Received 15 April 2024; revised 21 May 2024; accepted 31 May 2024. Date of publication 4 June 2024; date of current version 1 July 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3409451

Lightweight Authenticated Key Agreement Protocol for Smart Power Grid Systems Using PUF

AMINA ZAHOOR[1], KHALID MAHMOOD[2] (Senior Member, IEEE), MUHAMMAD ASAD SALEEM[3], HAFIZ MUHAMMAD SANAULLAH BADAR[4], TUAN-VINH LE[5] (Member, IEEE), AND ASHOK KUMAR DAS[6] (Senior Member, IEEE)

[1]Department of Computer Science, COMSATS University Islamabad (Sahiwal Campus), Sahiwal 57000, Pakistan

[2]Graduate School of Intelligent Data Science, National Yunlin University of Science and Technology, Douliu 64002, Taiwan

[3]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

[4]Department of Emerging Computing Technologies, Emerson University Multan, Multan 61000, Pakistan

[5]Bachelor's Program of Artificial Intelligence and Information Security, Fu Jen Catholic University, New Taipei City 24206, Taiwan

[6]Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad 500032, India

CORRESPONDING AUTHORS: K. MAHMOOD AND T.-V. LE (e-mail: khalidm.research@gmail.com; 155315@mail.fju.edu.tw)

This work was supported by the National Science and Technology Council, Taiwan, under Grant NSTC-112-2222-E-030-001.

ABSTRACT The Smart Power Grid (SPG) is pivotal in orchestrating and managing demand response in contemporary smart cities, leveraging the prowess of Information and Communication Technologies (ICTs). Within the immersive SPG environment, the ubiquitous deployment of smart meters stands as a testament to their paramount importance in the realm of vigilant monitoring and oversight. These smart meters are installed on high-tension electricity lines and transmit information about electricity outages and other issues to the utility center. To access services from utility centers, smart meters need to communicate securely over a public channel, even though the network itself is insecure. However, potential attacks from adversaries ($\mathcal{A}_d$) can exploit this communication. Therefore, protecting this communication is of utmost importance. Several privacy-preserving authentication protocols designed for SPG have been introduced in the literature. Nevertheless, a significant number of these protocols exhibit vulnerabilities to diverse security attacks. This article introduces a lightweight and anonymous authentication protocol specifically designed to address these concerns in the SPG environment. Our protocol ensures both security and efficiency, surpassing other comparable protocols in terms of its lightweight nature. By employing both formal and informal analysis, we showcase the robustness of our protocol against significant attacks while remaining lightweight. The proposed protocol provides added security features and incurs 23.8879% lower computation cost than related protocols. Consequently, our protocol is highly suitable for implementation in the SPG system.

INDEX TERMS Authentication, IoT, information security, smart power grids, physically unclonable function (PUF).

I. INTRODUCTION

THIS recent advancements in Information and Communication Technology (ICT) have propelled SPGs to become a fundamental component of the Internet of Things (IoT) [1]. SPG systems primarily encompass intelligent appliances, smart homes, advanced metering infrastructure, renewable energy-based vehicle-to-grid systems, and intelligent buildings [2]. The traditional electrical grid infrastructure is inadequate to accommodate the evolving and increasing power requirements characteristic of the 21st century. Efficient utilization of electrical grids is crucial in the present times [3]. SPG introduces a modern digitalized infrastructure that supersedes the outdated framework of the conventional electrical grid [4]. While the

traditional grid framework only allows for unidirectional electricity flow from the grid to end users, SPG establishes a bidirectional connection between smart meters ($\mathcal{SM}_a$) and utility centers ($\mathcal{UC}_b$) [5]. Despite the benefits, its extensive dependence on networking systems and communication renders the SPG framework vulnerable to various attacks. Numerous authentication protocols have recently been proposed to tackle the privacy of SPG communications. Mahmood et al. [6] introduced an authentication protocol for SPG communication in 2018. However, their protocol fails to ensure resistance against various security threats, including insider, password guessing, and impersonation threats. In 2019, Li et al. [7] presented an efficient authentication protocol by applying the Public Key Infrastructure (PKI). Furthermore, Kumar et al. [8] presented a Demand Response (DR) management authentication protocol for SPG in 2019. Nonetheless, the protocol of [8] does not resist SPG device stolen attacks, $\mathcal{S}_{key}$ disclosure attacks, and impersonation attacks. Furthermore, the protocol of [8] cannot provide secure mutual authentication characteristics. In 2020, Yu et al. [9] proposed a lightweight privacy-preserving authenticated protocol for DR management in the SPG environment. Yu et al. claimed that their protocol protects against various security attacks. However, after detailed observation, it is analyzed that the protocol of [9] lacks mutual authentication and is prone to denial of service (DOS) and replay threats. In 2020, Khan et al. [10] presented a password-based anonymous key agreement protocol using the hash method and ECC. In addition, Khan et al. simulated their protocol using the AVISPA tool to prove it is safe against side-channel and MITM threats. In 2021, Srinivas et al. [11] presented a signature-based authentication protocol for SPG systems. Moreover, it is observed that the protocol of [11] fails to prevent various attacks, including $\mathcal{SM}_a$, and $\mathcal{SP}$ impersonation attacks. In 2021, Irshad et al. [12] introduced a DR management mutual authentication protocol for SPG. They claimed that the protocol of [9] is prone to replay and DOS threats. In 2022, Taqi and Jalili [13] presented an authentication protocol for SPG that ensures anonymity, yet it exhibits susceptibility to de-synchronization attacks and involves substantial communication overhead. This arises from the necessity to update transmitted parameters during the authentication stage. In 2023, Badar et al. [14] presented a robust authentication protocol designed for secure home domain networks. The paper leverages the non-destructive physical systems of ECC and $\mathcal{SM}_a$ to uphold identity privacy and security. Later on, Lee et al. [15] introduced a conserved data aggregation protocol for SPG. This protocol encompasses a unique key pair and signature mechanism, thereby enhancing the privacy safeguards for blockchain applications within SPG, and Tomar et al. [16] introduced an aggregate signature protocol devoid of certificates, leveraging blockchain technology to ensure authentication and integrity within the envisioned SPG framework. However, after detailed observation, it is analyzed that the protocol of [16] lacks mutual authentication. Table 1 depicts a short description of different existing protocols along with their cryptographic primitives, advantages, and disadvantages. Consequently, our motivations and contributions to this paper are explained in the subsections below.

