Journal of *Cybersecurity and Privacy*

MDPI

Article

A Secure and Scalable Authentication and Communication Protocol for Smart Grids

Muhammad Asfand Hafeez [1], Kazi Hassan Shakib [2] and Arslan Munir [1,*]

[1] Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA; mhafeez2024@fau.edu
[2] Department of Computer Science, Kansas State University, Manhattan, KS 66506, USA; kshakib@ksu.edu
* Correspondence: arslanm@fau.edu

Abstract: The growing adoption of smart grid systems presents significant advancements in the efficiency of energy distribution, along with enhanced monitoring and control capabilities. However, the interconnected and distributed nature of these systems also introduces critical security vulnerabilities that must be addressed. This study proposes a secure communication protocol specifically designed for smart grid environments, focusing on authentication, secret key establishment, symmetric encryption, and hash-based message authentication to provide confidentiality and integrity for communication in smart grid environments. The proposed protocol employs the Elliptic Curve Digital Signature Algorithm (ECDSA) for authentication, Elliptic Curve Diffie–Hellman (ECDH) for secure key exchange, and Advanced Encryption Standard 256 (AES-256) encryption to protect data transmissions. The protocol follows a structured sequence: (1) *authentication*—verifying smart grid devices using digital signatures; (2) *key establishment*—generating and securely exchanging cryptographic keys; and (3) *secure communication*—encrypting and transmitting/receiving data. An experimental framework has been established to evaluate the protocol's performance under realistic operational conditions, assessing metrics such as time, throughput, power, and failure recovery. The experimental results show that the protocol completes one server–client request in 3.469 ms for a desktop client and 41.14 ms for a microcontroller client and achieves a throughput of 288.27 requests/s and 24.30 requests/s, respectively. Furthermore, the average power consumed by the protocol is 37.77 watts. The results also show that the proposed protocol is able to recover from transient network disruptions and sustain secure communication.

Keywords: public key cryptography; smart grid; cryptographic protocol; key establishment; authentication

Academic Editor: Danda B. Rawat

Received: 31 December 2024
Revised: 14 March 2025
Accepted: 17 March 2025
Published: 21 March 2025

Citation: Hafeez, M.A.; Shakib, K.H.; Munir, A. A Secure and Scalable Authentication and Communication Protocol for Smart Grids. *J. Cybersecur. Priv.* **2025**, *5*, 11. https://doi.org/10.3390/jcp5020011

1. Introduction

The evolution of modern smart grids has marked a significant advancement in energy management systems, characterized by notable improvements in the efficiency, reliability, and adaptability of electricity distribution [1]. Unlike traditional electrical systems, which operate on a centralized and one-way communication model (i.e., from supervisory control and data acquisition (SCADA) to other grid nodes), smart grids are defined by their two-way communication capabilities (i.e., between SCADA and other grid nodes) and the integration of distributed generation sources. This contemporary framework utilizes a diverse range of advanced technologies, including smart meters for real-time data acquisition, phasor measurement units (PMUs) for accurate monitoring of electrical waveforms,

and phasor data concentrators (PDCs) to aggregate data from PMUs and to facilitate data sharing with SCADA.

Smart grids additionally incorporate distributed energy resources (DERs) [2] such as solar panels, wind turbines, and electric vehicle (EV) chargers, which support the expanding EV market. Advanced supervisory systems, including Advanced Metering Infrastructure (AMI) [3], Supervisory Control and Data Acquisition (SCADA) [4], and Advanced Distribution Management Systems (ADMS) [5], are also integral to enhancing dynamic grid management. This interconnected ecosystem promotes enhanced operational efficiency and optimizes electricity distribution, contributing to a more sustainable approach to energy consumption.

These advancements in smart grids require robust communication protocols to protect the integrity and reliability of grid operations. The integration of Internet of Things (IoT) devices into smart grids presents transformative opportunities to enhance grid intelligence, reliability, and overall efficiency [6]. Simultaneously, this integration introduces significant cybersecurity challenges [7–9]. The communication among devices and supervisory-level data flows within systems such as AMI, SCADA, and Industrial Control Systems (ICSs) are increasingly susceptible to interception and manipulation by malicious actors. Existing standards for smart grid communication, including IEEE C37.118.2 [10] and IEC 61850-90-5 [11], do not specify node authentication, secret key establishment, and secure communication mechanisms, consequently leaving the power grid vulnerable to various cybersecurity threats, including unauthorized access and data tampering. These vulnerabilities not only jeopardize operational safety but also raise significant concerns regarding the overall security of electrical infrastructure.

The secure operation of smart grids is dependent upon provision of various security services, including (i) authentication, which verifies the authenticity of smart grid nodes, service providers (SPs), and other users to prevent impersonation, replay, or insider attacks; (ii) integrity, which protects transmitted PMU/smart meter measurements and readings from unauthorized alteration or tampering; (iii) confidentiality, which ensures that sensitive information, such as energy consumption, generation data, and operational commands, remains confidential during transmission; (iv) non-repudiation, which prevents any party engaged in data exchange from denying actions conducted; (v) authorization mechanisms, which provide access control and must implement role-based access controls to restrict system privileges; and (vi) anonymity, which is essential to protect consumer identity and prevent data tracking. Security safeguards must ensure that data extracted from compromised devices do not compromise the security of the overarching SCADA/AMI system. The provision of these security services explicitly mitigates threats such as man-in-the-middle (MITM) attacks and denial-of-service (DoS) [12] interruptions, thus ensuring the resilience and reliability of smart grid operations.

This study addresses the critical need for a secure communication protocol specifically designed for smart grid environments. The protocol incorporates a robust authentication method utilizing the Elliptic Curve Digital Signature Algorithm (ECDSA) [13] and Elliptic Curve Diffie–Hellman (ECDH) [14] for key establishment and AES [15] for encryption and decryption. It also emphasizes scalability and resilience against potential cybersecurity threats. The primary contributions of this research are as follows:

1. Development of a secure and lightweight cryptographic protocol that addresses the evolving needs of smart grid infrastructure while ensuring strong protection against potential security threats, thereby maintaining the integrity and reliability of smart grids.

2. Efficient implementation of the proposed protocol across diverse platforms to ensure a thorough evaluation of its performance for different smart grid nodes with varying computation capabilities.
3. Experimental evaluation of performance, scalability, and security of the proposed protocol under real-world conditions. Our evaluation includes testing across both high-performance and resource-constrained devices and assessing resilience against cyber threats to ensure the robustness and reliability of the protocol for effective deployment in smart grid systems.

The structure of this paper is organized as follows: Section 2 provides the necessary background for the proposed study and reviews relevant literature. Section 3 outlines the design of the proposed cryptographic protocol. In Section 4, the development of the experimental setup is detailed. Section 5 presents the experimental results, while Section 6 concludes this paper.

2. Background

2.1. Vulnerabilities of Smart Grid Infrastracture

The transition from traditional grids to smart grids addresses the increasing demand for reliable, efficient, and sustainable energy solutions [16]. Traditional grids, operating as one-directional systems, were limited in communication, control, and their ability to integrate renewable energy, manage peak demand, and prevent outages. In contrast, smart grids enable real-time monitoring, bi-directional energy flow, and enhanced grid management, creating a dynamic and resilient energy ecosystem [17].

Smart grids integrate electricity generation, transmission, distribution, and consumption, as shown in Figure 1, through a layered architecture of physical infrastructure, communication networks, and control systems. Power generation incorporates traditional plants, renewable energy sources, and distributed generation (DG) resources like rooftop solar panels and microturbines. While DG and renewables enhance grid flexibility and reduce reliance on fossil fuels [18], they necessitate secure communication protocols to ensure reliable data exchange and prevent unauthorized access or manipulation. The transmission layer, comprising high-voltage lines and substations, ensures efficient long-distance electricity transport. Systems like wide-area measurement systems (WAMSs) [19] and PMUs enhance grid stability analysis but rely on communication protocols such as IEC 60870-5-104 and IEEE C37.118.2 [20], which are vulnerable to cyberattacks like MITM attacks, compromising operational integrity [21].

