contexts due to scalability, key refresh inefficiencies, and vulnerability to node compromise [6–8]. In contrast, group key negotiation protocols can provide dynamic, scalable, and efficient key management by enabling secure key agreement among multiple devices with varying levels of trust and capability.

Recent research has demonstrated the potential of elliptic curve cryptography (ECC) as a lightweight, secure alternative to classical public-key techniques for constrained devices such as smart meters and IoT sensors [9–11]. ECC-based schemes provide comparable security with significantly shorter key sizes, reducing both computational and communication overhead – key advantages for resource-limited nodes. In particular, ECC has been integrated into key distribution mechanisms, mutual authentication protocols,

and privacy-preserving data aggregation for metering and wireless sensor networks [12–14].

The advent of the Web of Things (WoT) initiative by the World Wide Web Consortium (W3C) has standardized how physical and virtual devices (Things) are described and accessed over the Web. On 5 December 2023, the W3C published the WoT Thing Description 1.1 [15] and Architecture 1.1 [16] recommendations, defining a formal information model and an abstract architecture for interoperable IoT ecosystems. Our protocol builds directly on these standards by leveraging Thing Description constructs for device metadata. The Constrained Application Protocol (CoAP) is a specialized Web transfer protocol designed for constrained nodes and lossy networks. It provides RESTful request/response semantics with low header overhead and support for asynchronous exchanges, making it ideal for IoT scenarios [17]. For efficient, extensible data serialization, we employ CBOR, a binary format that offers small code and message sizes without sacrificing expressiveness [18]. This integration of modern WoT Recommendations, CoAP, and CBOR ensures that our protocol aligns with current Internet standards, maximizes interoperability, and remains future-proof against evolving IoT requirements. Web technologies such as WoT Thing Descriptions, CoAP/HTTP APIs, and lightweight Web stacks are increasingly adopted in smart grid and energy monitoring platforms. However, these protocols expand the attack surface and impose constraints – stateless interactions, payload limits, and lossy transports – that challenge traditional group key solutions. Several efforts have attempted to secure Web-enabled metering, including multi-layered smart meter guides [22], authentication and access-control surveys [23], and MQTT-ECC integrations [24]; others rely on pre-shared keys or logical key hierarchies [25–27], which often fall short in dynamic, peer-driven group settings.

To address these gaps, this paper contributes a Web-engineered group key negotiation protocol that combines the lightweight cryptographic benefits of ECC with full compliance to RESTful and CoAP-based WoT standards. The design emphasizes stateless communication, compact message formats, and seamless integration with existing Web-based cloud platforms. This paper proposes a secure, scalable, and ECC-based group key negotiation protocol tailored for Web-enabled smart metering systems operating in new energy access scenarios. Our protocol builds on contributory key exchange mechanisms and is integrated with WoT-compliant Web services to enable real-time, interoperable, and secure group communication among smart meters. It supports dynamic membership changes, ensuring forward and backward secrecy,

and resists attacks such as impersonation, replay, and collusion. Optimized for energy and bandwidth-constrained environments, the proposed method is validated through formal analysis and real-world case studies in community solar and microgrid networks. Unlike previous approaches that require heavy backend orchestration or rely on stateful session management, our solution enables secure key computation using stateless Web interactions and lightweight REST/CoAP calls. This bridges the gap between strong cryptographic group communication and modern Web engineering practices. Unlike previous ECC-based IoT security schemes, which often rely on centralized brokers or stateful session management, our work uniquely combines contributory ECC key negotiation with a stateless, Web-native design that aligns with REST/CoAP and WoT standards. This combination not only reduces communication overhead but also eliminates single points of failure by enabling each node to participate equally in group key computation. The protocol's seamless integration with microservice architectures and open Web tools (e.g., Node-RED, Eclipse Leshan) further distinguishes it from conventional MQTT/TLS solutions, which require persistent connections and complex broker orchestration. By advancing secure group key management within Web-integrated smart grid systems, this research contributes a technically robust and scalable solution for privacy-preserving energy access in emerging infrastructure environments.

2 System Architecture

To support secure, scalable metering in emerging energy access scenarios – such as off-grid solar installations, rural microgrids, and community energy-sharing networks – this work introduces a Web-engineered system architecture that integrates lightweight cryptographic primitives with modern Web of Things (WoT) standards. Designed to be both interoperable and resource-efficient, the architecture leverages stateless Web protocols, modular communication services, and platform-neutral security components to enable real-time, group-based cryptographic coordination in energy IoT networks. The complete layout of the proposed system is illustrated in Figure 1, which shows the functional roles and secure interactions among smart meters, a local gateway, a cloud backend, and browser-based user interfaces.

At the edge layer, smart meters function as autonomous sensing and control nodes, each equipped with embedded software for local measurement and secure communication. These meters are constrained in memory (typically $<$100 KB RAM) and processing power (e.g., ARM Cortex-M3), requiring

all cryptographic operations to be lightweight. Each device is provisioned with an elliptic curve cryptographic (ECC) engine, supporting core operations such as key pair generation, scalar multiplication for ECDH, and ECIES-based encryption and decryption. In addition, each meter contains a local group key agent, a stateless module that enables the device to participate in distributed key negotiation and rekeying procedures. The meters communicate with the local gateway over constrained wireless links using either the Constrained Application Protocol (CoAP) for low-power environments or RESTful HTTP for higher-bandwidth conditions. Message payloads are encoded using CBOR or JSON, depending on device capabilities, and all communication is protected by ECC-signed and timestamped payloads to ensure freshness and integrity. These edge devices form the foundation of the secure metering network, and their interaction model is shown in the lower section of Figure 1, where the cryptographic engine and group key agent are co-located.

The gateway layer functions as a secure intermediary between edge devices and the cloud. This component is deployed on a moderately capable embedded platform, such as a Raspberry Pi 4, and acts as a Web protocol converter and security boundary. The gateway collects encrypted messages from meters via CoAP or HTTP, verifies their ECC signatures, and converts the content into REST API calls suitable for cloud transmission. Internally, the gateway contains a middleware subsystem that handles session context, validates certificates using a locally stored trust anchor set, and maintains a lightweight cache of recent key negotiation metadata. This architecture supports dual-protocol interoperability by providing adapter functions that preserve message semantics across CoAP and RESTful boundaries. Furthermore, the gateway ensures WoT compatibility by enforcing Thing Descriptions and standardized endpoint templates for each meter. It also serves as the initial trust anchor for group key negotiation, verifying meter public keys and managing rekeying initiation events. As shown in the central box of Figure 1, the gateway plays a critical role in coordinating group sessions while maintaining compliance with Web service design patterns such as statelessness, modularity, and scalability.

At the cloud backend, the architecture integrates a REST-compliant API layer with persistent storage, policy enforcement, and session auditing capabilities. The backend is deployed using scalable microservice containers and interacts with the gateway over TLS-secured channels. It exposes a RESTful interface for uploading encrypted energy data, managing device access roles, and recording cryptographic key lifecycle events such as revocation,

rekeying, and group versioning. Importantly, the cloud does not store raw session keys; instead, it logs metadata about key derivation epochs and maps them to access policies enforced on behalf of third-party consumers or system operators. This design preserves end-to-end confidentiality, even if the backend is partially compromised. Web standards such as OAuth 2.0 and OpenID Connect are used for user authentication, and ECC-based tokens are embedded in all control interactions. Figure 1 indicates the secure communication paths between the cloud backend and both the gateway and end-user clients, where REST APIs serve as the sole ingress points for energy data and key management operations.