A. MOTIVATION

The above literature review analysis revealed that various designed protocols are not effectively preventing mutual authentication and different security attacks, such as security against MITM attacks and security against physical and cloning attacks. Furthermore, many protocols incur a significant computational cost, rendering them impractical for entities with constrained resources. Moreover, studies require consideration of the physical privacy and security of $\mathcal{SM}_a$. Nevertheless, some related papers emphasize the significance of the Physically Unclonable Function (PUF) in SPG infrastructure.

This led us to the motivation of developing a new lightweight authentication protocol that not only provides mutual authentication but also addresses existing flaws. Consequently, our contribution of this paper is defined in the next subsection.

B. MAIN CONTRIBUTIONS

The primary contributions of our paper are outlined below:

- We proposed a lightweight anonymous authenticated key agreement protocol for SPG environments using lightweight cryptographic operations such as XOR, concatenation operations, and hash functions.
- We implemented the PUF function in our proposed protocol to enhance its robustness against physical tampering and cloning attacks. The PUF function's inherent uniqueness, similar to the individuality of a human fingerprint, prevents duplication.
- The informal analysis validates that our protocol encompasses robust security measures to withstand diverse attacks.
- The comparison of security attributes shows that our protocol is enhanced with more security attributes.
- The formal security analysis and validation are observed using the Real or Random (ROR) model to evaluate the security and privacy of our protocol.
- Additionally, a performance comparison reveals that our protocol outperforms other comparative protocols regarding communication and computation costs, achieving efficiency improvements of 29.16% and 30.43%, respectively.
- The testbed experimentation and performance evaluation results conclusively demonstrate our protocol's superior performance compared to several competing protocols.

C. PAPER ORGANIZATION

The subsequent sections of this article are organized in the following manner: Section II presents the cryptographic

TABLE 1. Comparison summary of relevant authentication protocols.

Protocol	Cryptographic Primitives	Advantages	Disadvantages
Khan et al. [10]	∗ ECC ∗ Hash function	∗ Lightweight ∗ Prevents MITM attacks ∗ Prevents side-channel attacks	∗ Susceptible to impersonation attacks ∗ Inaccurate login and authentication phase
Srinivas et al. [11]	∗ ECC	∗ Lower communication cost ∗ Lower computation cost	∗ Anonymity violation ∗ Susceptible to $\mathcal{SM}_a$ impersonation attacks ∗ Vulnerable to $\mathcal{SP}$ impersonation attacks
Irshad et al. [12]	∗ Hash function	∗ Anonymous and lightweight ∗ Ensures anonymity and untraceability	∗ Susceptible to desynchronization attacks ∗ Ensures mutual authentication
Jalili et al. [13]	∗ ECC ∗ XOR ∗ Hash function	∗ Anonymous and lightweight ∗ Prevents denial-of-service attack ∗ Prevents reply and stolen verifier attacks	∗ Does not consider the internal attackers ∗ High communication costs
Hafiz et al. [14]	∗ ECC ∗ Hash function	∗ Physical attack resilience ∗ Prevention of MITM attack ∗ Prevention of replay attack	∗ Not provide integrity and batch verification
Kumari and Singh. [17]	∗ ECC ∗ Hash function	∗ Resists MITM attacks ∗ Resists smart meter impersonation attacks	∗ Vulnerable to physical and cloning attacks ∗ High computation cost
Rostampour et al. [18]	∗ Hash Function ∗ XOR	∗ Resists smart meter impersonation attacks ∗ Resists replay attacks ∗ Offers smart meter anonymity	∗ Vulnerable to physical and cloning attacks ∗ High computation cost

preliminaries of this article. Section III provides the proposed protocol. Sections IV and V presented our protocol's security analysis and performance comparison with comparative protocols. Section VI provides the conclusion of this article.

II. PRELIMINARIES

In this section, we introduce the primitive notations, system model, threat model, and design goals, respectively.

A. PRIMITIVE NOTATIONS

Within this section, all pertinent notations for the protocol are detailed as presented in Table 2.

TABLE 2. Notations and their explanation.

Notations	Explanation
$\mathcal{TR}_A$	Trusted authority
$\mathcal{SM}_a$	a^{th} smart meter
$\mathcal{UC}_b$	b^{th} utility center
$\mathcal{ID}_{\mathcal{SM}_a}, \mathcal{ID}_{\mathcal{UC}_b}$	Identity of smart meter and utility center, respectively
$n_{\mathcal{UC}_b}$	Number of utility centers
$\mathcal{E}_s(m,n)$	Non-singular elliptic curve: $y^2 \longleftarrow x^3 + mx + n \ (mod\ s)$, $m, n \in \mathcal{Z}_s \longleftarrow \{0,1,2,...,s-1\}, 4m^3 + 27n^2 \neq 0 \ (mod\ s)$
$\mathcal{P}$	Base point on $\mathcal{E}_s(m,n)$
$\mathcal{P} + \mathcal{Q}$	Elliptic curve point addition: $\mathcal{P}, \mathcal{Q} \in \mathcal{E}_s(m,n)$
$k\ .\ \mathcal{Q}$	Elliptic curve point multiplication: $k \in \mathcal{Z}_s^*, \mathcal{Q} \in \mathcal{E}_s(m,n)$
$\mathcal{A}_d$	Adversary
$(\mathcal{T}_{pub}, t_{pri})$	Private and public key pair of $\mathcal{TR}_A$, with $\mathcal{T}_{pub} = t_{pri}.\mathcal{P}$
$\mathcal{K}_{\mathcal{UC}_b}$	Shared key of a utility center
$\mathcal{PUF}$	Physical unclonable function
$(\mathcal{C}_h, \mathcal{R}_s)$	Challenge response pair
$t_1, t_2, \mathcal{T}_a, \mathcal{T}_b$	Current timestamps
$\Delta\mathcal{T}$	Maximum transmission delay
$\mathcal{PID}_{\mathcal{SM}_a}$	Pseudo identity of $\mathcal{SM}_a$
$\mathcal{PID}_{\mathcal{UC}_b}$	Pseudo identity of $\mathcal{UC}_b$
$\mathcal{E}_{\mathcal{K}_{\mathcal{UC}_b}}$	Encryption using $\mathcal{UC}_b$ shared key
$\mathcal{D}_{\mathcal{K}_{\mathcal{UC}_b}}$	Decryption using $\mathcal{UC}_b$ shared key
$\mathcal{PID}^{new}_{\mathcal{SM}_a}$	New pseudo identity of $\mathcal{SM}_a$
$h(.)$	One-way cryptographic hash method
$\|\ , \oplus$	Concatenation and XOR operations, respectively
$\mathcal{S}_{key}$	Session key among $\mathcal{SM}_a$ and $\mathcal{UC}_b$

B. SYSTEM MODEL

In SPG system, communication occurs among three key entities: $\mathcal{SM}_a$, $\mathcal{UC}_b$, and $\mathcal{TR}_A$. Fig. 1 illustrates the overall communication architecture of SPG. At the lower level, the architecture includes three distinct distribution networks: Building Area Network (BAN), Home Area Network (HAN), and Neighborhood Area Network (NAN).