The distribution network connects electricity to end-users through substations, transformers, and smart meters. AMI supports real-time data exchange and dynamic load management, enhancing efficiency but exposing the system to risks such as data interception, manipulation, and denial-of-service attacks [22]. This vulnerability is further increased by the integration of legacy power systems with newer smart grid technologies, such as renewable energy resources. Legacy devices, lacking contemporary security features, introduce additional risks due to their inability to support modern cryptographic protocols [23]. Simultaneously, the adoption of smart meters, while simplifying operations and enabling precise monitoring, presents its own set of challenges, including risks to data confidentiality and user privacy [24].

At the consumer level, devices like smart meters, energy storage systems, and EV chargers facilitate renewable integration and demand-side management. However, these advancements introduce new attack surfaces, with challenges such as weak authentication mechanisms in smart meters and neighborhood area network (NAN) gateways, leaving systems vulnerable to threats like key compromise impersonation (KCI) attacks [25].

Supporting systems such as SCADA, ADMS [26], and distributed energy resource management systems (DERMSs) [27] coordinate grid operations using devices like PMUs and smart meters. Yet, traditional data aggregation schemes in smart grids often rely on fully trusted authorities, posing significant security risks. This reliance makes them susceptible to malicious behavior or collusion by control centers and gateways, enabling adversaries to intercept, modify, or delete messages exchanged over public channels.

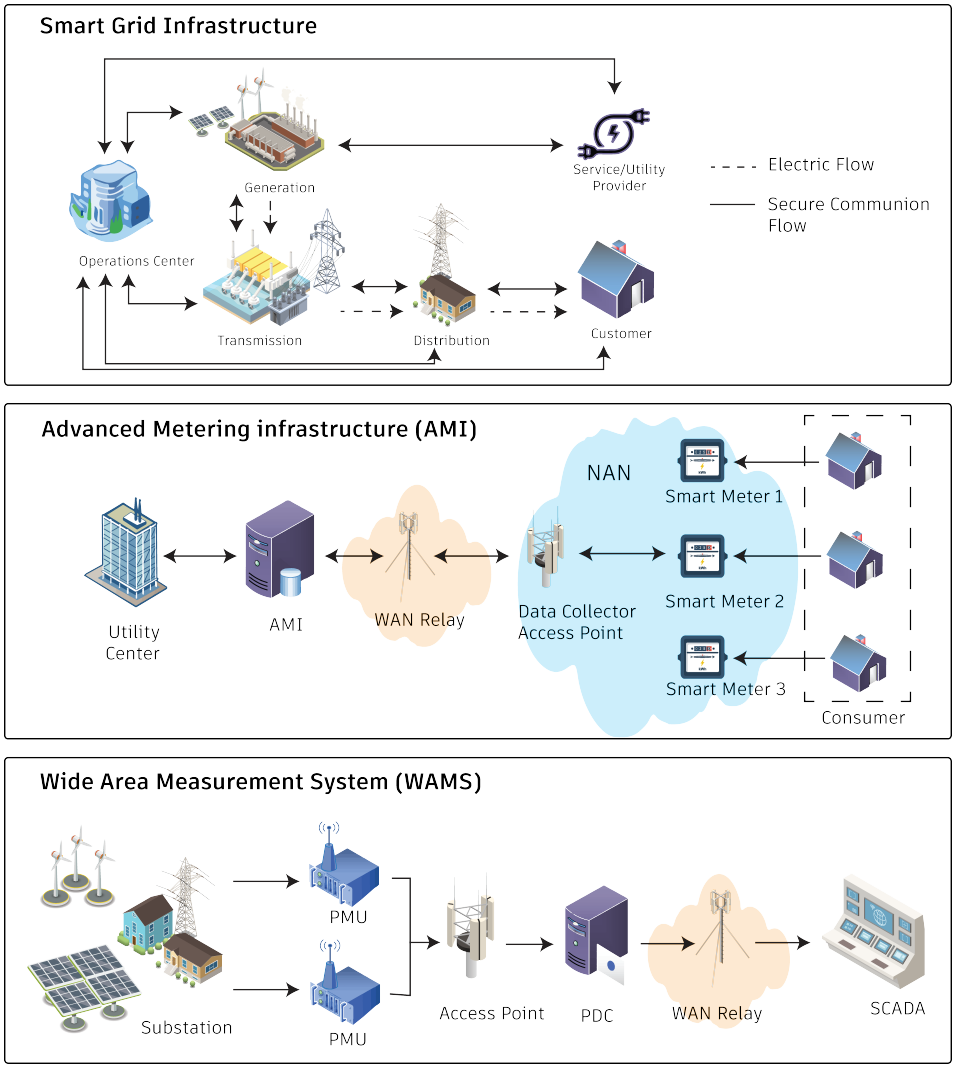

Figure 1. Overview of smart grid's secure communication between nodes.

Despite the numerous benefits of smart grid technology, its interconnected and cyber-physical nature introduces vulnerabilities. Key entry points for attackers include diagnostic ports on smart meters, wireless data transmission channels, EV charging stations, and SCADA/AMI head-end servers [28]. These components are attractive targets for cyberattacks aiming to disrupt operations or compromise data integrity, underscoring the urgent need for robust and secure cryptographic solutions for smart grid systems.

2.2. Related Work

The interconnected networks in smart grid technologies continue to be significantly susceptible to a range of sophisticated cyberattacks, including DoS, replay Attacks (RA), time delay attacks (TDA), false data injection attacks (FDIA), malware attacks, and even social engineering. The ramifications of these threats can encompass economic losses, data breaches, and interruptions to essential services.

Throughout the years, numerous high-profile incidents have underscored the catastrophic potential of cyberattacks on smart grids. In 2007, the Aurora attack [29], conducted by the Idaho National Laboratory, illustrated how malicious software could manipulate grid control systems, culminating in the severe failure of a power generator. In 2010, the infamous Stuxnet worm attack [30] compromised Iranian programmable logic controllers (PLCs) through a USB flash drive, exploiting vulnerabilities to alter SCADA commands and inflicting considerable damage on the nuclear program. The Dragonfly 2.0 [31] campaign in 2014 specifically targeted energy companies, employing spear-phishing techniques and malware-infected websites to gain unauthorized access to critical systems. Between 2015 and 2016, the cyberattack on the Ukrainian power grid [32] successfully leveraged spear-phishing tactics to infiltrate SCADA systems, resulting in extensive substation outages. More recently, in 2019, a DoS attack briefly disrupted the U.S. power grid, and in 2022, the Sandworm group utilized the NikoWiper malware strain to assault Ukraine's energy sector. In 2023, a DoS attack targeted Hydro-Quebec [33], rendering the company's website and customer portal inaccessible. These incidents emphasize the urgent necessity for security standards that incorporate advanced authentication, encryption, and real-time monitoring to protect critical infrastructure.

The integration of cryptographic protocols within smart grids is imperative in addressing the escalating risks associated with sophisticated cyberattacks, such as DoS, RA, and FDIA. End-to-end encryption, alongside advanced authentication mechanisms, ensures secure communication and prevents unauthorized access across the vast and interconnected network of the smart grid. Symmetric encryption, renowned for its efficiency, and asymmetric encryption methods, including Rivest–Shamir–Adleman (RSA) [34], elliptic curve cryptography (ECC) [13], and Diffie–Hellman (DH) [35]-based handshakes, are frequently employed in hybrid strategies to optimize security, scalability, and performance. Furthermore, robust authentication protocols enhance security by validating devices and users prior to granting access, while advanced key management frameworks such as advanced key management architecture (ASKMA) and scalable method of cryptographic key (SMOCK), in addition to certificate-based and certificateless encryption methods, offer effective solutions for maintaining high levels of security.

Authentication and key agreement (AKA) protocols aim to balance security and efficiency, ensuring the confidentiality of communication messages while minimizing computational, storage, and communication overheads—an essential requirement for practical deployment, particularly in resource-constrained devices. Since the introduction of an AKA protocol for smart grids by Fouda et al. in 2011 [36], researchers have proposed various protocols to address challenges like resistance to ephemeral secret leakage (ESL) attacks, forward secrecy, and user anonymity. ECC-based AKA protocols are increasingly favored due to their smaller key sizes and equivalent security levels to other asymmetric cryptography techniques. However, many existing protocols, as shown in Table 1, such as those by Nicanfar et al. [37] and Tanveer et al. [38], have been found lacking in areas like users' anonymity and untraceability.