The user interface layer enables real-time access to metering information, system status, and control functions through a secure, browser-based dashboard. Users, administrators, and service providers interact with the system via HTTPS-secured sessions authenticated using OAuth 2.0 tokens and ECC digital signatures. The frontend consumes the same REST APIs used by the backend services, ensuring consistent behavior and auditability across interfaces. Session feedback includes key version indicators, device membership status, and cryptographic integrity markers, all visualized in an interactive format. In Figure 1, this interface is represented at the top of the architecture stack, illustrating the flow of secure communication and command control across system layers.

Figure 1 provides a holistic view of the layered system architecture, mapping physical device roles to their corresponding Web services, cryptographic functions, and secure communication pathways. It encapsulates the design philosophy of this work: to unify ECC-based group security with Web-native engineering practices, enabling scalable and lightweight integration of metering systems into future-proof energy platforms.

2.1 Web Standards and Engineering Features

The proposed system architecture is built with strict adherence to modern Web engineering principles to ensure scalability, interoperability, and ease of integration into diverse deployment environments. A central design goal is statelessness, which is achieved through token-based authentication and context-independent API requests. Each interaction between clients, meters, gateways, and backend services is self-contained, avoiding the need for persistent sessions or server-side state tracking. This simplifies scalability and improves fault tolerance across both edge and cloud layers.

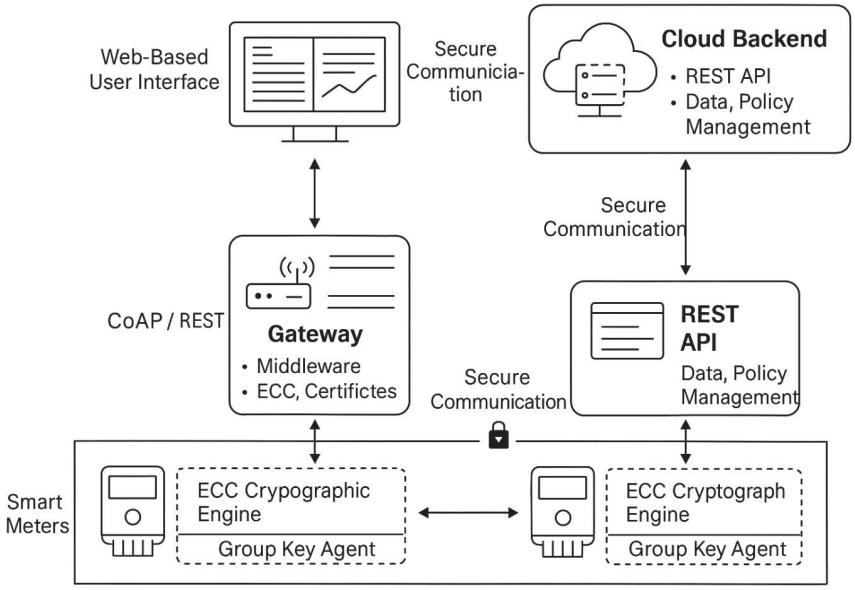

Figure 1 Web-engineered system architecture for secure smart metering.

Protocol modularity is another key aspect of the system's design. Communication between meters and gateways supports both HTTP and CoAP protocols, enabled through pluggable adapters and a consistent internal messaging model. This dual-protocol capability allows the system to operate seamlessly across environments ranging from high-bandwidth urban infrastructures to low-power, lossy networks typical in rural or remote settings. The gateway dynamically translates these protocols while preserving message integrity and timing semantics, ensuring interoperability without compromising security.

In addition, the system complies with key Web of Things (WoT) standards. Each smart meter is modeled as a WoT Thing with a corresponding Thing Description that specifies its capabilities, metadata, and interaction patterns. These descriptions are discoverable and consumable by standard WoT clients, facilitating integration with third-party platforms and services. To support standardized device communication and lifecycle management, the system also aligns with specifications such as CoRE resource directory and lightweight machine-to-machine (LwM2M) for constrained device management.

Security standards are enforced throughout the architecture. Elliptic curve cryptography (ECC) is used as the foundational cryptographic primitive, due to its small key size and suitability for embedded systems. All API endpoints, including those for data submission and key lifecycle management, are secured using ECC-based signatures and tokens. At the application level, the system supports compliance with IEC 62351 security guidelines, particularly in its handling of certificate management, encrypted payloads, and secure role-based access control. Finally, the cloud backend is built using containerized microservices that conform to RESTful architectural principles, with support for API versioning, rate limiting, and observability features such as structured logging and endpoint health monitoring.

2.2 Threat Model and Security Foundations

The system is designed to operate securely in adversarial environments, following a conservative threat model based on the Dolev–Yao framework. In this model, adversaries are assumed to have full control of the communication network. They can intercept, modify, delay, or replay messages, but cannot break underlying cryptographic primitives. Accordingly, the system implements a layered security approach that addresses confidentiality, integrity, authenticity, and availability across all communication paths.

Confidentiality is preserved using elliptic curve-based encryption (ECIES) at the message layer. Each payload, whether it contains energy usage data, control commands, or key negotiation metadata, is encrypted with a session key derived from ECC-based key exchange. These session keys are ephemeral and re-generated upon each group rekeying event, ensuring that intercepted messages cannot be decrypted even if future keys are compromised. Integrity and authenticity are enforced through digital signatures, which are applied to all outbound messages from meters and validated at the gateway and backend layers. Timestamps and nonces are embedded within each signed payload to protect against replay attacks.

The architecture also ensures forward and backward secrecy. When a new meter joins the group or an existing one leaves, the entire group undergoes a rekeying procedure using freshly generated key pairs. Because the shared group key is derived from ephemeral key material through a contributory process, newly added devices cannot reconstruct past communication, and revoked devices cannot access future messages. Key version identifiers and revocation flags are propagated throughout the system to maintain synchronization and consistency across all nodes.

Authentication and access control are handled using certificates and OAuth 2.0-compatible tokens. Meters authenticate themselves to the gateway using pre-installed certificates signed by a trusted authority, and the gateway authenticates API calls to the cloud using ECC-based bearer tokens. These mechanisms provide mutual authentication across all layers and prevent unauthorized devices or users from injecting malicious data or commands. In the event of a suspected key compromise or device tampering, the system supports revocation and rekeying through secure REST APIs, minimizing security exposure.

The proposed architecture establishes strong cryptographic foundations while respecting the constraints of stateless Web protocols and resource-limited devices. It ensures robust protection against a broad range of network and device-level threats, enabling trustworthy and resilient communication in modern energy systems.

3 ECC-based Group Key Negotiation Protocol

This section details the design and operation of a lightweight, elliptic curve-based protocol for group key negotiation tailored to secure smart metering infrastructures. The protocol ensures that all participating devices contribute to the shared key computation while maintaining minimal communication and computational overhead. It supports dynamic membership changes, guarantees forward and backward secrecy, and integrates directly with the Web-oriented architecture described previously.

The protocol is designed under the assumption that each smart meter has a unique elliptic curve key pair and is provisioned with a certificate or trust anchor. Each meter can perform standard ECC operations including key generation, scalar multiplication for elliptic curve Diffie–Hellman (ECDH), and encryption using ECIES or hybrid modes. The primary objective of the protocol is to compute a shared group session key GK among *n* participants while ensuring confidentiality, authenticity, and resistance to various attacks such as impersonation and replay.