Each DS is responsible for a single neighborhood, a single NAN has numerous BANs, a single BAN has multiple HANs, and n DSs are responsible for n neighborhoods. Consequently, each DS possesses a single NAN, wherein every NAN comprises K BANs, and each BAN consists of m HANs. $\mathcal{SM}_a$ authorizes the bidirectional connection between end users and electricity providers. $\mathcal{SM}_a$ possesses two interfaces: one serves as a gateway (GW) for transmission purposes, while the other is designated for electricity reading. $\mathcal{SM}_a$ positioned at various points in the classification are labeled as BANGW, HANGW, and NANGW, respectively. Through these $\mathcal{SM}_a$ or GW, consumers can verify their ongoing power consumption level of electricity. They also can adjust their power consumption level by running or shutting some devices [19]. This distributed intelligence, and increased automation in SPG gives its process better performance, stability, and security.

C. THREAT MODEL

We employed the Dolev-Yao (DY) threat model [20] to assess our protocol's security. According to this model, $\mathcal{A}_d$ can execute the following:

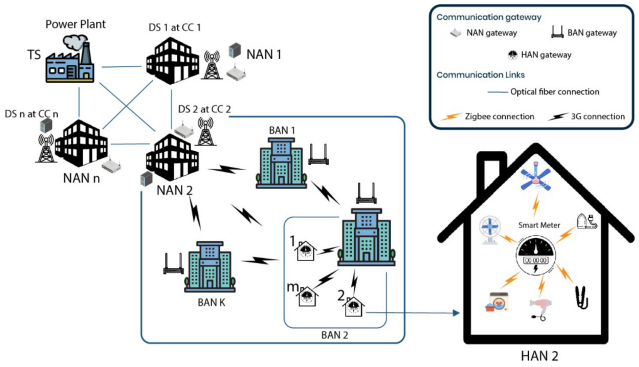

FIGURE 1. Smart Power Grid System.

- The communication network between $\mathcal{SM}_a$ and $\mathcal{UC}_b$ is considered public, granting $\mathcal{A}_d$ complete access to the messages transmitted through the open channel.
- $\mathcal{A}_d$ can inject, delete, edit, or eavesdrop on the original messages via a public network.
- $\mathcal{A}_d$ can capture $\mathcal{SM}_a$ and steal all stored content through power analysis [21].
- $\mathcal{A}_d$ can perform various attacks like modification, impersonation, MITM, replay attacks, etc.

We also adopt the Honest-But-Curious (HBC) threat model as outlined in [22], viewing Cent $\mathcal{UC}_b$ serving as an honest but curious participant of the system. Under this model, $\mathcal{UC}_b$ is permitted to perform passive attacks, enabling them to gather information about the locations of $\mathcal{SM}_a$ without interfering with their communications.

D. DESIGN GOALS

Within this subsection, we delve into the design objectives that necessitate careful consideration throughout the development of authentication protocols:

1) *Smart Meter Impersonation Attack (1A):* The protocol must prevent unauthorized entities from impersonating a legitimate $\mathcal{SM}_a$ to ensure that only authenticated devices can access the network and communicate with $\mathcal{UC}_b$.
2) *Utility Center Impersonation Attack (2A)* The protocol should ensure that$\mathcal{SM}_a$ can reliably authenticate $\mathcal{UC}_b$, preventing attackers from impersonating $\mathcal{UC}_b$ to gain unauthorized access or disrupt services.
3) *Man-in-the-Middle Attack (3A):* The protocol must safeguard against man-in-the-middle attacks, where an adversary intercepts and potentially alters the communication between $\mathcal{SM}_a$ and $\mathcal{UC}_b$, ensuring data integrity and confidentiality.
4) *Mutual Authentication (4A):* The designed protocol should guarantee the key agreement process is successfully completed only by authorized $\mathcal{SM}_a$'s and $\mathcal{UC}_b$'s, confirming their authentic identities. Upon mutual authentication, authorized entities must establish a shared $\mathcal{S}_{key}$ to facilitate subsequent communication between them.
5) *Smart Meter Anonymity and Untraceability (5A):* The protocol should protect the privacy of $\mathcal{SM}_a$ by ensuring that its identity remains anonymous and its activities untraceable to prevent tracking and profiling by unauthorized parties.
6) *Physical and Cloning Attacks (6A):* The protocol must be resilient against physical attacks and cloning attempts, ensuring that even if $\mathcal{SM}_a$ is physically tampered with or duplicated, the security and integrity of the system remain uncompromised.
7) *Forward Secrecy (7A):* Our Protocol ensure that even if a long-term key used in the protocol is compromised, past communications remain secure and undecipherable. Implement forward secrecy to protect against future compromises of keys.
8) *Efficient Resource Utilization:* Optimize the protocol to minimize computational and communication overhead for both $\mathcal{SM}_a$ and $\mathcal{UC}_b$, ensuring efficient use of resources such as bandwidth and processing power without compromising security.

E. PHYSICAL UNCLONABLE FUNCTION

A Physical Unclonable Function (PUF) is a physical system or structure that produces a unique and repeatable output response to a given input challenge. PUFs leverage inherent variations in physical properties, such as manufacturing variations or environmental factors, to generate responses that are difficult to predict or replicate. The response generated by a PUF is typically used for device authentication and security applications, as each PUF instance exhibits unique characteristics that can be exploited for cryptographic purposes without the need for stored secrets.

The characteristics of a PUF are as follows:

- It provides a single response for each challenge.
- It quickly generates responses known as CRPs.
- It requires a large number of CRPs to defend against modeling attacks.
- Changing each bit of the challenge results in at least a 50% change in the response.
- There is a limited range of methods to attack it.

In addition to these features, the PUF's suitability for device authentication is further enhanced by its unique properties, including:

- Uniqueness: When different devices are given the same challenge, the Hamming Distance of their responses should be 50%.
- Reliability: Despite varying environmental conditions, the same challenge produces the same response.
- Randomness: The response exhibits an equal probability of 0s and 1s.
- Correctness: Each response is trustworthy.
- Bit aliasing: The response's bit error rate is 50%.
- Uniformity: The distribution of 0s and 1s in the response is random.
- Steadiness: The response's bit error rate remains consistent and stable over time.