Recent advancements include efforts to design protocols robust against emerging threats while maintaining operational efficiency. For example, Hu et al. [39] introduced an ECC-based authentication and key agreement protocol that addressed the issue of secure smart meter registration over open communication channels, though its reliance on a completely trusted registration center presents a potential point of vulnerability. Similarly, Chai et al. [40] proposed an Shangyong Mima 2 (SM2)-based AKA protocol that offers user anonymity and reduced computational demands. However, its requirement for secure channels during registration restricts its applicability in environments where such channels are unavailable.

Certificateless cryptographic schemes have gained significant attention due to their efficiency and reduced reliance on traditional certificates, thereby lowering key management overhead. Early certificateless public key cryptography (CPKC) models, such as Al-Riyami and Paterson's work in 2003 [41], introduced a framework that eliminates the need for traditional certificate authorities. However, early implementations faced scalability issues and dependency on centralized or semi-centralized key generation centers (KGCs), making them susceptible to single points of failure and trust issues. Recent advancements have explored certificateless cryptographic techniques in smart grid applications. For instance, Liu et al. (2023) [42] proposed a certificateless multi-dimensional data aggregation scheme leveraging Paillier homomorphic encryption within a fog computing architecture. This approach ensures secure data aggregation and key negotiation among users while mitigating collision risks from control centers and fog nodes. Additionally, it enhances user privacy protection against potential threats from malicious KGCs and significantly reduces computational overhead for smart meters and aggregators compared to conventional aggregation techniques.

Table 1. Comparative analysis of cryptographic protocols for smart grid security.

Reference	Cryptographic Approach	Key Features/Advantages	Limitations
Fouda et al. [36]	Authentication and key agreement (AKA) protocol	Early establishment of secure communication channels	Lacks anonymity and scalability.
Nicanfar et al. [37]	ECC-based, X.1035 standard and Diffie–Hellman based protocol	Provides basic authentication and key agreement	Does not ensure user anonymity and untraceability.
Tanveer et al. [38]	Authenticated encryption with associative data (AEAD) primitives used as access control protocol	Efficient protocol design	Limited anonymity and vulnerability to certain attacks.
Hu et al. [39]	ECC-based authentication and key agreement (AKA) protocol	Enables secure registration over open channels; flexible design	Relies on a fully trusted registration center.
Chai et al. [40]	SM2-based authentication and key exchange	Provides user anonymity; exhibits low computational overhead	Requires secure channels during the registration phase.
Al-Riyami & Paterson [41]	Certificateless public key cryptography	Eliminates the need for traditional certificates; efficient key management	Scalability issues; dependency on centralized key generation centers.
Liu et al. [42]	Certificateless aggregation with Paillier encryption	Enables secure data aggregation; reduces computational overhead for smart meters	Potential collision risks and implementation challenges.
Wang et al. [43]	Certificateless with blockchain	Offers decentralization and enhanced transparency	Significant computational overhead; less practical for resource-constrained devices.

Certificateless aggregate signatures (CLASs) have also evolved to improve security and bandwidth efficiency by aggregating signatures from multiple participants. However, existing certificateless methods still face scalability constraints and implementation challenges in real-world smart grid environments. While blockchain-based certificateless architectures [43] offer decentralization and transparency, they impose significant computational overhead, making them less practical for low-power, resource-constrained smart grid devices.

To address the gaps in prior works related to smart grid security, our proposed work introduces a secure and lightweight cryptographic protocol to address the evolving security needs of smart grid infrastructure. The protocol ensures strong protection against potential security threats while maintaining system integrity and reliability. By evaluating the protocol with different computing platforms and benchmarking its performance, we demonstrate the applicability of our proposed protocol to different smart grid nodes.

3. Methodology

This section outlines the proposed cryptographic protocol designed for a smart grid environment. It facilitates interactions between distributed nodes and the central management server. It provides a step-by-step implementation and describes the key components involved in the cryptographic processes.

3.1. System Architecture

The system comprises a certificate authority (CA), a central server, and multiple client nodes, simulating a typical smart grid communication framework as shown in Figure 2. Each client node represents various distributed smart grid elements, including smart meters that monitor energy consumption and EV chargers that support the charging of EVs.

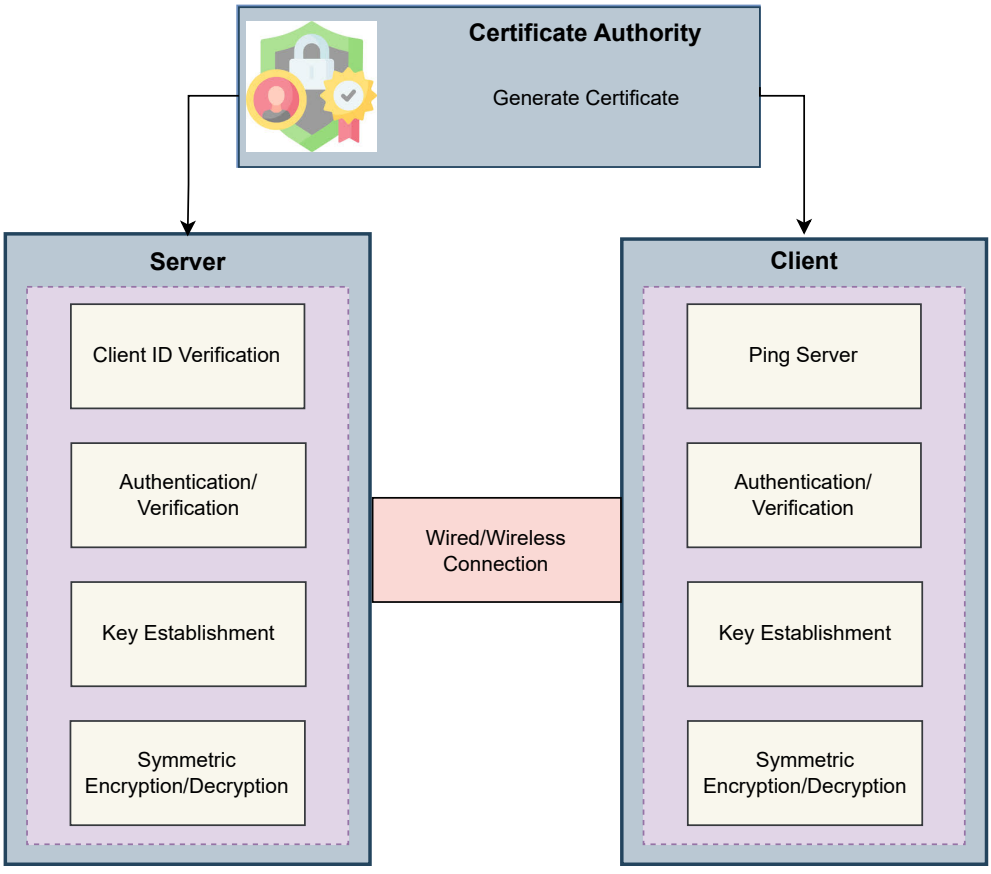

Figure 2. System architecture illustrating the interaction between the certificate authority, server, and client.

The client and server nodes generate their public and private keys and store them in secure files in their respective memory systems. The CA generates certificates for both the serve and the client nodes binding their public keys to their identities. The server and clients use these keys to derive a shared secret during the key exchange phase. The established shared secret can be used to encrypt and decrypt the communication between the server

and clients, which is conducted over a wired or a wireless channel, such as wired/wireless Ethernet or radio frequency or cellular communication (4G, 5G, etc.) or a hybrid channel (i.e., wired and wireless over different intermediary nodes). In our experimental setup, server and client nodes are connected via a wired Ethernet network, and they communicate with each other using the Hypertext Transfer Protocol (HTTP). This architecture offers a simulated environment for real-time data exchange in a smart grid to test and verify the overall functionality of the cryptographic protocol and intelligence of the smart grid.