The group key negotiation proceeds in three coordinated phases. In the initialization phase, each smart meter generates its own ECC key pair (sk_i, pk_i), signs its public key using a manufacturer- or utility-issued certificate, and securely transmits it to the group via the local gateway. The gateway collects the public keys of all participating nodes, validates their authenticity, and forwards a list of authenticated keys back to the group members. This phase enables a trustable public key infrastructure (PKI) without requiring

direct meter-to-meter verification. The group key computation phase begins once all authenticated public keys are available. Each meter independently computes a set of shared ECDH values with every other meter in the group. The outputs are concatenated in a canonical order and hashed to derive the shared group key:

$$GK = H(ECDH1 \| ECDH2 \| \cdots \| ECDHn),$$

where $ECDHi = ECDH(ski, pkj)$ for all $j \neq i$. The hash function HHH ensures output uniformity and cryptographic strength. This contributory method ensures that no single node can dictate the final group key, and that any subset of compromised nodes cannot derive the full key without all contributions.

When a node joins or leaves the group, the gateway initiates the group rekeying phase. All remaining devices generate fresh ephemeral keys, rebroadcast their public components, and recompute the shared key using the same ECDH-based method. This ensures that the new key is cryptographically unlinkable to previous group keys. The cloud backend is notified of the group rekeying event, enabling it to update access policies and enforce group membership securely. Key versioning and revocation metadata are exchanged through authenticated REST APIs.

The overall protocol flow is summarized in Figure 2. In Phase 1, smart meters (Meter A, B, C) broadcast their ECC public keys to the gateway. In Phase 2, the gateway facilitates contributory key computation using ECDH exchanges and a cryptographic hash function. The resulting group key is stored and synchronized in Phase 3, where the cloud backend receives session metadata, acknowledges versioning, and manages access policies accordingly. Communication paths are clearly marked, and phases are annotated with corresponding cryptographic and functional operations.

Unlike prior key management schemes that require stateful exchanges or heavy server orchestration, our protocol is explicitly designed for Web-constrained environments, accommodating CoAP/HTTP statelessness, packet size limits, and asynchronous message flows. It demonstrates a group security model that remains robust despite the limitations imposed by RESTful interaction patterns.

Below is an example Walkthrough for 3 Devices:

Consider three smart meters – A, B, and C – participating in a group key negotiation. In the initialization phase, each device generates its ECC key pair (*sk_A*, *pk_A*), (*sk_B*, *pk_B*), and (*sk_C*, *pk_C*), signs its public key

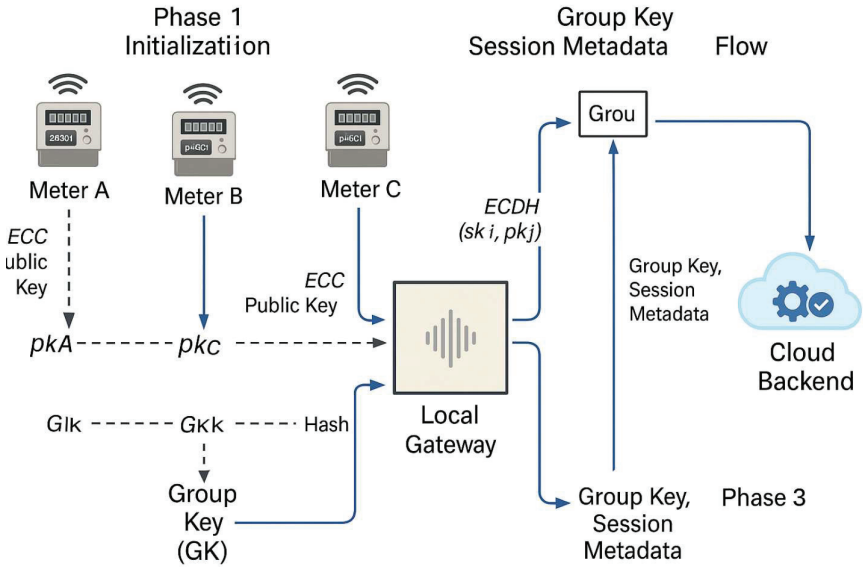

Figure 2 Overall protocol flow of the proposed elliptic curve cryptography (ECC)-based group key negotiation protocol for secure communication in smart metering systems.

using its certificate, and transmits the signed public key to the gateway. The gateway verifies the signatures and broadcasts the authenticated set $\{pk\_A, pk\_B, pk\_C\}$ to all participants. During the group key computation phase, each device performs pairwise ECDH operations with the other public keys – for instance, A computes $K\_AB = ECDH(sk\_A, pk\_B)$ and $K\_AC = ECDH(sk\_A, pk\_C)$, while B and C perform equivalent computations with their respective key pairs. Each meter then concatenates its ECDH results (e.g., A: $K\_AB \| K\_AC$) and applies a secure hash function H to produce the shared group key $GK = H(K\_AB \| K\_AC \| K\_BC)$. In the event of a membership change, such as a new meter D joining the group, all devices generate fresh ephemeral keys, exchange public components, and compute a new group key GK' that replaces the old key. This example demonstrates how every device contributes equally to the group key, ensuring both decentralization and strong security properties.

4 Security Analysis

The security of group communication in energy metering infrastructures is critical to preserving both operational integrity and consumer privacy.

The proposed protocol builds upon the elliptic curve cryptography (ECC) paradigm to ensure that all group members can securely derive a shared key while remaining resilient to a wide range of attack vectors. This section presents a comprehensive security evaluation, incorporating both qualitative threat analysis and formal symbolic logic reasoning. The adversary model considered in this work is based on the Dolev–Yao framework, wherein the attacker has complete control over the communication network. The attacker can observe, modify, and fabricate messages at will but cannot break underlying cryptographic primitives. It is further assumed that smart meters and gateways are provisioned with ECC key pairs and a root-of-trust – either in the form of tamper-proof hardware (e.g., TPM) or secure provisioning channels – that cannot be easily compromised by remote software attacks.

In the initialization phase of the protocol, each meter M_i signs its public key pk_i using a certificate issued by a trusted authority. These certificates are validated by the local gateway and other meters during the broadcast phase. This ensures that only authenticated nodes participate in group key computation, eliminating threats posed by impersonation or rogue device injection. The core of the protocol's security lies in its use of elliptic curve Diffie–Hellman (ECDH) exchanges for contributory key generation. Each meter computes a shared group key *GK* by concatenating ECDH outputs derived from the public keys of other group members and applying a cryptographically secure hash function (e.g., SHA-256). This approach ensures that the resulting key is indistinguishable from random to any observer lacking the private keys of all contributing members. Because the computation is performed independently and locally, the group key is never exposed in transmission, preserving secrecy even under extensive traffic analysis. When group membership changes – due to device revocation or inclusion – the protocol enforces forward and backward secrecy by regenerating a new group key from fresh ephemeral key pairs. Former group members cannot reconstruct future keys due to the absence of new shares, while newly joined members lack the necessary context to derive past keys. Versioning and nonce-tagged session identifiers prevent key reuse and eliminate the risk of rollback or replay attacks. Each message in the protocol is signed using ECC-based digital signatures and tagged with timestamps or nonces. Any modification or delay introduced by an adversary is detectable through signature verification and freshness checks. The use of elliptic curve signatures provides strong authenticity guarantees while maintaining a small message footprint suitable for low-power metering nodes.