III. PROPOSED PROTOCOL

This section solicits our protocol for SPG systems. Through our protocol, $\mathcal{SM}a$ can establish a symmetry key with $\mathcal{UC}b$ to securely exchange real-time information about energy usage while ensuring the privacy of users. The following are the phases of our protocol:

1) System setup phase,
2) Smart meter registration phase,
3) Utility center registration phase,
4) Authentication and key agreement phase, and
5) Dynamic node update phase.

A. SYSTEM SETUP PHASE

In this process, $\mathcal{TR}_A$ performs the subsequent steps to pick the system variables:

Step 1: Initially, $\mathcal{TR}_A$ selects a non-singular elliptic curve $\mathcal{E}_s(m, n)$ of the form $y^2 \leftarrow x^3 + mx + n \ (mod\ s)$ over a prime (finite) field $\mathcal{Z}_s \leftarrow \{0, 1, 2, \ldots, s-1\}$ and base point $(\mathcal{P})$.

Step 2: Next, $\mathcal{TR}_A$ selects random secret $t \in \mathcal{Z}_s^*$ as the private key of the system and computes the public key of the system as $\mathcal{T}_{pub} = t \ . \ \mathcal{P}$.

Step 3: Further, $\mathcal{TR}_A$ chooses a cryptographic hash method represented as $h : \{0, 1\}^* \rightarrow \{0, 1\}^l$ capable of accepting any arbitrary-length input string and producing a predetermined range output as a hash value.

B. UTILITY CENTER REGISTRATION PHASE

This subsection describes the registration process for each $\mathcal{UC}_b$. In this phase, $\mathcal{TR}_A$ uses Schnorr's signature protocol [23] to generate the private keys. The subsequent process are executed between $\mathcal{UC}_b$ and $\mathcal{TR}_A$:

Step 1: At first, $\mathcal{UC}_b$ selects a unique identity $(\mathcal{ID}_{\mathcal{UC}_b})$. $\mathcal{UC}_b$ then transmits its $\mathcal{ID}_{\mathcal{UC}_b}$ to $\mathcal{TR}_A$ through a private network.

Step 2: On the arrival of $\mathcal{ID}_{\mathcal{UC}_b}$, $\mathcal{TR}_A$ selects a shared secret key $(\mathcal{K}_{\mathcal{UC}_b})$ for $\mathcal{UC}_b$ and generates random number $t_{\mathcal{UC}_b} \in \mathcal{Z}_s^*$. Thereafter, $\mathcal{TR}_A$ computes $\mathcal{T}_{\mathcal{UC}_b} \leftarrow t_{\mathcal{UC}_b}.\mathcal{P}$. Moreover, $\mathcal{TR}_A$ generates timestamp $\mathcal{T}_b$ and computes $\mathcal{PID}_{\mathcal{UC}_b} \leftarrow (\mathcal{ID}_{\mathcal{UC}_b} \| \mathcal{T}_b)$, $\mathcal{M}_{\mathcal{UC}_b} \leftarrow t_{\mathcal{UC}_b} + h(\mathcal{T}_{\mathcal{UC}_b} \| \mathcal{PID}_{\mathcal{UC}_b} \ . \ t \ (mod\ s)$. Finally, it keeps $\{\mathcal{PID}_{\mathcal{UC}_b}, \mathcal{M}_{\mathcal{UC}_b}, \mathcal{T}_{\mathcal{UC}_b}, \mathcal{K}_{\mathcal{UC}_b}\}$ in $\mathcal{TR}_A$'s memory and transmits the information $\{\mathcal{PID}_{\mathcal{UC}_b}, \mathcal{M}_{\mathcal{UC}_b}, \mathcal{T}_{\mathcal{UC}_b}, \mathcal{K}_{\mathcal{UC}_b}\}$ to $\mathcal{UC}_b$ via private channel. $\mathcal{UC}_b$ receives $\{\mathcal{PID}_{\mathcal{UC}_b}, \mathcal{M}_{\mathcal{UC}_b}, \mathcal{T}_{\mathcal{UC}_b}, \mathcal{K}_{\mathcal{UC}_b}\}$ and stores it in its own memory.

C. SMART METER REGISTRATION PHASE

This subsection describes the registration process for every $\mathcal{SM}_a$. $\mathcal{TR}_A$ uses Schnorr's signature protocol [23] to generate the private keys. The subsequent process are executed between $\mathcal{SM}_a$ and $\mathcal{TR}_A$, which are given below:

Step 1: At first, $\mathcal{SM}_a$ selects an identity $(\mathcal{ID}_{\mathcal{SM}_a})$. $\mathcal{SM}_a$ then transmits its $\mathcal{ID}_{\mathcal{SM}_a}$ to $\mathcal{TR}_A$ via private channel.

Step 2: $\mathcal{TR}_A$ receives $\mathcal{ID}_{\mathcal{SM}_a}$ and computes $t_{\mathcal{SM}_a} \leftarrow h(\mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{K}_{\mathcal{UC}_b})$, $\mathcal{T}_{\mathcal{SM}_a} \leftarrow t_{\mathcal{SM}_a}.\mathcal{P}$ and $\mathcal{M}_{\mathcal{SM}_a} \leftarrow t_{\mathcal{SM}_a} + h(\mathcal{T}_{\mathcal{SM}_a} \| \mathcal{ID}_{\mathcal{SM}_a}).t(mod\ s)$. Next, $\mathcal{TR}_A$ generates timestamp $\mathcal{T}_a$ and further calculates $\mathcal{PID}_{\mathcal{SM}_a} \leftarrow \mathcal{E}_{\mathcal{K}_{\mathcal{UC}_b}}(\mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{T}_a)$. Moreover, $\mathcal{TR}_A$ produces $\mathcal{C}_h$ and then transmits the information $\{\mathcal{PID}_{\mathcal{SM}_a}, \mathcal{M}_{\mathcal{SM}_a}, \mathcal{T}_{\mathcal{SM}_a}, \mathcal{C}_h, t_{\mathcal{SM}_a}, \{\mathcal{PID}_{uc_b} | (b = 1, 2, \ldots, n_{\mathcal{UC}_b})\}\}$ to $\mathcal{SM}_a$ via private channel.

Step 3: Upon receiving $\mathcal{C}_h$, $\mathcal{SM}_a$ calculates $\mathcal{R}_s \leftarrow \mathcal{PUF}(\mathcal{C}_h)$ and keeps the receive information, i.e., $\{\mathcal{PID}_{\mathcal{SM}_a}, \mathcal{M}_{\mathcal{SM}_a}, \mathcal{T}_{\mathcal{SM}_a}, t_{\mathcal{SM}_a}, \{\mathcal{PID}_{\mathcal{UC}_b} | (b = 1, 2, , \ldots, n_{\mathcal{UC}_b})\}\}$ in its own memory. Next, $\mathcal{SM}_a$ sends $\{\mathcal{R}_s\}$ to $\mathcal{TR}_A$ via private channel.