3.2. Proposed Protocol Design

Figure 3 illustrates the design of the proposed protocol. The proposed protocol establishes a secure communication mechanism between server and client nodes by integrating multiple cryptographic techniques. In the proposed protocol, we have used ECDSA for authentication instead of traditional methods like RSA. The primary reasons for this choice are efficiency and security with smaller key sizes; RSA requires significantly larger key sizes. For example, a 2048-bit RSA key provides equivalent security to a 256-bit ECDSA key. In contrast, ECDSA offers strong security with shorter key lengths, which reduces processing time and memory usage while maintaining robust protection against attacks. The steps of our proposed protocol are outlined as follows:

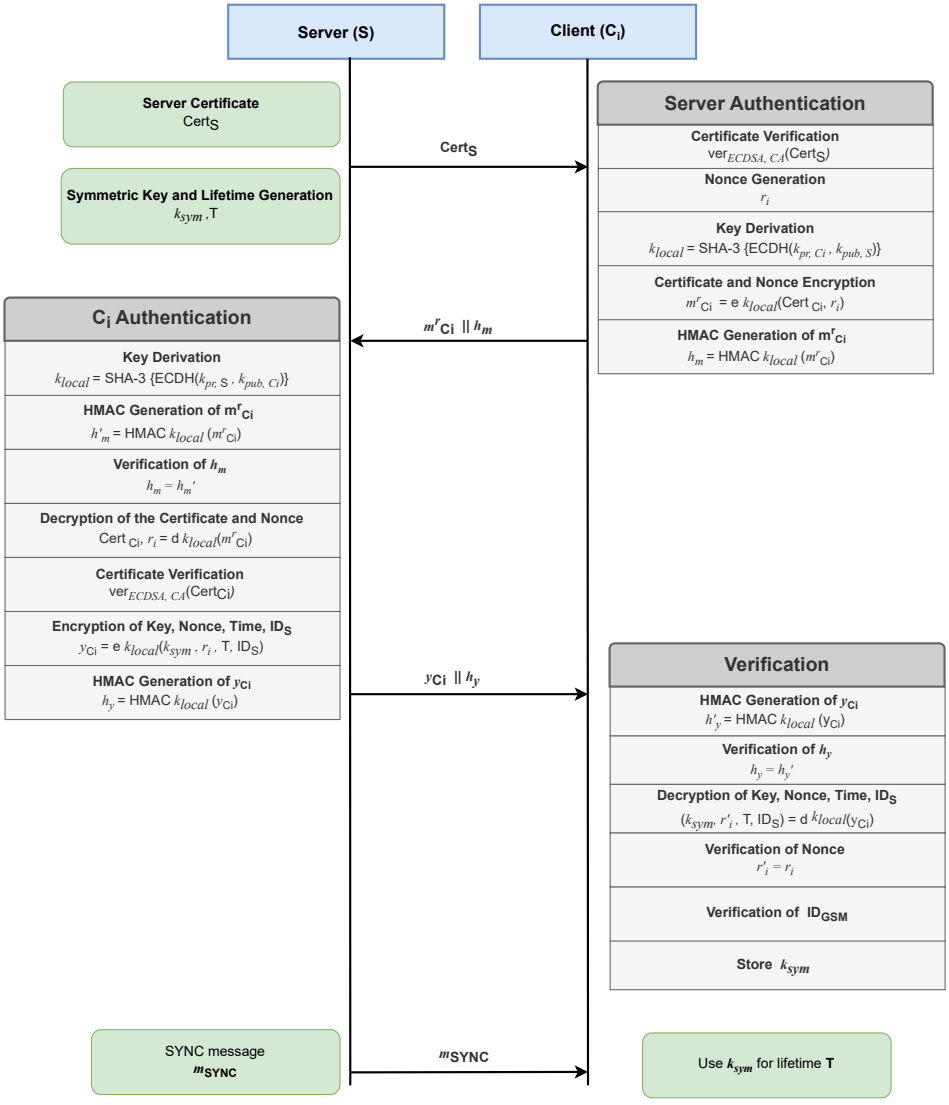

Figure 3. Proposed server–client secure communication protocol.

3.2.1. Certificate Authority (CA) Initialization

The CA initiates the system by generating certificates for both the server and client nodes. The certificate for client is generated by integrating the client's public key k_{pub,C_i}, its identity ID_{C_i}, and the signature S_{C_i}, which is formulated using the CA's private key $k_{pr,CA}$. The server certificate is generated by integrating the server's public key $k_{pub,S}$, its identity ID_S, and the signature S_S, which is created using the CA's private key $k_{pr,CA}$. Additionally, the CA ensures that the certificates are embedded with essential metadata, including expiration dates, to deter reuse or exploitation. These certificates authenticate the server and clients involved in the communication and ensure the integrity of the exchanged data.

3.2.2. Server Authentication

The protocol begins with the client sending a ping (authentication) message to the server, *"Authenticate me"*. The server sends its signed certificate $Cert_S$ to the client. The client verifies the certificate with $k_{pub,S}$ using ECDSA to validate the server's authenticity. Upon successful validation, the client generates a nonce/coin value (r_i) that will be utilized in the protocol to prevent replay attacks. The client then derives a shared secret using ECDH from k_{pr,C_i} and $k_{pub,S}$. The client then generates a shared local key (k_{local}) by applying SHA-3 to the shared secret.

$$\text{Shared Secret} = ECDH(K_{pr,C_i}, K_{pub,S})$$

$$k_{local} = \text{SHA-3}(\text{Shared Secret})$$

Afterwards, the client encrypts its certificate $Cert_{C_i}$ along with the nonce r_i using k_{local} to generate the encrypted message $m^r_{C_i}$ using AES-256 in cipher block chaining (CBC) mode. The client then generates an HMAC $h_m = HMAC_{k_{local}}(m^r_{C_i})$ of the encrypted message. The encrypted message $m^r_{C_i}$ and the corresponding HMAC h_m are transmitted to the server.

3.2.3. Client Authentication

Upon receiving the client's message, the server computes the local key k_{local} using its private key and the client's public key:

$$\text{Shared Secret} = ECDH(k_{pr,S}, k_{pub,C_i})$$

$$k_{local} = \text{SHA-3}(\text{Shared Secret})$$

Using k_{local}, the server verifies the HMAC and decrypts $m^r_{C_i}$ using AES to extract the client's certificate and nonce. The client's certificate is then validated using the CA's public key ($k_{pub,CA}$). The server responds by generating a symmetric key (k_{sym}), assigning it a lifetime T, and encrypting these values, along with the nonce and server's ID (ID_S) using the derived k_{local}:

$$y_{C_i} = e_{k_{local}}(k_{sym}, r_i, T, \text{ID}_S)$$

where $e_{k_{local}}$ denotes the encryption operation with key k_{local}. In our protocol, we have used AES-256 in CBC mode for this encryption. An HMAC (h_y) is also computed of y_{C_i}. The server sends $y_{C_i}||h_y$ to the client.

3.2.4. Symmetric Key Establishment

The client performs verification by generating the HMAC h'_y of the received message y_{C_i} and comparing it with the received HMAC h_y. Following successful verification, the client decrypts the message and verifies the nonce and server ID for authenticity. Upon confirming these elements, the client securely stores k_{sym} for use in future communication.

3.2.5. Secure Communication

Once k_{sym} is established, all further communication is encrypted by a symmetric protocol. In our protocol, we have used AES-256 in cipher block chaining (CBC) mode. For message encryption, the sender uses k_{sym} to encrypt the plaintext message:

$$\text{Ciphertext} = e_{k_{sym}}(\text{Plaintext}).$$

The message integrity of symmetric encryption in our protocol is ensured through an HMAC also generated using k_{sym}.

$$\text{Message HMAC} = \text{HMAC}_{k_{sym}}(\text{Ciphertext}).$$

The encrypted ciphertext and the HMAC are sent together to the receiver. Upon receipt, the receiver first verifies the HMAC to confirm the integrity and authenticity of the ciphertext:

$$\text{Computed HMAC} = \text{HMAC}_{k_{sym}}(\text{Ciphertext}').$$

If the computed HMAC matches the received HMAC, the ciphertext is deemed intact. The receiver then decrypts the ciphertext using k_{sym} to retrieve the plaintext message:

$$\text{Plaintext} = d_{k_{sym}}(\text{Ciphertext}').$$

This method ensures both confidentiality and integrity of the communication. The symmetric key k_{sym} provides consistent encryption and HMAC generation, protecting against unauthorized tampering or interception of messages within a smart grid environment.

3.2.6. Security Analysis

The proposed protocol exhibits comprehensive security features designed to effectively mitigate significant cyber threats within smart grid environments. It addresses man-in-the-middle (MITM) attacks through the implementation of mutual certificate verification and shared secret derivation, which collectively renders it highly challenging for adversaries to impersonate either party involved in communication. Furthermore, the incorporation of timestamps and random nonce generation enhances protection against replay attacks, ensuring that each communication session is distinct and not vulnerable to reuse by malicious entities.