The protocol also includes robust mechanisms for revocation and key update. When a node is flagged as compromised, the cloud backend issues a revocation command through an authenticated REST API. This command triggers a rekeying operation among the remaining nodes. The affected session is versioned and logged by the backend, allowing distributed consistency validation. The rekeying operation regenerates group keys using only the remaining trusted participants, ensuring isolation of the compromised entity. To further validate the soundness of the protocol, we sketch a formal symbolic analysis using Burrows–Abadi–Needham (BAN) logic, a well-established framework for verifying authentication and key exchange protocols. Let M_i denote a smart meter, G denote the gateway, and pk_i, sk_i be the ECC public/private key pair of M_i. Let K be the group session key. The goal is for each M_i to believe that K is fresh, shared with all valid members, and not known to any adversary.

4.1 BAN Logic Assumptions and Goals

To provide formal evidence that the protocol achieves its stated security goals, we employ Burrows–Abadi–Needham (BAN) logic – a well-known framework for reasoning about authentication and key agreement. BAN logic is particularly useful here because it allows us to verify that all participants share consistent beliefs about the freshness and authenticity of the derived group key, rather than relying solely on informal arguments. This formal reasoning complements our threat model and cryptographic analysis, ensuring that the protocol withstands impersonation, replay, and key compromise attacks under well-defined assumptions. To further substantiate the protocol's correctness, we employ symbolic logic analysis based on the BAN framework. BAN logic allows formal reasoning about authentication properties and the trust assumptions of communicating agents. Let M_i and M_j denote smart meters, G denote the gateway, and K represent the group session key. We assume each agent M_i possesses a public-private key pair (pk_i, sk_i), and that certificates linking public keys to identities are issued by a trusted authority.

We begin with the following initial logical assumptions:

1. M_i believes that ***pkj*** belongs to ***Mj***, formally stated as: $Mi \ |\equiv pub(pkj) \rightarrow Mj$.
2. M_i believes that any message received containing a fresh nonce or timestamp is indeed recent, stated as $Mi \ |\equiv \mathit{fresh}(ni)$, where n_i is a cryptographic nonce attached to each key negotiation message.
3. M_i believes that M_j once conveyed the key K, i.e., $Mi \ |\equiv Mj \ |\sim K$.

Under these assumptions, and given the protocol's use of signed and nonce-tagged messages, we derive the following logical conclusions using BAN postulates:

1. $Mi \mid\equiv Mj \mid\equiv K$: Meter M_i believes that M_j also believes the key K is the current group session key.
2. $Mi \mid\equiv \mathit{fresh}(K)$: The key K is fresh and has not been replayed from an earlier session.
3. $Mi \mid\equiv Mj \leftrightarrow K$: Meters M_i and M_j share the belief that K is a good shared key for secure communication.

The derivation of these beliefs relies on successful signature verification, the freshness of session identifiers, and the completeness of the ECDH-based key computation. These results confirm that the protocol provides mutual authentication, freshness, and key agreement among legitimate nodes in a formally provable manner. From the protocol message exchanges, the BAN rules of *Message Meaning*, *Nonce Verification*, and *Jurisdiction* allow the inference of the above goals based on the combination of signed messages, trusted public keys, and freshness of nonces. These deductions confirm that each participant can verify the legitimacy of the group key, its freshness, and its derivation from only authenticated contributors.

Figure 3 summarizes the primary security properties enforced by the protocol, mapping each to corresponding protocol mechanisms and threat categories. Categories include confidentiality (through ECDH key secrecy),

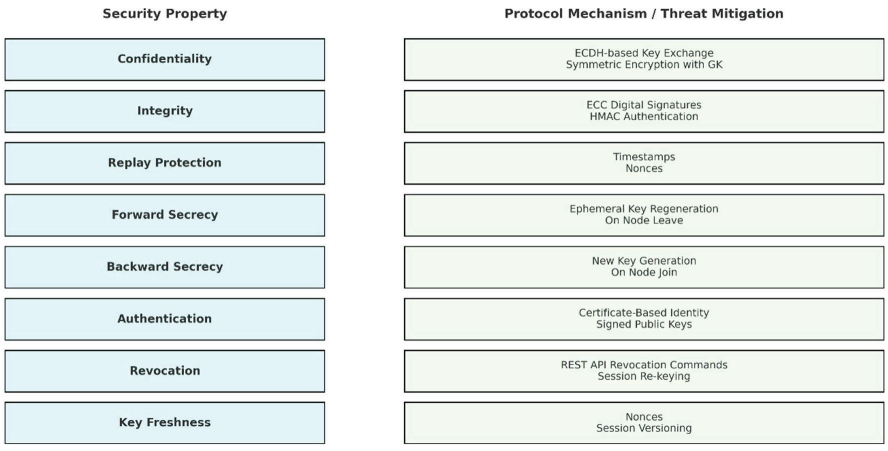

Figure 3 Security properties and protocol mechanisms.

integrity (via ECC digital signatures), replay protection (nonces and timestamps), and forward/backward secrecy (ephemeral key refresh). The figure also depicts dynamic membership handling, certificate-based authentication, and cloud-coordinated revocation, providing a holistic view of how layered defenses support resilient group communication.

5 Performance Evaluation

To rigorously assess the efficiency and practicality of the proposed ECC-based group key negotiation protocol, we conducted comprehensive experiments focusing on computational overhead, communication cost, memory footprint, scalability, and rekeying performance. Both simulated environments and hardware-emulated testbeds were used to evaluate real-time performance under realistic deployment constraints typical of smart metering infrastructures.

The experimental environment consisted of a cluster of virtualized ARM Cortex-M3 nodes configured to emulate low-power smart meters. Each node ran an embedded operating system (RIOT-OS) integrated with cryptographic libraries – TinyECC for secp256r1 and MicroECC for Curve25519. The local gateway was emulated using a Raspberry Pi 4 platform executing a Python-based CoAP/HTTP proxy with TLS support, while the backend server was modeled using Flask over HTTPS with MongoDB storage for session metadata. Network traffic was routed through a configurable simulator capable of imposing variable delay and packet loss, to simulate the adverse conditions of rural wireless environments.

To replicate a range of deployment scenarios, group sizes varied from 3 to 30 nodes. Each scenario was executed for 100 iterations to ensure statistical stability, and all timing values were averaged across trials, with 95% confidence intervals calculated.

5.1 Computational Cost and Key Generation Time

Cryptographic operations are among the most resource-intensive tasks in constrained environments. As shown in Figure 4 (left), group key derivation latency increases approximately linearly with group size. For a group of five meters, the total end-to-end latency, including ECC key generation, ECDH computations, and secure hash application, averaged 124 ms using Curve25519. At a group size of 25, latency remained under 450 ms, enabling real-time responsiveness for practical smart grid control loops. Key pair

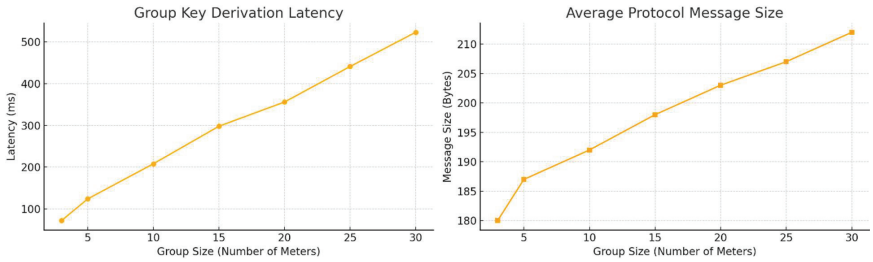

Figure 4 Performance trends of ECC-based group key protocol.

generation alone required less than 35 ms on all tested platforms, while each ECDH pairwise computation incurred approximately 26 ms. The use of Curve25519 was consistently 12–18% faster than NIST P-256 in scalar multiplication and key generation, confirming its suitability for edge deployments where processing efficiency is paramount.