Step 4: On getting $\{\mathcal{R}_s\}$, $\mathcal{TR}_A$ keeps the receive information, i.e., $\{\mathcal{PID}_{\mathcal{SM}_a}, \mathcal{M}_{\mathcal{SM}_a}, \mathcal{T}_{\mathcal{SM}_a}, \{\mathcal{PID}_{\mathcal{UC}_b} | (b = 1, 2, \ldots, n_{\mathcal{UC}_b})\}\}$ *and* $(\mathcal{C}_h, \mathcal{R}_s)$ against $\mathcal{ID}_{\mathcal{SM}_a}$ in memory and encrypts it with $\mathcal{K}_{\mathcal{UC}_b}$.

D. AUTHENTICATION AND KEY AGREEMENT PHASE

In Figure 2, we outline a secure authentication process between $\mathcal{SM}_a$ and $\mathcal{UC}_b$. It can be seen that $\mathcal{SM}_a$, possessing its secret parameters, firstly constructs a login message $\mathcal{M}_1$ containing $\mathcal{M}_1 \leftarrow \{\mathcal{PID}_{\mathcal{SM}_a}, \mathcal{G}_a, \mathcal{H}_a\}$ and sends it to $\mathcal{UC}_b$. Using its secret parameters, $\mathcal{UC}_b$ verifies the authenticity of $\mathcal{H}_a$. If $\mathcal{H}_a$ is not authenticated, the process aborts. Upon successful authentication, $\mathcal{UC}_b$ constructs $\mathcal{S}_{key}$ and a challenge message $\mathcal{M}_2$ containing $\mathcal{M}_2 \leftarrow \{\mathcal{R}_b, \mathcal{W}_b, \mathcal{Y}_b, \mathcal{Z}_b, t_2\}$, and sends it back to $\mathcal{SM}_a$. $\mathcal{SM}_a$ then verifies W_b. The process aborts if $\mathcal{W}_b$ is not authenticated. Upon successful authentication of $\mathcal{W}_b$, $\mathcal{SM}_a$ constructs $\mathcal{S}_{key}$. Since the communication happens through a public channel, this mutual authentication process ensures that both parties, $\mathcal{SM}_a$ and $\mathcal{UC}_b$, can securely communicate using $\mathcal{S}_{key}$. A detailed authentication procedure of our protocol is illustrated in Fig. 3 while discussed below:

Step 1: Initially, $\mathcal{SM}_a$ produces a random number $r_{\mathcal{SM}_a} \in \mathcal{Z}_s^*$ and generate timestamp t_1, and calculates $\mathcal{G}_a \leftarrow h(r_{\mathcal{SM}_a} \| t_1)$. $\mathcal{P}$, $\mathcal{H}_a \leftarrow h(\mathcal{T}_{\mathcal{UC}_b} \| \mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{T}_{\mathcal{SM}_a} \| \mathcal{G}_a \| t_{\mathcal{SM}_a} \| t_1)$. Next, $\mathcal{SM}_a$ sends the request message $\mathcal{M}_1 \leftarrow \{\mathcal{PID}_{\mathcal{SM}_a}, \mathcal{G}_a, \mathcal{H}_a, \ t_1\}$ to $\mathcal{UC}_b$ via a public channel.

Step 2: $\mathcal{UC}_b$ receives the request messages and verifies the validity of t_1 verifying the condition $|t_1 - t_c| < \Delta\mathcal{T}?$. Then, $\mathcal{UC}_b$ computes $(\mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{T}_a) \leftarrow \mathcal{D}_{\mathcal{K}_{\mathcal{UC}_b}}(\mathcal{PID}_{\mathcal{SM}_a})$, $t_{\mathcal{SM}_a} \leftarrow h(\mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{K}_{\mathcal{UC}_b})$ and checks whether $\mathcal{H}_a \stackrel{?}{\leftarrow} h(\mathcal{T}_{\mathcal{UC}_b} \| \mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{T}_{\mathcal{SM}_a} \| \mathcal{G}_a \| t_{\mathcal{SM}_a} \| t_1)$ or not. If it fails, $\mathcal{UC}_b$ aborts the whole connection. Otherwise, it reads $(\mathcal{C}_h, \mathcal{R}_s)$ from memory against

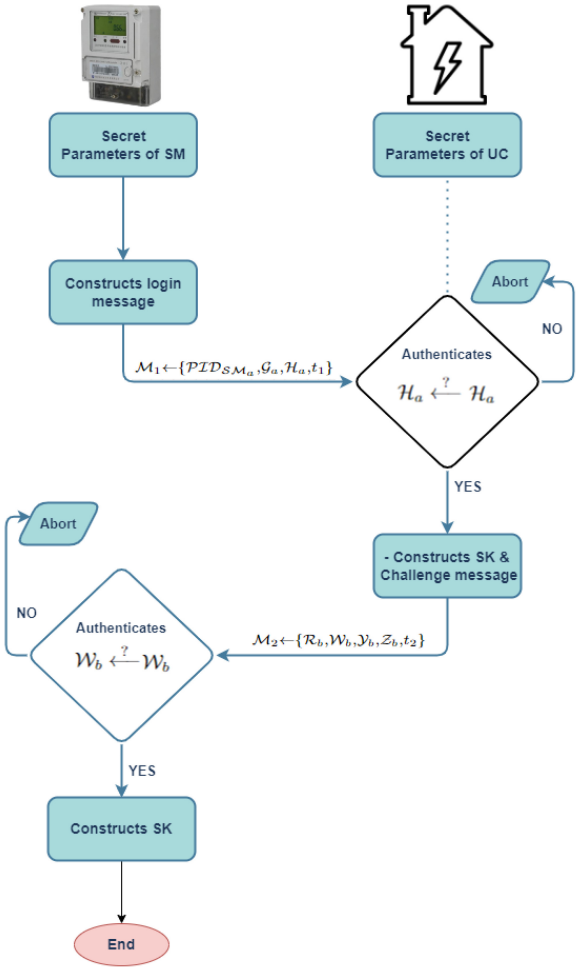

FIGURE 2. Data Flow of Authentication phase.