To reduce the likelihood of Denial-of-Service (DoS) attacks, the protocol incorporates an early verification mechanism that suspends authentication when certificate validation fails, thereby preventing the unnecessary consumption of resources. In terms of session security, the protocol generates unique symmetric keys for each session, effectively mitigating risks associated with session key attacks. It also ensures the confidentiality of sensitive identifiers through secure encryption, while robust mutual authentication measures—anchored in secure private key usage and HMAC verification—provide deterrence against impersonation attempts.

Lastly, mutual authentication is established through a bidirectional exchange and validation of certificates, ensuring that only authorized participants can engage in secure communications. These security features collectively highlight the protocol's resilience and suitability for protecting critical smart grid infrastructures.

4. Experimental Setup

This section describes the experimental setup used to evaluate the proposed protocol designed for smart grid environments. The setup features a server–client architecture and simulates real-world network conditions for a smart grid environment.

4.1. Hardware Configuration

Figure 4 depicts the communication between the server and clients. The server node, which functions as the SCADA, has been implemented on a high-performance machine equipped with an Intel Core-i9 32-core processor operating at 2.0 GHz, with 32 GB of random access memory (RAM) and a Gigabit Ethernet interface. This server has been assigned a static IP address of `192.168.1.1`.

In the experimental testbed, client nodes, which represent various distributed components of the smart grid, including smart meters and EV chargers, have been deployed on a diverse array of hardware configurations. These configurations range from embedded devices (e.g., ESP-8266 modules) to high-performance octa-core machines. Client machines have been allocated dynamic IP addresses within the range of `192.168.1.101` to `192.168.1.150`.

All nodes are interconnected through a Gigabit Ethernet local area network (LAN) that is managed by a network switch. Manual delays ranging from 0.1 ms to 5 ms have been introduced to emulate real-world conditions.

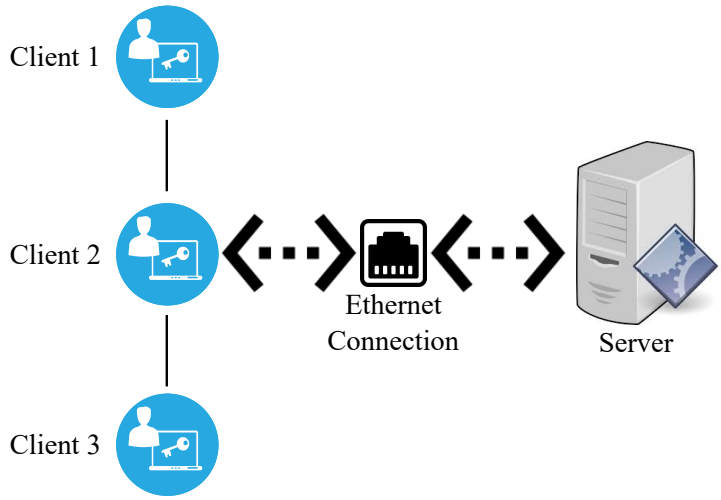

Figure 4. Overview of server–client communication architecture.

4.2. Software Configuration

The server and client applications developed in the C programming language employ HTTP communication to enable efficient data exchange over a network. It is important to highlight that HTTP was utilized primarily for experimental evaluation and to facilitate ease of implementation. The proposed protocol is designed to be protocol-agnostic, allowing for adaptation and deployment across industry-standard smart grid communication protocols, such as IEC 61850, DNP3, or MQTT over TLS. Furthermore, the security mechanisms implemented, including ECDSA for authentication, ECDH for key exchange, and AES-256 encryption, operate independently of HTTP, ensuring compatibility with secure transport layers.

The communication protocol encompasses three fundamental operations:

1. `/authenticate`: handles client authentication using ECDSA.
2. `/exchange-ecc`: establishes a shared secret using ECC-based ECDH.
3. `/send-message`: facilitates secure message exchange using AES-256-CBC encryption.

The server was hosted on port 3000 and was configured to handle incoming HTTP requests at `http://192.168.1.1:3000`. Clients connect to the server using their designated IP addresses and interact with predefined API endpoints for secure communication. For cryptographic operations, we utilized Open Secure Sockets Layer (OpenSSL) libraries,

implementing ECDSA for authentication, ECDH for secret key establishment, and AES-256-CBC for encryption and decryption.

Both server and client operations were orchestrated using a custom Makefile, which streamlines the processes of key generation, compilation, and execution. The ECC keys for both parties were pre-generated utilizing a script and securely stored in designated files for future reference. This methodology significantly enhanced security by circumventing the need for public key transmission across the network. The private keys were securely stored in designated files, while the public keys were embedded in certificates issued by a trusted Certificate Authority (CA). These certificates allow the public keys to be shared and verified securely without exposing sensitive private key information.

4.3. Experimental Scenarios

The following scenarios were assessed to validate the functionality and performance of the protocol:

- **Secure Authentication and Communication:** Legitimate clients authenticated with the server performed ECDH-based key exchange and securely exchanged encrypted messages using AES. The decrypted message was verified on the server side to ensure successful decryption.
- **Authentication Failure:** Malicious clients attempting to access the server were denied after failed authentication attempts. Such events were logged for further investigation.
- **Network Delays:** Manual delays were introduced to assess the protocol's robustness under varying communication delays.

4.4. Evaluation Metrics

The effectiveness of the protocol was evaluated using the following metrics:

- **Time analysis:** The time taken for each protocol step was recorded to assess the system's responsiveness, including authentication, key exchange, and message exchange.
- **Throughput:** The server's capability to manage multiple clients was assessed by systematically increasing the number of connected clients to 10, 50, and 100.
- **Power analysis:** The power consumption of the protocol quantification during its execution.
- **Cryptographic Overhead:** The time taken to perform ECDH key exchange and AES encryption/decryption operations was measured.

4.5. Key Generation and Security Measures

To establish secure communication, ECC keys for both the server and client were generated utilizing a script before authentication begins. These keys were securely stored in designated files in respective nodes and loaded during runtime.

The experimental setup helps us to evaluate the protocol's capability to deliver secure, scalable, and efficient communication within a smart grid system.

5. Results

This section presents the findings from our proposed protocol. We evaluated the protocol based on different metrics such as latency, scalability, and cryptographic overhead. Experiments were conducted on a high-end server equipped with an Intel Xeon 24-core processor operating at 2.0 GHz, 32 GB of RAM, and nodes of various specifications.

5.1. Performance Analysis

The performance of the proposed protocol was evaluated using metrics, including latency, scalability, cryptographic overhead, and resource utilization, to assess its effectiveness in real-world scenarios.

5.1.1. Time Analysis Based on Each Cryptographic Operation

Table 2 presents a detailed analysis of the time required for each cryptographic operation involved in the protocol, illustrating performance metrics across three platforms: the server, a Core-i7 desktop client, and an ESP-8266 embedded client. This evaluation assesses the protocol's responsiveness and its effectiveness across diverse hardware configurations.

The authentication process is divided into two phases: signing, performed on the server, and verification, carried out by the clients. The signing operation is highly efficient on the server, requiring only 0.019 ms, while the verification process is more computationally demanding. The Core-i7 client requires 1.247 ms for verification, whereas the ESP-8266 embedded client takes 4.070 ms to complete the same task.

Table 2. Time utilized by each cryptographic operation involved in the protocol on both server and client.

Protocol Steps	Server	Client	
		Core-i7	ESP-8266
	Time (ms)		
Authentication (Signing)	0.019	0.021	1.979
Authentication (Verification)	0.066	1.247	4.070
Local Key Derivation	1.268	2.597	4.601
HMAC Generation	0.006	0.009	0.040
HMAC Verification	0.008	0.011	0.400
AES Encryption	0.002	0.003	0.158
AES Decryption	0.002	0.005	0.138

In addition, local key derivation, which involves elliptic curve computations and hash functions, reveals significant differences in performance depending on the device. The server completes this task in 1.268 ms, the Core-i7 client in 2.597 ms, and the ESP-8266 in 4.601 ms. These differences highlight the additional processing overhead associated with embedded environments that possess limited computational resources.

HMAC operations exhibit impressive efficiency across all platforms. The generation phase requires 0.006 ms on the server, 0.009 ms on the Core-i7 client, and 0.040 ms on the ESP-8266. Verification times are also minimal, with 0.008 ms on the server, 0.011 ms on the Core-i7 client, and 0.400 ms on the ESP-8266.