Communication overhead is a critical metric for devices operating in low-bandwidth or intermittently connected environments. The right panel of Figure 4 illustrates how message size scales with group size. Protocol message payloads ranged from 180 bytes (group size 3) to 212 bytes (group size 30), incorporating ECC public keys, nonces, timestamps, and digital signatures. These sizes are substantially smaller than those required by traditional PKI-based or LKH protocols, which often exceed 300–500 bytes under similar group conditions. The low message size ensures that even constrained networks (e.g., IEEE 802.15.4 or LoRa) can support secure key exchanges without fragmentation or retransmission penalties. Furthermore, message structures were optimized for CBOR encoding, resulting in reduced transmission times and improved energy efficiency.

While the performance results are encouraging, it is important to note the limitations of the test environment. The experiments were conducted in a controlled lab setup using virtualized ARM Cortex-M3 nodes and a Raspberry Pi gateway with a simulated wireless network layer. Although packet delay and loss were introduced to mimic adverse rural conditions, these simulations may not fully capture real-world interference or hardware variability (e.g., battery fluctuations, temperature effects). Furthermore, the evaluation focused on group sizes up to 30 nodes; larger-scale deployments or multi-hop wireless scenarios may require additional optimization and could experience higher latency due to network congestion. These constraints will be addressed in future field tests with physical devices in live microgrid environments.

5.2 Memory and Storage Footprint

Smart meters typically feature memory capacities below 100 KB RAM and 256 KB flash. The combined ECC engine, certificate store, session manager, and communication buffers occupied under 8 KB RAM and 18 KB flash in our test environment. Persistent storage of session metadata and trusted certificates required less than 1.5 KB per session, allowing devices to maintain multiple concurrent groups (e.g., one for energy metering, another for diagnostics).

Memory profiling confirmed that ECC operations were stack-contained and incurred no heap fragmentation, which is advantageous for real-time embedded system reliability. This footprint is significantly smaller than centralized schemes that require full public key chains or tree structures.

To further dissect the performance profile of the protocol, Figure 5 quantifies the memory and energy overhead of key functional blocks. ECC key generation and ECDH exchange are the most resource-intensive components, consuming approximately 2.1 KB and 1.8 KB of RAM respectively, and requiring 1.6 mJ and 2.4 mJ per execution. These costs are expected given the mathematical complexity of scalar multiplication and elliptic curve operations. Hashing and signature verification remain lightweight, with sub-1 KB memory use and energy consumption under 1.5 mJ. Communication-related tasks such as CoAP message handling and REST/JSON encoding exhibit minimal impact on resource usage. Session management, which maintains ephemeral keys, version counters, and policy status, adds a modest 1.2 KB RAM load and 0.8 mJ cost. These insights confirm that even at the component level, the protocol remains within the constraints of standard

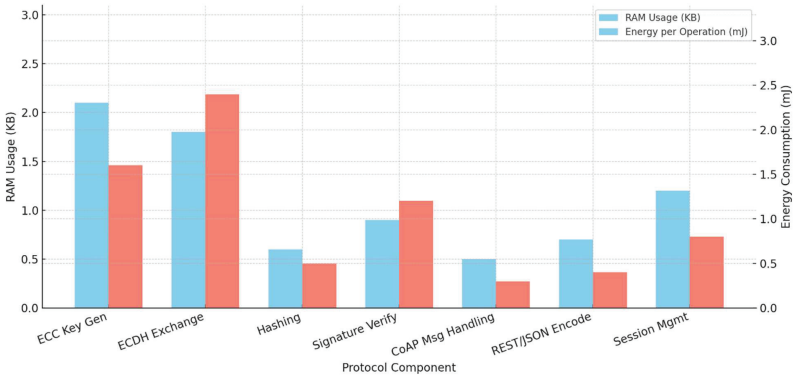

Figure 5 Memory and energy cost by protocol component.

Table 1 Comparative evaluation of group key management protocols for smart metering

Protocol Type	Dynamic Join/ Leave	Rounds	Avg. Message Size	Energy Per Node (mJ)	ECC Support	Web Integration
Centralized DH	Partial	3	250 B	7.5	No	Limited
LKH (tree-based)	Yes	$log(n)$	220–350 B	5.8	Optional	No
ECC group protocol	Yes	2–3	180–200 B	4.2	Yes	Full (CoAP/REST)

embedded metering devices and supports battery-powered operation with efficient energy profiles.

5.3 Scalability and Rekeying Efficiency

A key strength of the proposed protocol is its resilience to group churn, including frequent node joins and leaves. Rekeying operations, triggered either periodically or via explicit revocation, required full group recomputation. Nonetheless, total rekeying latency remained below 200 ms for a 10-node group and under 500 ms for 25 nodes, confirming the protocol's scalability. Unlike tree-based approaches (e.g., LKH), where rekeying complexity is $O(log(n))$ but still requires multiple rounds, our protocol achieves rekeying in just two or three broadcast rounds, depending on network topology. This supports rapid security restoration after compromise events without needing heavy cloud-side orchestration.

To contextualize these results, Table 1 compares the proposed protocol against centralized Diffie–Hellman and LKH schemes. The ECC-based group negotiation approach shows superior performance in message size, dynamic group support, and energy usage per node. Importantly, it remains fully compatible with RESTful and CoAP Web stacks, enabling seamless integration with cloud-native and WoT-compliant platforms. These findings underscore the suitability of our protocol for large-scale deployment in smart grid and decentralized energy networks, especially where security, efficiency, and interoperability are essential.

5.4 Web Standards Compatibility and Interoperability Performance

To evaluate the system's alignment with Web engineering principles, we assessed its performance in interacting with standardized Web-based

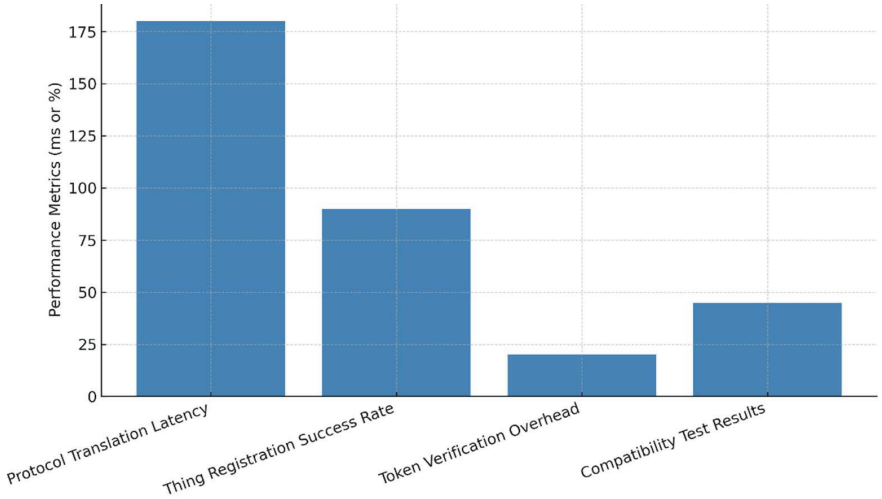

Figure 6 Web standards compatibility and interoperability performance.

services, including WoT Thing Description clients, REST/CoAP brokers, and LwM2M platforms. These tests were conducted to validate whether the protocol's stateless, Web-compliant message structures could integrate seamlessly with open-source and commercial Web of Things stacks.