$\mathcal{ID}_{\mathcal{SM}_a}$, generates random number $r_{\mathcal{UC}_b} \in \mathcal{Z}_s^*$ and current timestamp t_2. Thereafter, $\mathcal{UC}_b$ computes $\mathcal{R}_b \leftarrow h(r_{\mathcal{UC}_b} \| t_2).\mathcal{P}$, $\mathcal{D}_b \leftarrow \mathcal{M}_{\mathcal{UC}_b}.\mathcal{G}_a$, $\mathcal{W}_b \leftarrow h(\mathcal{G}_a \| \mathcal{T}_{\mathcal{UC}_b} \| \mathcal{D}_b \| \mathcal{R}_b \| t_1 \| t_2)$, $\mathcal{F}_b \leftarrow h(r_{\mathcal{UC}_b} \| t_2).(\mathcal{T}_{\mathcal{SM}_a} + h(\mathcal{T}_{\mathcal{SM}_a} \| \mathcal{ID}_{\mathcal{SM}_a}).\mathcal{T}_{pub})$. Furthermore, $\mathcal{UC}_b$ generates timestamp t_3^{new} and calculates $\mathcal{Y}_b \leftarrow h(\mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{K}_{\mathcal{UC}_b}) \oplus t_3^{new}$, $\mathcal{PID}^{new}_{\mathcal{SM}_a} \leftarrow \mathcal{E}_{\mathcal{K}_{\mathcal{UC}_b}}(\mathcal{ID}_{\mathcal{SM}_a} \| t_3^{new})$, $\mathcal{Z}_b \leftarrow h(h(\mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{K}_{\mathcal{UC}_b}) \| t_3^{new}) \oplus (\mathcal{PID}^{new}_{\mathcal{SM}_a} \| \mathcal{C}_h)$. Finally, $\mathcal{UC}_b$ establishes $\mathcal{S}_{key} \leftarrow h(\mathcal{F}_b \| \mathcal{D}_b \| \mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{PID}_{\mathcal{UC}_b} \| \mathcal{R}_s \| t_3^{new})$ and transmits the response message $\mathcal{M}_2 \leftarrow \{\mathcal{R}_b, \mathcal{W}_b, \mathcal{Y}_b, \mathcal{Z}_b, t_2\}$ to $\mathcal{SM}_a$ via open channel.

Step 3: $\mathcal{SM}_a$ receives the response message from $\mathcal{UC}_b$ and verifies the validity of timestamp t_2 through $|t_2 - t_c| < \Delta\mathcal{T}$?. If authentic, $\mathcal{SM}_a$ proceeds to compute $\mathcal{D}_a \leftarrow h(r_{\mathcal{SM}_a} \| t_1).(\mathcal{T}_{\mathcal{UC}_b} + h(\mathcal{T}_{\mathcal{UC}_b} \| \mathcal{PID}_{\mathcal{UC}_b}).\mathcal{T}_{pub})$ and check whether $\mathcal{W}_b \stackrel{?}{\leftarrow} h(\mathcal{G}_a \| \mathcal{T}_{\mathcal{UC}_b} \| \mathcal{D}_b \| \mathcal{R}_b \| t_1 \| t_2)$ or not. If it fails, $\mathcal{SM}_a$ then aborts the whole connection. Otherwise, $\mathcal{SM}_a$ considers the response message to be non-tampered and acknowledged from the genuine $\mathcal{UC}_b$. Moreover, $\mathcal{SM}_a$ calculates $\mathcal{F}_a \leftarrow \mathcal{M}_{\mathcal{SM}_a}.\mathcal{R}_b$, $t_3^{new} \leftarrow t_{\mathcal{SM}_a} \oplus \mathcal{Y}_b$, $(\mathcal{PID}^{new}_{\mathcal{SM}_a} \| \mathcal{C}_h) \leftarrow h(t_{\mathcal{SM}_a} \| t_3^{new}) \oplus \mathcal{Z}_b$, $\mathcal{R}_s \leftarrow \mathcal{PUF}(\mathcal{C}_h)$ and $\mathcal{S}_{key} \leftarrow h(\mathcal{F}_a \| \mathcal{D}_a \| \mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{PID}_{\mathcal{UC}_b} \| \mathcal{R}_s \| t_3^{new})$. Lastly, $\mathcal{UC}_b$ replaces $\mathcal{PID}_{\mathcal{SM}_a}$ with $\mathcal{PID}^{new}_{\mathcal{SM}_a}$, and establishes session key $\mathcal{S}_{key}$. After successful authentication, both $\mathcal{SM}_a$ and $\mathcal{UC}_b$ use the established $\mathcal{S}_{key}$ for their secure connections in the future.

E. DYNAMIC NODE UPDATE PHASE

Assume a new $\mathcal{SM}_a^{new}$ needs to be installed in the existing system. The subsequent steps are executed between $\mathcal{SM}_a^{new}$ and $\mathcal{TR}_A$:

Step 1: $\mathcal{SM}_a^{new}$ firstly generates its identity $(\mathcal{ID}^{new}_{\mathcal{SM}_a})$ and then sent it to $\mathcal{TR}_A$ via a private network.

Step 2: When the registration request is received, $\mathcal{TR}_A$ computes $t^{new}_{\mathcal{SM}_a} \leftarrow h(\mathcal{ID}^{new}_{\mathcal{SM}_a} \| \mathcal{K}_{\mathcal{UC}_b})$, $\mathcal{T}^{new}_{\mathcal{SM}_a} \leftarrow t^{new}_{\mathcal{SM}_a}.\mathcal{P}$ and $\mathcal{M}^{new}_{\mathcal{SM}_a} \leftarrow t^{new}_{\mathcal{SM}_a} + h(\mathcal{T}^{new}_{\mathcal{SM}_a} \| \mathcal{ID}^{new}_{\mathcal{SM}_a}).t \ (mod\ s)$.

Step 3: $\mathcal{TR}_A$ selects $\mathcal{T}_a^{new}$ and further calculates $\mathcal{PID}^{new}_{\mathcal{SM}_a} \leftarrow \mathcal{E}_{\mathcal{K}_{\mathcal{UC}_b}}(\mathcal{ID}^{new}_{\mathcal{SM}_a} \| \mathcal{T}_a^{new})$. Next, $\mathcal{TR}_A$ produce $\mathcal{C}_h^{new}$ and sends all the information, i.e., $\{\mathcal{PID}^{new}_{\mathcal{SM}_a}, \mathcal{M}^{new}_{\mathcal{SM}_a}, \mathcal{T}^{new}_{\mathcal{SM}_a}, \mathcal{C}_h^{new}, t^{new}_{\mathcal{SM}_a}, \{\mathcal{PID}_{\mathcal{UC}_b} | (b = 1, 2, \ldots, n_{\mathcal{UC}_b})\}\}$ to $\mathcal{SM}_a^{new}$ through private channel.