AES operations further highlight the protocol's adaptability and speed, making it ideal for both high-performance and resource-constrained devices. Encryption times are 0.002 ms (for encrypting k_{sym}, r_i, T, and ID_S to generate y_{C_i} as shown in Figure 3) on the server, 0.003 ms (for encrypting $Cert_{C_i}$ and r_i to generate $m^r_{C_i}$ as shown in Figure 3) on the Core-i7 client, and 0.158 ms on the ESP-8266 client. Decryption times are similarly efficient at 0.002 ms on the server, 0.005 ms on the Core-i7 client, and 0.138 ms on the ESP-8266. While processes like authentication verification and local key derivation take more time, especially on the ESP-8266. However, the overall performance of the protocol remains efficient.

5.1.2. Time Analysis Based on Steps Involved in Protocol

Table 3 provides a comprehensive overview of the duration required for critical protocol steps executed on the server, Core-i7 client, and the ESP-8266 embedded client. In Step 2 of the protocol, the server exclusively generates the symmetric key and the time period T for the symmetric key, while the client side involves certificate verification, key derivation, HMAC, and nonce generation. The server exhibits high efficiency, completing this step in 0.015 ms due to a reduced number of operations, whereas the Core-i7 client takes 2.767 ms. In contrast, the ESP-8266 experiences significantly greater latency at 41.143 ms, which can be attributed to hardware limitations.

Step 3, which is executed solely on the server and entails the execution of all operations from Step 2 on the client side, requires 3.766 ms. In addition, Step 4, which primarily focuses on decryption and verification, is completed in 0.039 ms on the Core-i7 and 0.283 ms on the ESP-8266.

The total execution time reflects the cumulative duration of all steps, with the server completing the entire process in 3.940 ms and the Core-i7 client achieving a time of 3.469 ms, indicating substantial efficiency. Although the ESP-8266 records a longer duration of 44.426 ms, it is noteworthy that this device remains capable of supporting the protocol, demonstrating its viability even under resource-constrained conditions.

Table 3. Time utilized by each step on both the server and client to evaluate the protocol responsiveness.

Protocol Steps	Server	Client	
		Core-i7	Esp-8266
	Time (ms)		
Step 2	0.015	2.767	41.143
Step 3	3.766	–	–
Step 4	–	0.039	0.283
Total Time	3.940	3.469	44.426

5.1.3. Throughput Analysis

Table 4 provides a comprehensive evaluation of the protocol's performance across varying client loads on different platforms, specifically a core-i7 desktop client and the ESP-8266 microcontroller. The data indicate that as the number of clients increases, the total response time correspondingly rises, reflecting the additional computational demands of managing multiple simultaneous requests. For the core-i7 desktop system, response times ranged from 3.47 ms for a single client to 387.28 ms when 100 clients were active. In contrast, the ESP-8266, constrained by limited hardware resources, exhibited higher response times, beginning at 41.14 ms for a single client and escalating to 4589.67 ms for 100 clients.

Table 4. Throughput analysis with a different number of clients.

No. of Clients	Total Response Time (ms)		Throughput (Clients/s)	
	Core-i7	ESP-8266	Core-i7	ESP-8266
1	3.47	41.14	288.27	24.30
10	36.90	427.27	271.01	23.40
50	189.21	2198.15	264.56	22.75
100	387.28	4589.67	258.21	21.79

Throughput, measured as the number of requests processed per second, gradually declined as the number of clients increased, attributed to resource contention. The desktop

system maintained a commendable throughput, decreasing from 288.27 requests per second for 1 client to 258.21 requests per second for 100 clients. Meanwhile, the ESP-8266 achieved lower throughput, starting at 24.30 requests per second for a single client and decreasing to 21.79 requests per second at full capacity. Despite the ESP-8266 exhibiting relatively lower performance metrics, it succeeded in maintaining reasonable response times and throughput, highlighting the protocol's efficacy for resource-constrained grid edge devices.

5.1.4. Resource Utilization and Energy Efficiency

The analysis of resource usage, encompassing memory and power consumption, was conducted during various protocol phases. The findings underscore the protocol's efficiency in performance:

1. **Memory Usage:** Peak memory utilization remained below 6 MB for both the client and server, even under high load conditions. The memory usage results indicate modest overhead for devices operating with limited resources/memory.
2. **Power Consumption:** Figure 5 presents the power consumption metrics observed during the execution of the protocol. The power consumption data were collected using the `powerstat` tool, which captured readings at one second intervals over a total duration of 60 s. These readings were sourced from RAPL (Running Average Power Limit), an Intel processor feature that allows real-time measurements of CPU and RAM. To enhance accuracy, we collected power readings at a frequency of 10 samples per second, computing the average of these 10 readings for each second over a total duration of 60 s. This averaging approach smooths out transient fluctuations and provides a more reliable representation of the protocol's power consumption. While short-term variations were observed, the overall trend remained stable. During the execution of the proposed protocol, the system exhibited an average power consumption of 37.77 watts, accompanied by a geometric mean of 37.37 watts and a standard deviation of 5.66 watts. The maximum recorded power consumption was 54.14 watts, while the minimum was 28.46 watts. The close alignment between the arithmetic average and the geometric mean indicates a stable power consumption pattern with minimal skewness. Furthermore, the moderate standard deviation reinforces this observation, suggesting that the power demands of the protocol remain consistent throughout the cryptographic operations executed. These findings underscore the protocol's efficiency and its appropriateness for energy-constrained environments, where predictable and moderate power usage is crucial.
3. **Power Efficiency:** An assessment based on average power consumption and throughput data reveals that the protocol achieves a commendable balance between performance and power usage. For instance, with a throughput of 594 requests per second, the estimated energy cost per request is approximately 0.063 watts/second, showcasing its energy efficiency.

The combination of low resource utilization and power efficiency demonstrates the proposed protocol's suitability for smart grids with resource-constrained grid edge nodes.

5.1.5. Security Metrics and Failure Recovery

The resilience of the protocol was thoroughly evaluated through controlled disruptions during client–server communication, including sudden disconnections and instances of packet loss. The server demonstrated an effective response to these challenges, achieving recovery within an average of 3 ms. This recovery process included the re-establishment of client sessions and the continuation of protocol procedures, ensuring that both the shared secret and the integrity of the encrypted messages remained intact.

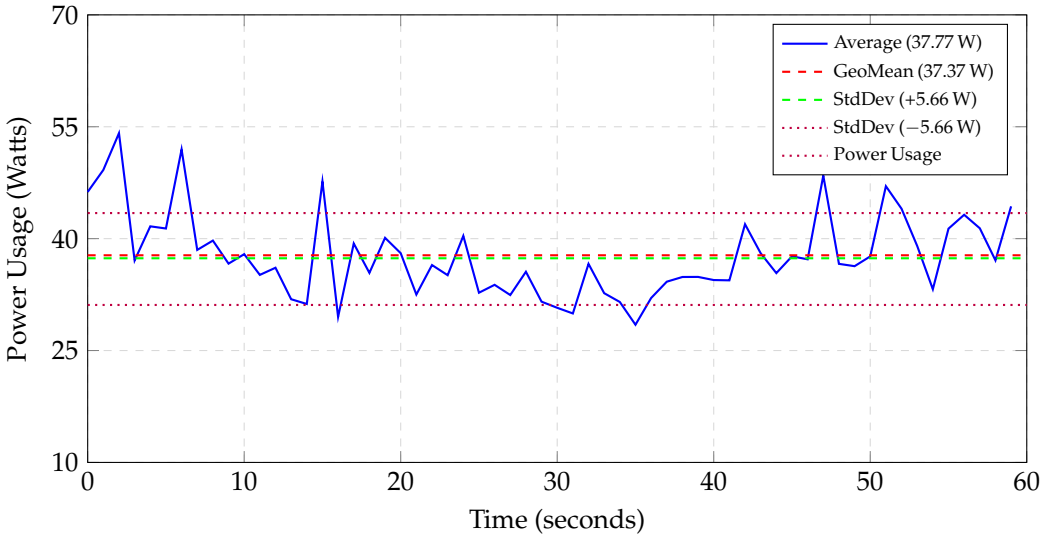

Figure 5. Power usage over time during protocol execution.