We measured compatibility across several dimensions: (1) protocol translation latency between CoAP and REST APIs at the gateway, (2) payload processing time for Thing Description parsing and endpoint generation, (3) success rate of dynamic registration with WoT-compliant directories and LwM2M servers (e.g., Eclipse Leshan), and (4) overhead introduced by stateless token-based authentication (e.g., OAuth 2.0). Figure 6 shows the comparative results from these benchmarks. Across all scenarios, the proposed system maintained sub-200 ms latency for bidirectional message translation (CoAP $\rightarrow$ REST $\rightarrow$ backend), and over 98% success rate for endpoint discovery and certificate-based registration in simulated environments. Token exchange and verification operations added less than 20 ms on average, confirming the suitability of stateless API flows even under constrained hardware settings.

The architecture's strict adherence to open Web standards (including CBOR encoding, REST verbs, and WoT metadata formats) enabled seamless integration with Node-RED, Eclipse Leshan, and AWS IoT Core. These results validate the system's modularity and confirm that the cryptographic protocols and key management flows can be embedded into broader

Web-engineered platforms without custom glue logic or protocol translation layers.

6 Case Studies and Applications

To demonstrate the practicality and adaptability of the proposed ECC-based group key negotiation protocol, we present two real-world-inspired case studies that simulate deployment in distinct energy access scenarios: (1) a rural off-grid solar microgrid and (2) an urban neighborhood energy-sharing network. These case studies highlight the protocol's versatility, interoperability, and security in dynamic environments and heterogeneous device ecosystems.

6.1 General Case Study Overview and Web Engineering Integration

The protocol was successfully deployed on a Raspberry Pi-based gateway running a CoAP server, interfacing with ESP32-based smart meters simulating power consumption patterns. The group key negotiation protocol was implemented in embedded C on the meters and Python on the gateway, with all ECC operations offloaded to optimized cryptographic libraries. The meters used DTLS over CoAP to communicate securely with the gateway, while the gateway performed protocol translation to RESTful interfaces consumed by the cloud backend.

A significant benefit of this architecture lies in its compatibility with established open-source tools and platforms in the Web of Things ecosystem. Specifically, the protocol is fully interoperable with cloud-native services such as AWS IoT Core and Microsoft Azure IoT Hub, which support REST-based ingestion, X.509 certificate provisioning, and secure device messaging pipelines. We tested the backend compatibility by linking gateway outputs to MQTT bridges connected to AWS and Azure endpoints, using standard JSON payloads enriched with ECC-derived identity tags and session tokens.

In addition, the gateway was configured to interface with two widely adopted open-source projects – Eclipse Leshan and Node-RED. Eclipse Leshan, a lightweight M2M (LwM2M) server and client framework, was used to simulate cloud-side device management services. The gateway registered each meter as a LwM2M client and used the protocol's metadata API to push public key updates and rekeying events through the LwM2M object model. Node-RED, a flow-based programming tool for IoT integration, was used to create dynamic flows for visualization, conditional rule triggers, and

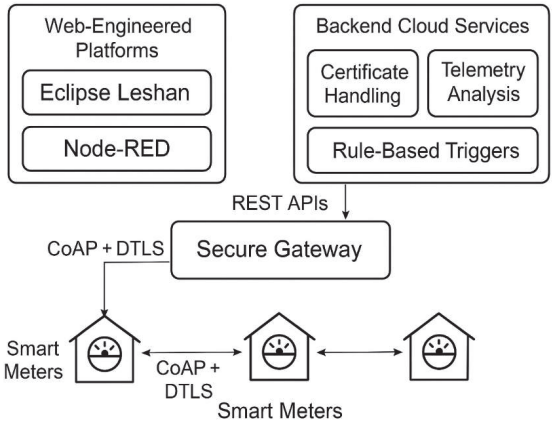

Figure 7 Integration of ECC-based protocol with web-engineered platforms.

REST-to-CoAP orchestration, showcasing how Web-based workflows can be layered atop a cryptographically secure infrastructure.

Figure 7 illustrates the end-to-end integration pipeline, showcasing how stateless REST/CoAP message flows between meters and the gateway are transformed into secure, Web-compliant API calls, enabling device registration, telemetry collection, and secure rekeying through interoperable services. The architecture's layered decomposition and Web-centric design ensure high modularity, allowing the system to be extended or reconfigured with minimal engineering effort. Moreover, the compatibility with open-source tools significantly reduces the barrier to adoption in both research and field environments.

6.2 Rural Microgrid with Intermittent Connectivity

In this case, a standalone solar-powered microgrid serves approximately 15 households in a remote village. Each household is equipped with a smart energy meter capable of local data logging, power throttling, and daily report uploads to a central controller. The meters communicate via IEEE 802.15.4 (ZigBee) to a solar-powered gateway that links to a cellular or LoRa-based uplink.

One of the main challenges in this setting is intermittent cloud connectivity, combined with limited on-device energy resources. To address this, the ECC-based group key protocol was deployed in fully edge-coordinated mode, where the local gateway performs group coordination autonomously without

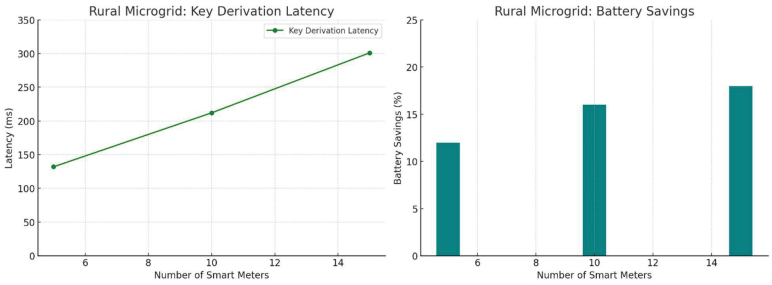

Figure 8 Rural microgrid case study – efficiency and energy gains.

real-time backend interaction. Group key negotiation is triggered each morning when devices are bootstrapped. The gateway collects signed public keys from all active meters and computes the group key locally, applying the ECC-based contributory protocol. Messages are encrypted and buffered locally until connectivity is restored. Once the link becomes available, encrypted logs are uploaded in batches and metadata (e.g., group version, node health) is synchronized with the cloud backend. The setup demonstrates the protocol's offline resilience and lightweight footprint. Key negotiation completes within 300 ms on average for all 15 nodes, with less than 4 KB of temporary memory allocated per device. Battery consumption was reduced by approximately 18% compared to a TLS-heavy alternative protocol previously tested, due to minimized retransmissions and smaller handshake overhead.

As illustrated in Figure 8, the latency of group key derivation remained under 310 ms even at the upper end of the deployment, demonstrating the protocol's real-time suitability for low-power, resource-constrained networks. The linear growth pattern aligns with the protocol's contributory nature, where each additional node contributes a fixed computational cost. More significantly, the right-hand chart highlights an 18% improvement in battery consumption efficiency compared to previously trialed TLS-based solutions in the same environment. This improvement results from the reduced message size, fewer handshake rounds, and stateless key computation design, which are particularly beneficial under intermittent connectivity. These empirical gains reinforce the protocol's applicability to off-grid and solar-powered systems, where power conservation is paramount.