Step 4: After receiving the information, $\mathcal{SM}_a^{new}$ computes $\mathcal{R}_s^{new} \leftarrow \mathcal{PUF}(\mathcal{C}_h^{new})$ and stores all data in memory for mutual authentication with $\mathcal{UC}_b$ when it is needed. Next, $\mathcal{SM}_a^{new}$ sends $\mathcal{R}_s^{new}$ to $\mathcal{TR}_A$ via a private channel.

Step 5: Finally, $\mathcal{TR}_A$ stores $\{\mathcal{PID}^{new}_{\mathcal{SM}_a}, \mathcal{M}^{new}_{\mathcal{SM}_a}, \mathcal{T}^{new}_{\mathcal{SM}_a}, \{\mathcal{PID}_{\mathcal{UC}_b} | (b = 1, 2, \ldots, n_{\mathcal{UC}_b})\}\}$ and $(\mathcal{C}_h^{new}, \mathcal{R}_s^{new})$ against $\mathcal{ID}^{new}_{\mathcal{SM}_a}$ in memory and encrypts it with $\mathcal{K}_{\mathcal{UC}_b}$.

F. OBSERVING CONDITIONAL-PRIVACY

Our proposed protocol ensures conditional privacy by incorporating several key mechanisms that protect user identities and usage data under specific conditions. First, we utilize pseudo identities ($\mathcal{PID}_{\mathcal{SM}_a}$ and $\mathcal{PID}_{\mathcal{UC}_b}$) and temporary identities ($\mathcal{PID}^{new}_{\mathcal{SM}_a}$) during the authentication process, ensuring that real identities are never exposed during communication. This approach prevents unauthorized tracking and profiling of smart meter activities. Second, integrating the Physical Unclonable Function ($\mathcal{PUF}$) provides unique, unclonable challenge-response pairs for each smart meter, enhancing security against physical tampering and cloning attacks. Additionally, all sensitive information, including identities and session keys, is encrypted using symmetric encryption ($\mathcal{E}_{\mathcal{K}_{\mathcal{UC}_b}}$) and hashed ($h(.)$) before transmission. This guarantees that the data remains confidential and secure even if communications are intercepted. These measures collectively ensure that our protocol maintains user privacy

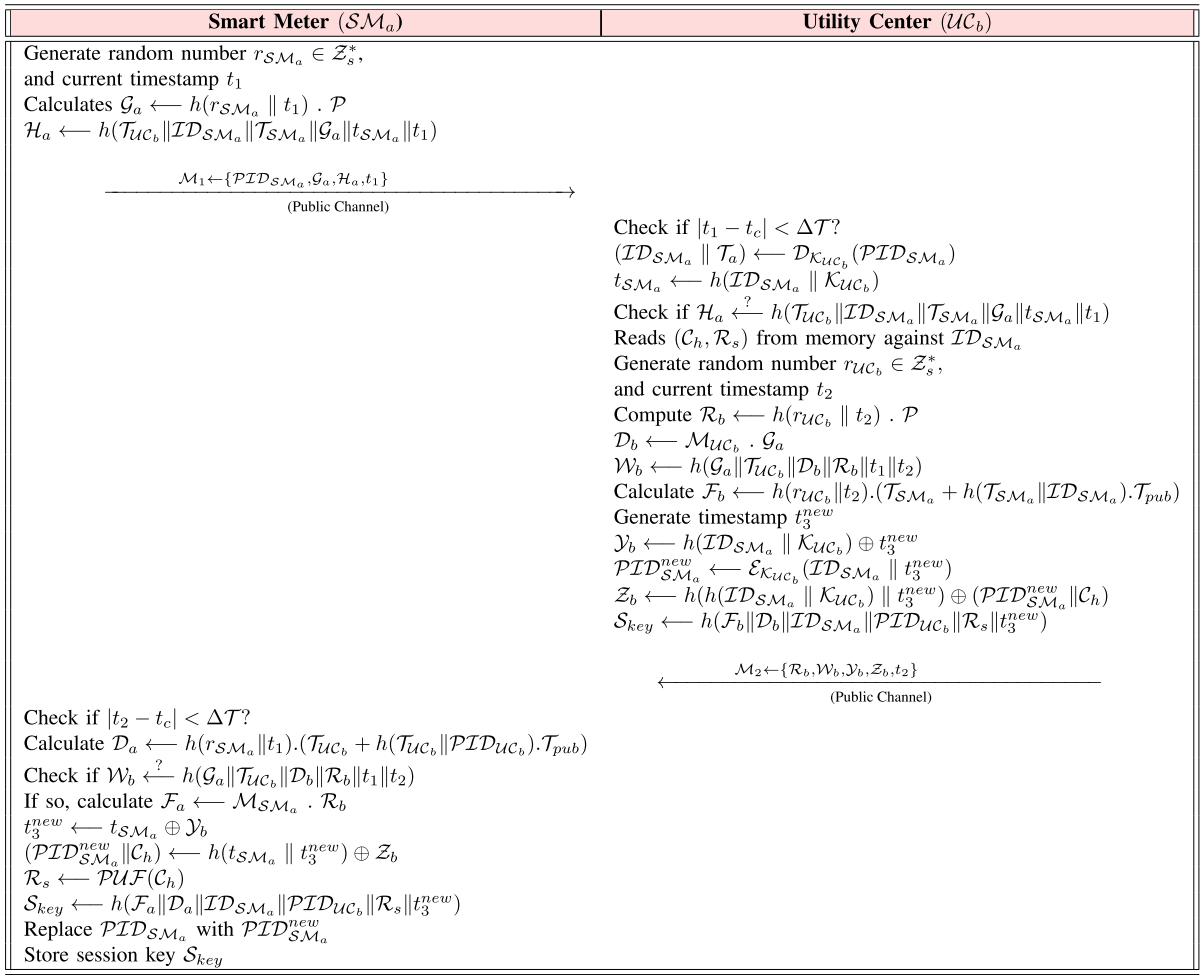

FIGURE 3. Authentication and Key Agreement Phase of Proposed Protocol.

and aligns with the necessary privacy requirements, thereby enhancing the overall security and robustness of the SPG system.