To further assess the robustness of the protocol, transient network failures and disruptions have been conducted. In all instances, the protocol maintained a secure state and seamlessly resumed operations, underscoring its capability to adapt to dynamic and adverse conditions without compromising security and mitigating performance degradations.

The protocol's ability to support rapid recovery and sustained, secure communication highlights its robustness, making it a dependable solution for applications that demand high resilience to network failures alongside stringent security standards.

5.2. Comparison with State-of-the-Art Methods

Table 5 illustrates the average computational time of the proposed protocol compared to state-of-the-art methods. To facilitate a fair comparison, we scaled the computational values of other protocols to match our testbed specifications [44]. While scaling may not provide 100% accuracy for system runtime due to variations in the instruction set architectures, different numbers of cores and memory subsystems, it offers reasonable estimates that allow for valid relative comparisons.

Table 5. Average computational time of our proposed protocol compared to other state-of-the-art methods.

Methods	Server (ms)	Client (ms)	Total (ms)	Speedup
Garg et al. [45]	15.82	17.92	33.73	4.55×
SM2 [46]	15.54	16.58	32.12	4.33×
PSLA [40]	16.83	14.34	31.17	4.20×
Proposed	3.94	3.47	7.41	-

The results show that the proposed protocol achieves significantly lower computation time on both the server and client sides, reducing the overall execution time to 7.41 ms, which is 4.55×, 4.33× and 4.20× faster than Garg et al. [45], SM2 [46], and PSLA [40]. The substantial reduction in execution time highlights the lightweight nature of our protocol, making it well suited for resource-constrained smart grid environments without compromising security.

5.3. Discussion on Smart Grid Security Standards and Real-World Applicability

To ensure our proposed protocol complies with smart grid security standards, we evaluated its cryptographic overhead on high-performance (Core-i7) and resource-constrained (ESP32) hardware. The results indicate that while computational efficiency varies between these platforms, the protocol is sufficiently feasible for low-power embedded devices typically utilized in smart grid environments. This aspect is particularly important, given that a smart grid embodies edge devices such as smart meters, sensors, and remote terminal units (RTUs) with limited processing capabilities. Additionally, our protocol conforms to the fundamental principles of the IEC 62351 protocol, which serves as the security standard for power system communication. The incorporation of ECDSA for authentication, ECDH for secure key exchange, and AES-256 for encryption combined with HMAC for data confidentiality and integrity, respectively, is consistent with the requirements outlined in IEC 62351-3 (secure transport), IEC 62351-5 (SCADA security), and IEC 62351-8 (role-based authentication). Given that IEC 61850-based systems often necessitate lightweight yet secure cryptographic implementations, our findings suggest that the proposed protocol can seamlessly integrate into smart grid infrastructures with minimal computational overhead. By demonstrating its adaptability to real-world constraints, our study offers a scalable and secure solution for authentication and communication within smart grid systems.

6. Conclusions

The proposed secure communication protocol addresses critical security challenges within smart grid environments with a comprehensive approach. By incorporating ECDSA- and ECDH-based key exchange in conjunction with AES-256, the protocol delivers robust authentication, secure key distribution, and reliable data encryption. Performance evaluations highlight its suitability for real-world deployment, achieving a total server-side execution time of 3.940 ms, a throughput of 288.27 requests per second on high-performance devices, and 21.79 requests per second on embedded platforms like ESP-8266. The protocol demonstrates a low power consumption of 37.77 W on average with an energy cost of 0.063 joules per request and resilience to network failures with an average recovery time of 3 ms. Minimal cryptographic overhead, efficient AES encryption (0.002–0.158 ms), and HMAC generation times (0.006–0.040 ms) further establish its adaptability for dynamic and resource-constrained environments such as smart grids.

Future efforts will expand the protocol's capabilities to adapt to emerging technologies and tackle the challenges posed by quantum computing, especially concerning Shor's algorithm. Integrating post-quantum cryptographic (PQC) solutions and exploring hybrid handshaking approaches will facilitate secure transactions in smart grid systems. Furthermore, upcoming research will focus on innovative security mechanisms, including physically unclonable functions (PUFs) and AI-driven anomaly detection to strengthen smart grid robustness.

Author Contributions: Conceptualization, A.M.; methodology, A.M. and M.A.H.; software, M.A.H. and K.H.S.; validation, A.M.; formal analysis, A.M., M.A.H. and K.H.S.; investigation, A.M.; resources, A.M.; data curation, M.A.H. and K.H.S.; writing—original draft preparation, M.A.H. and K.H.S.; writing—review and editing, A.M.; visualization, M.A.H. and K.H.S.; supervision, A.M.; project administration, A.M.; funding acquisition, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work supported by the U.S. Department of Energy, Office of Cybersecurity, Energy Security, and Emergency Response (CESER), under Award Number DE-CR0000050. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their

Data Availability Statement: The code of the developed security protocol is available at: https://github.com/iscaas/DOE-CESER-2024-2027/tree/main/JCP_PKE.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nafi, N.S.; Ahmed, K.; Gregory, M.A.; Datta, M. A survey of smart grid architectures, applications, benefits and standardization. *J. Netw. Comput. Appl.* **2016**, *76*, 23–36. [CrossRef]
2. Babayomi, O.; Zhang, Z.; Dragicevic, T.; Hu, J.; Rodriguez, J. Smart grid evolution: Predictive control of distributed energy resources—A review. *Int. J. Electr. Power Energy Syst.* **2023**, *147*, 108812.
3. Mohassel, R.R.; Fung, A.S.; Mohammadi, F.; Raahemifar, K. A survey on advanced metering infrastructure and its application in smart grids. In Proceedings of the 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE), Toronto, ON, Canada, 4–7 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–8.
4. Sayed, K.; Gabbar, H.A. SCADA and smart energy grid control automation. In *Smart Energy Grid Engineering*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 481–514.
5. Amiri, S.S.; Rahmani, M.; McDonald, J.D. An updated review on distribution management systems within a smart grid structure. In Proceedings of the 2021 11th Smart Grid Conference (SGC), Tabriz, Iran, 7–9 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5.
6. Bagherzadeh, L.; Shahinzadeh, H.; Shayeghi, H.; Dejamkhooy, A.; Bayindir, R.; Iranpour, M. Integration of cloud computing and IoT (CloudIoT) in smart grids: Benefits, challenges, and solutions. In Proceedings of the 2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE), Keonjhar, India, 29–31 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
7. Faquir, D.; Chouliaras, N.; Sofia, V.; Olga, K.; Maglaras, L. Cybersecurity in smart grids, challenges and solutions. *AIMS Electron. Electr. Eng.* **2021**, *5*, 24–37.
8. Leszczyna, R. A review of standards with cybersecurity requirements for smart grid. *Comput. Secur.* **2018**, *77*, 262–276. [CrossRef]
9. Admass, W.S.; Munaye, Y.Y.; Diro, A.A. Cyber security: State of the art, challenges and future directions. *Cyber Secur. Appl.* **2024**, *2*, 100031.
10. Martin, K.; Brunello, G.; Adamiak, M.; Antonova, G.; Begovic, M.; Benmouyal, G.; Bui, P.; Falk, H.; Gharpure, V.; Goldstein, A.; et al. An overview of the IEEE standard C37. 118.2—synchrophasor data transfer for power systems. *IEEE Trans. Smart Grid* **2014**, *5*, 1980–1984.
11. Khan, R.; McLaughlin, K.; Laverty, D.; Sezer, S. Analysis of IEEE C37. 118 and IEC 61850-90-5 synchrophasor communication frameworks. In Proceedings of the 2016 IEEE Power and Energy Society General Meeting (PESGM), Boston, MA, USA, 17–21 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–5.
12. Burgos, D.L. DoS Threat to Smart Grids: Review, Analysis, and Challenges. 2024. Available online: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3154983#:~:text=The%20cumulative%20effect%20of%20DoS,for%20maintaining%20uninterrupted%20service%20in%publisher:{NTNU} (accessed on 20 December 2024)
13. Amara, M.; Siad, A. Elliptic curve cryptography and its applications. In Proceedings of the International Workshop on Systems, Signal Processing and Their Applications, WOSSPA, Tipaza, Algeria, 9–11 May 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 247–250.
14. Haakegaard, R.; Lang, J. The Elliptic Curve Diffie-Hellman (ecdh). 2015. Available online: https:////koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Haakegaard+Lang.pdf (accessed on 20 December 2024).
15. Abdullah, A.M. Advanced encryption standard (AES) algorithm to encrypt and decrypt data. *Cryptogr. Netw. Secur.* **2017**, *16*, 11.
16. Joseph, A.; Balachandra, P. Smart Grid to Energy Internet: A Systematic Review of Transitioning Electricity Systems. *IEEE Access* **2020**, *8*, 215787–215805. [CrossRef]
17. Saidani Neffati, O.; Sengan, S.; Thangavelu, K.D.; Dilip Kumar, S.; Setiawan, R.; Elangovan, M.; Mani, D.; Velayutham, P. Migrating from traditional grid to smart grid in smart cities promoted in developing country. *Sustain. Energy Technol. Assess.* **2021**, *45*, 101125. [CrossRef]