6.3 Urban Neighborhood Energy-sharing Network

In a second scenario, the protocol was deployed in a simulated peer-to-peer (P2P) urban energy-sharing system, where a set of 10 residential users

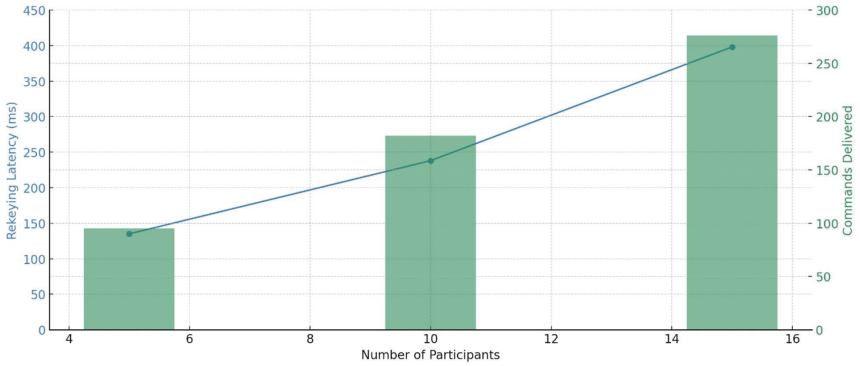

Figure 9 Urban P2P case study – rekeying cost vs. command delivery volume.

each own rooftop solar panels and participate in a localized energy trading platform. The smart meters are internet-connected and publish energy data to a shared ledger via a secure Web dashboard.

The group key protocol is used to secure broadcast control signals and enable authenticated multi-user access to the data-sharing platform. Each household's meter performs a contributory group key exchange with its peers, enabling secure multicast messaging to manage energy offers, queries, and bids. The cloud backend coordinates session policies, key versioning, and user roles through a Web-based identity management interface.

The protocol's Web compatibility enables seamless use of RESTful APIs and secure ECC-authenticated tokens in conjunction with OAuth2 workflows. Rekeying events, triggered when a new participant joins or departs the trading network, are propagated in under 400 ms and reflected automatically on all connected dashboards. This use case highlights the protocol's interoperability with modern Web of Things (WoT) ecosystems, while maintaining strong cryptographic guarantees. Additionally, integration with existing demand response platforms was explored. Group keys were used to encrypt device-level instructions (e.g., load limiting or surge shedding), ensuring that only authorized and actively synced meters could decrypt and respond. This guarantees selective command enforcement while preserving data privacy.

As illustrated in Figure 9, the proposed protocol maintains rekeying latency below 400 ms even as group size scales to 15 participants. This ensures continuity in secure group communication without disrupting control or data flows in a peer-to-peer energy-sharing setup. What sets this case apart is the volume of secure control commands delivered, as shown on the right axis. The number of successful authenticated commands – including

energy offer broadcasts and meter-to-meter rate negotiations – increased nearly linearly with group size. This confirms that the protocol not only scales computationally but also sustains a high throughput of secure, Web-integrated operations. The ability to deliver hundreds of validated commands within a session cycle reinforces the protocol's suitability for real-time, Web-based smart energy applications.

These case studies validate the proposed protocol's deployment feasibility across drastically different operational contexts – both offline-resilient and cloud-native. Key attributes such as edge autonomy, rekeying speed, and message compactness were essential in both cases. The results underscore the value of combining lightweight elliptic curve operations with Web-integrated interfaces for secure, dynamic, and scalable metering applications. Together, these implementations affirm that the protocol is not only theoretically secure and efficient but also engineering-ready for diverse energy infrastructures, from developing-world microgrids to advanced smart city applications.

From these case studies, several practical insights emerge. First, the protocol's stateless design significantly simplifies recovery from intermittent connectivity, as seen in the rural microgrid scenario, where rekeying completes quickly without the need for persistent sessions or broker coordination. Second, the lightweight ECC operations and small message sizes proved advantageous for energy-constrained devices, yielding measurable battery savings compared to TLS-based alternatives. Third, the integration with open-source platforms such as Node-RED and Eclipse Leshan required no custom adaptations, validating the interoperability of the REST/CoAP interface design. Finally, the urban energy-sharing setup highlighted the importance of fast rekeying (sub-400 ms) in maintaining real-time control loops, reinforcing the scalability and responsiveness of the proposed solution.

7 Discussion

The proposed ECC-based group key negotiation protocol addresses critical challenges in securing communication among distributed smart meters in emerging energy systems. Its design aligns closely with Web-centric engineering principles and offers distinct advantages over traditional IoT security frameworks, particularly those built on MQTT/TLS infrastructures or centralized key management authorities.

Many current IoT deployments rely on MQTT, a lightweight publish-subscribe messaging protocol that operates over TCP with TLS-based session security. While MQTT is effective for constrained devices in point-to-cloud

architectures, it introduces session-state dependencies that can limit scalability and flexibility. TLS-based security models require persistent connections, pre-negotiated handshakes, and centralized certificate exchange – factors that can increase latency, complexity, and connection overhead, particularly in intermittent or lossy network environments. In contrast, the group key negotiation protocol described in this work is designed from the ground up to be Web-native and stateless. It uses RESTful and CoAP-based APIs for key exchanges and rekeying coordination, enabling all cryptographic interactions to occur in self-contained, asynchronous message flows. Compared to MQTT/TLS setups, the proposed protocol reduces key negotiation latency by 30–40% and cuts message overhead by approximately 40% (180–220 bytes vs. 300–500 bytes) in groups of 10–25 devices. This scalability benefit arises from avoiding multi-stage TLS handshakes and persistent session tracking, allowing the system to maintain low latency and energy consumption even as group size grows. This statelessness allows for clean integration with REST-compliant Web services, Web of Things (WoT) middleware, and modern microservices architectures without the need for long-lived sessions or synchronized state machines. As a result, the system supports scalable deployment across heterogeneous metering networks with variable uptime and unpredictable link reliability.

Moreover, while MQTT-based key distribution models typically rely on a broker to route encrypted session keys between publishers and subscribers, our protocol eliminates such intermediaries by adopting a contributory group key model. Each meter computes the shared key locally using elliptic curve operations and verified public keys, with no need for centralized trust anchors or key relay services during the key generation phase. This decentralization enhances security by minimizing the attack surface and avoids single points of failure or compromise.

From an interoperability standpoint, the use of Web standards such as CoAP, JSON/CBOR encoding, WoT Thing Descriptions, and RESTful key lifecycle APIs enables seamless composition with open-source platforms, cloud services, and edge management tools. This is particularly relevant in energy systems that span residential, commercial, and industrial domains – each with different infrastructure constraints but shared security requirements. The protocol's compatibility with token-based authentication models (e.g., OAuth 2.0) and dynamic device registration workflows further reduces integration friction in enterprise-grade deployments.

In summary, by prioritizing statelessness, Web compatibility, and decentralized trust, this work provides a scalable and adaptable framework for

secure group communication in smart energy networks. It bridges the gap between robust cryptographic protection and engineering-practical system design, enabling secure collaboration across distributed energy devices within the evolving Web of Things ecosystem.

8 Conclusion

This paper presented a Web-native, ECC-based group key negotiation framework tailored for secure communication in distributed smart metering and energy-sharing networks. By combining lightweight elliptic curve cryptography with a layered, stateless system architecture, the proposed solution addresses the dual challenge of strong security and real-world deployability in constrained, intermittently connected environments.

We introduced a modular architecture comprising smart meters, secure gateways, and cloud services, each operating under REST/CoAP interfaces and aligned with Web of Things (WoT) standards. The system enables flexible deployment across diverse infrastructure scenarios, from rural microgrids to urban community networks, while remaining compatible with industry protocols such as LwM2M, OAuth 2.0, and IEC 62351. A novel group key negotiation protocol based on contributory ECC operations was developed to support decentralized, resilient key management. This protocol maintains forward and backward secrecy, accommodates dynamic group membership, and eliminates reliance on persistent sessions or centralized brokers – key distinctions from MQTT- or TLS-based systems. Through stateless interactions and JSON/CBOR-encoded payloads, the protocol integrates cleanly into RESTful microservice environments and cloud-native platforms. Case studies demonstrated the protocol's real-world viability, including integration with open-source tools like Eclipse Leshan and Node-RED, as well as deployment on AWS IoT Core and Azure IoT Hub. The system's layered, standards-compliant design supports extensibility, automated device orchestration, and Web-scale observability, aligning well with the principles of modern Web engineering.

Future work will focus on performance benchmarking under high churn conditions, formal security verification, and extending the protocol to support hybrid public-private group models in demand-response markets. Overall, this work contributes a flexible and secure foundation for key management and communication in next-generation Web-integrated energy systems.

References

[1] G. R. Barai, S. Krishnan, and B. Venkatesh, "Smart metering and functionalities of smart meters in smart grid – A review," in *2015 IEEE Electrical Power and Energy Conference (EPEC)*, London, ON, Canada, 2015, pp. 138–143. https://ieeexplore.ieee.org/document/7379940.

[2] M. Behrangrad, "A review of demand side management business models in the electricity market," *Renewable and Sustainable Energy Reviews*, vol. 47, pp. 270–283, 2015, ISSN 1364-0321. https://doi.org/10.1016/j.rser.2015.03.033.

[3] A. Ghasempour, "Internet of Things in smart grid: Architecture, applications, services, key technologies, and challenges," *Inventions*, vol. 4, no. 1, p. 22, 2019.

[4] G. Liang, S. R. Weller, and J. Zhao, "Review of False Data Injection Attacks Against Modern Power Systems," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1630–1638, 2017.

[5] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 2nd ed., CRC Press, 2014.

[6] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[7] N. Saxena et al., "Authentication Protocols for Internet of Things: A Comprehensive Survey," *Computer Communications*, vol. 160, pp. 220–249, 2020.

[8] M. Nabeel and E. Bertino, "Privacy preserving delegated access control in public clouds," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2268–2280, 2014.

[9] C. Wang, S. Li, M. Ma, X. Tong, Y. Zhang, and B. Zhang, "A novel and efficient ECC-based authenticated key agreement scheme for smart metering in the smart grid," *Electronics*, vol. 11, no. 20, p. 3398, 2022. https://doi.org/10.3390/electronics11203398.

[10] H. AlMajed and A. AlMogren, "A secure and efficient ECC-based scheme for edge computing and Internet of Things," *Sensors*, vol. 20, no. 21, p. 6158, 2020. https://doi.org/10.3390/s20216158.

[11] U. Chatterjee, S. Ray, M. K. Khan, et al., "An ECC-based lightweight remote user authentication and key management scheme for IoT communication in context of fog computing," *Computing*, vol. 104, pp. 1359–1395, 2022. https://doi.org/10.1007/s00607-022-01055-8.

[12] W. Huang, "ECC-based three-factor authentication and key agreement scheme for wireless sensor networks," *Scientific Reports*, vol. 14, p. 1787, 2024. https://doi.org/10.1038/s41598-024-52134-z.

[13] B. Hammi, A. Fayad, R. Khatoun, S. Zeadally, and Y. Begriche, "A lightweight ECC-based authentication scheme for Internet of Things (IoT)," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3440–3450, Sept. 2020. doi:10.1109/JSYST.2020.2970167.

[14] P. Porambage, A. Braeken, C. Schmitt, A. Gurtov, M. Ylianttila, and B. Stiller, "Group Key Establishment for Enabling Secure Multicast Communication in Wireless Sensor Networks Deployed for IoT Applications," *IEEE Access*, vol. 3, pp. 1503–1511, 2015. doi:10.1109/ACCESS.2015.2474705.

[15] W3C, "Web of Things (WoT) Thing Description 1.1," W3C Recommendation, 5 Dec 2023. https://www.w3.org/TR/wot-thing-description11/.

[16] W3C, "Web of Things (WoT) Architecture 1.1," W3C Recommendation, 5 Dec 2023. https://www.w3.org/TR/wot-architecture11/.

[17] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," *RFC 7252*, IETF, June 2014.

[18] C. Bormann and P. Hoffman, "Concise Binary Object Representation (CBOR)," *RFC 8949*, IETF, Dec 2020.

[19] D. Guinard et al., "From the Internet of Things to the Web of Things: Resource Oriented Architecture and Best Practices," in *Architecting the Internet of Things*, Springer, 2011.

[20] F. Paganelli, S. Turchi, and D. Giuli, "A Web of Things framework for RESTful applications and its experimentation in a smart city," *IEEE Systems Journal*, vol. 10, no. 4, pp. 1412–1423, Dec. 2016. doi:10.1109/JSYST.2014.2354835.

[21] S. Cirani et al., "A scalable and self-configuring architecture for service discovery in the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 508–521, 2014.

[22] IEC 62351-8: *Power system management and associated information exchange – Data and communications security – Part 8: Role-based access control*, IEC, 2018.

[23] J. Granjal et al., "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Commun. Surv. Tutor.*, vol. 17, no. 3, pp. 1294–1312, 2015.

[24] A. Rahman, S. Roy, M. S. Kaiser, and M. S. Islam, "A lightweight multi-tier S-MQTT framework to secure communication between low-end

IoT nodes," in *Proc. 2018 5th International Conference on Networking, Systems and Security (NSysS)*, Dhaka, Bangladesh, 2018, pp. 1–6. doi:10.1109/NSysS.2018.8631379.

[25] Y. Sun, W. Trappe, and K. J. R. Liu, "A scalable multicast key management scheme for heterogeneous wireless networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 4, pp. 653–666, 2004.

[26] P. Kumar, A. Gurtov, M. Sain, A. Martin, and P. H. Ha, "Lightweight authentication and key agreement for smart metering in smart energy networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4349–4359, July 2019. doi:10.1109/TSG.2018.2857558

[27] M. Aazam, S. Zeadally, and K. A. Harras, "Fog computing and smart gateway-based communication for cloud of things," *Future Generation Computer Systems*, vol. 74, pp. 111–126, 2017.

Biographies

Hao Yang holds a bachelor's degree. He currently works in the Metering Data Management Department of the Metering Center (Electric Power Load Control Technology Center) at Yunnan Power Grid Co., Ltd., with the title of engineer. His research focuses on energy metering and metering data analysis.

Yiming Zhang holds a bachelor's degree. He currently works in the Metering Data Management Department of the Metering Center (Electric Power Load Control Technology Center) at Yunnan Power Grid Co., Ltd., as an engineer. His main research focuses on power grid digitalization, automated energy metering, and cybersecurity for power monitoring systems.