18. Kakran, S.; Chanana, S. Smart operations of smart grids integrated with distributed generation: A review. *Renew. Sustain. Energy Rev.* **2018**, *81*, 524–535. [CrossRef]
19. Yang, Q.; An, D.; Min, R.; Yu, W.; Yang, X.; Zhao, W. On Optimal PMU Placement-Based Defense Against Data Integrity Attacks in Smart Grid. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1735–1750. [CrossRef]
20. Blair, S.M.; Syed, M.H.; Roscoe, A.J.; Burt, G.M.; Braun, J.P. Measurement and Analysis of PMU Reporting Latency for Smart Grid Protection and Control Applications. *IEEE Access* **2019**, *7*, 48689–48698. [CrossRef]
21. Swain, K.P.; Tiwari, A.; Sharma, A.; Chakrabarti, S.; Karkare, A. Comprehensive Demonstration of Man-in-the-Middle Attack in PDC and PMU Network. In Proceedings of the 2022 22nd National Power Systems Conference (NPSC), New Delhi, India, 17–19 December 2022; pp. 213–217. [CrossRef]
22. Zhang, K.; Hu, Z.; Zhan, Y.; Wang, X.; Guo, K. A Smart Grid AMI Intrusion Detection Strategy Based on Extreme Learning Machine. *Energies* **2020**, *13*, 4907. [CrossRef]
23. Abir, S.M.A.A.; Anwar, A.; Choi, J.; Kayes, A.S.M. IoT-Enabled Smart Energy Grid: Applications and Challenges. *IEEE Access* **2021**, *9*, 50961–50981. [CrossRef]
24. Alaklabi, A.; Munir, A.; Asfand Hafeez, M.; Khan Khattak, M.A. Z-Crypt: Chirp Z-Transform-Based Image Encryption Leveraging Chaotic Logistic Maps and Substitution Permutation Network. *IEEE Access* **2024**, *12*, 123401–123422. [CrossRef]
25. Wu, Y.; Guo, H.; Han, Y.; Li, S.; Liu, J. A Security-Enhanced Authentication and Key Agreement Protocol in Smart Grid. *IEEE Trans. Ind. Inform.* **2024**, *20*, 11449–11457. [CrossRef]
26. Hasan, M.K.; Habib, A.A.; Shukur, Z.; Ibrahim, F.; Islam, S.; Razzaque, M.A. Review on cyber-physical and cyber-security system in smart grid: Standards, protocols, constraints, and recommendations. *J. Netw. Comput. Appl.* **2023**, *209*, 103540. [CrossRef]
27. Liu, M.; Teng, F.; Zhang, Z.; Ge, P.; Sun, M.; Deng, R.; Cheng, P.; Chen, J. Enhancing Cyber-Resiliency of DER-Based Smart Grid: A Survey. *IEEE Trans. Smart Grid* **2024**, *15*, 4998–5030. [CrossRef]
28. Acharya, S.; Dvorkin, Y.; Pandžić, H.; Karri, R. Cybersecurity of Smart Electric Vehicle Charging: A Power Grid Perspective. *IEEE Access* **2020**, *8*, 214434–214453. [CrossRef]
29. Potvin, M.; Eng, P. The AURORA vulnerability: The sword of Damocles over the heads of rotating machines. In Proceedings of the CIGRE Canada Conference, Le Westin Montréal, QC, Canada, 16–19 September 2019.
30. Bakić, B.; Milić, M.; Antović, I.; Savić, D.; Stojanović, T. 10 years since Stuxnet: What have we learned from this mysterious computer software worm? In Proceedings of the 2021 25th International Conference on Information Technology (IT), Zabljak, Montenegro, 16–20 February 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–4.
31. Symantec, O. Dragonfly: Western Energy Sector Targeted by Sophisticated Attack Group. 2017. Available online: https://www.security.com/threat-intelligence/dragonfly-energy-sector-cyber-attacks (accessed on 20 December 2024).
32. Case, D.U. *Analysis of the Cyber Attack on the Ukrainian Power Grid*; Electricity Information Sharing and Analysis Center (E-ISAC): Washington, DC, USA, 2016; Volume 388, p. 3.
33. Pape, T. Insomnolence: The Sociability of Sleep at Agora Hydro-Québec. 2024. Available online: https://necsus-ejms.org/insomnolence-the-sociability-of-sleep-at-agora-hydro-quebec/ (accessed on 21 December 2024).
34. Milanov, E. *The RSA Algorithm*; RSA Laboratories: Hebron CT, USA, 2009; pp. 1–11.
35. Maurer, U.M.; Wolf, S. The diffie–hellman protocol. *Des. Codes Cryptogr.* **2000**, *19*, 147–171. [CrossRef]
36. Fouda, M.M.; Fadlullah, Z.M.; Kato, N.; Lu, R.; Shen, X.S. A lightweight message authentication scheme for smart grid communications. *IEEE Trans. Smart Grid* **2011**, *2*, 675–685.
37. Nicanfar, H.; Leung, V.C. Multilayer consensus ECC-based password authenticated key-exchange (MCEPAK) protocol for smart grid system. *IEEE Trans. Smart Grid* **2013**, *4*, 253–264. [CrossRef]
38. Tanveer, M.; Kumar, N.; Naushad, A.; Chaudhry, S.A. A robust access control protocol for the smart grid systems. *IEEE Internet Things J.* **2021**, *9*, 6855–6865.
39. Hu, S.; Chen, Y.; Zheng, Y.; Xing, B.; Li, Y.; Zhang, L.; Chen, L. Provably secure ECC-based authentication and key agreement scheme for advanced metering infrastructure in the smart grid. *IEEE Trans. Ind. Inform.* **2022**, *19*, 5985–5994. [CrossRef]
40. Chai, S.; Yin, H.; Xing, B.; Li, Z.; Guo, Y.; Zhang, D.; Zhang, X.; He, D.; Zhang, J.; Yu, X.; et al. Provably secure and lightweight authentication key agreement scheme for smart meters. *IEEE Trans. Smart Grid* **2023**, *14*, 3816–3827.
41. Al-Riyami, S.S.; Paterson, K.G. *Certificateless Public Key Cryptography, Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 452–473.
42. Liu, S.; Liu, Y.; Liu, W.; Zhang, Y. A certificateless multi-dimensional data aggregation scheme for smart grid. *J. Syst. Archit.* **2023**, *140*, 102890. [CrossRef]
43. Wang, W.; Xu, H.; Alazab, M.; Gadekallu, T.R.; Han, Z.; Su, C. Blockchain-based reliable and efficient certificateless signature for IIoT devices. *IEEE Trans. Ind. Inform.* **2021**, *18*, 7059–7067.
44. Munir, A.; Gordon-Ross, A.; Lysecky, S.; Lysecky, R. A lightweight dynamic optimization methodology and application metrics estimation model for wireless sensor networks. *Sustain. Comput. Inform. Syst.* **2013**, *3*, 94–108.

45. Garg, S.; Kaur, K.; Kaddoum, G.; Rodrigues, J.J.; Guizani, M. Secure and lightweight authentication scheme for smart metering infrastructure in smart grid. *IEEE Trans. Ind. Inform.* **2019**, *16*, 3548–3557. [CrossRef]
46. Bergman Martinkauppi, L.; He, Q. Performance Evaluation and Comparison of Standard Cryptographic Algorithms and Chinese Cryptographic Algorithms. Master's Thesis, Blekinge Institute of Technology, Karlskrona, Sweden, 2019.