IV. SECURITY ANALYSIS

In this section, both formal and informal analysis are discussed to showcase how our protocol effectively mitigates various well-known attacks, as elaborated below:

A. FORMAL ANALYSIS

In this subsection, we describe the formal analysis of our protocol by using the most extensively used mathematical ROR model [24]. This model is used to evaluate the proposed protocol's security. The formal analysis based on [24] has achieved prominence in examining for evaluating the privacy and security of numerous key agreement protocols discussed in the literature [25]. Under this model, the $\mathcal{A}_d$ associates with the t-th instance Π^t of an executing participant, such as ($\mathcal{SM}_a$ or $\mathcal{UC}_b$). $\mathcal{A}_d$ has access to different queries required for simulating a threat, which is illustrated in Table 3. In the subsequent section, the elements of the ROR model are discussed below

1) *Entities:* The associated entities in our protocol are $\mathcal{SM}_a$ or $\mathcal{UC}_b$. The instances of $\mathcal{SM}_a$ and $\mathcal{UC}_b$, such as t_1 and t_2, are denoted as $\Pi_{\mathcal{SM}_a}^{t_1}$ and $\Pi_{\mathcal{UC}_b}^{t_2}$, respectively. These instances are also referred to as oracles
2) *Accepted State:* Consider the scenario where Π^t enters an approved state after receiving the last legitimate message. The session identification (*Sid*) of Π^t for the going session is created after reorganizing the received and sent messages by Π^t in a sequential way.
3) *Partnering:* Π^{t_1} and Π^{t_2} are partners among each other when the subsequent conditions are fulfilled:
   - Both instances Π^{t_1} and Π^{t_2} are in accepted states.
   - Both instances Π^{t_1} and Π^{t_2} are mutual partners among each other.
   - Both instances share an identical *Sid*, and they mutually authenticate.
4) *Freshness:* $\Pi_{\mathcal{SM}_a}^{t_1}$ or $\Pi_{\mathcal{UC}_b}^{t_2}$ is known as fresh when $\mathcal{S}_{key}$ among $\mathcal{SM}_a$ and $\mathcal{UC}_b$ is not revealed to $\mathcal{A}_d$ by applying the *Reveal*(Π^t) query illustrated in Table 3. The semantic security of our protocol ($\mathcal{P}$) is explained in Definition 1.

TABLE 3. Different queries with their significance.

Query	Significance
$Corrupt(\mathcal{SM}_a)$	$\mathcal{A}_d$ can obtain all the credentials kept from compromised $\mathcal{SM}_a$'s memory by using this query.
$Execute(\mathcal{SM}_a, \mathcal{UC}_b)$	It helps $\mathcal{A}_d$ to intercept all the communication messages transmitted between $\mathcal{SM}_a$ and $\mathcal{UC}_b$.
$Reveal(\Pi^t)$	It helps $\mathcal{A}_d$ to achieve $\mathcal{S}_{key}$ generated among Π^t and its companion.
$Test(\Pi^t)$	It allows $\mathcal{A}_d$ to request Π^t for $\mathcal{S}_{key}$, and Π^t outputs probabilistically the outcome of a flipped unbiased coin c^n.

Definition 1: If $\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d}(pol_t)$ is the benefit of an $\mathcal{A}_d$ operating within polynomial time (pol_t) in compromising the secure semantic privacy and security of $\mathcal{P}$ to derive $\mathcal{S}_{key}$ between $\mathcal{SM}_a$ and $\mathcal{UC}_b$, $\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d}(pol_t) = |2.\mathcal{P}_r[b' = b] - 1|$, where b represents the original bits and b' represents the guessed bits. Furthermore, the Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP) and the one-way hash method are explained in Definitions 2 and 3, respectively, to examine the proposed protocol's security.

Definition 2: A collision-resistant one-way hash method is a deterministic function, such as $h : \{0,1\}^* \longrightarrow \{0,1\}^l$. It operates by taking an input string $x \in \{0,1\}^*$ of arbitrary length and producing a fixed-size output, typically l bits, as its hash value, denoted as $h(x) \in \{0,1\}^l$. Let $\mathcal{A}_d$ want to identify a hash method. Then, the benefit of $\mathcal{A}_d$ in finding the one-way hash collision is defined by $\mathcal{ADV}^{Hash}_{\mathcal{A}_d}(t_r) = \mathcal{P}_r[(y_1, y_2) \leftarrow_r \mathcal{A}_d : y_1 \neq y_2, h(y_1) = h(y_2)]$. Here, $\mathcal{P}_r$ denotes the chance of occurrence the random event $\mathcal{RE}$, and $(y_1, y_2) \leftarrow_r \mathcal{A}_d$ indicates that pair is chosen at random by $\mathcal{A}_d$. Suppose an (η, t_r) $\mathcal{A}d$ attempts to breach the one-way hash method h(.). This signifies that "$\mathcal{A}_d$'s runtime is at most t_r with $\mathcal{ADV}^{Hash}_{\mathcal{A}_d}(t_r) \leq \eta''$.

Definition 3: Let $\mathcal{P} \in \mathcal{E}_s(m, n)$ be an elliptic curve base/public point on the elliptic curve $\mathcal{E}_s(m, n)$. Our protocol defines a quadruple ($\mathcal{P}$, $u.\mathcal{P}$, $v.\mathcal{P}$, $w'.\mathcal{P}$), deciding whether $w' = u.v$ or a uniform value. We demonstrate the semantic strength of the proposed protocol in Theorem 1.

Theorem 1: Suppose an $\mathcal{A}_d$ is running in pol_t against $\mathcal{P}$. If q_{hash}, $|Hash|$ and $\mathcal{ADV}^{ECDDHP}_{\mathcal{A}_d}(pol_t)$ indicate the counts of one-way hash queries, the hash method's range space, and $\mathcal{A}_d$'s benefit in compromising ECDDHP in pol_t (refer to Definition 3), respectively, then $\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d}(pol_t) \leq \frac{q^2_{hash}}{|Hash|} + 2\mathcal{ADV}^{ECDDHP}_{\mathcal{A}_d}(pol_t)$.

Proof: We have presented the analogous proof of this theorem as done in other authentication protocols [25]. This proof explains the four games $Game_i$ where (i = 0, 1, 2, 3). We define $\mathcal{SUC}^{Game_i}_{\mathcal{A}_d}$ as the scenario wherein $\mathcal{A}_d$ accurately guesses all the random bits c_m in $Game_i$, and also advantage of $\mathcal{A}_d$ in successfully winning $Game_i$ as $\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d, Game_i} = \mathcal{P}_r[\mathcal{SUC}^{Game_i}_{\mathcal{A}_d}]$. Further elaboration on these games is given in the subsequent discussion:

- $Game_0$: Typically, the initial game ($Game_0$) is the same as the protocol's execution in the ROR model. Hence, according to the semantic privacy of SPG explained in Definition 1, we can conclude that

$$\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d}(pol_t) = |2.\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d, Game_0} - 1| \quad (1)$$

- $Game_1$: The eavesdropping attack (EA) is established in this $Game_1$, enabling $\mathcal{A}d$ to effortlessly capture all transmitted messages $\mathcal{M}_1 \leftarrow \{\mathcal{PID}_{\mathcal{SM}_a}, \mathcal{G}_a, \mathcal{H}_a, t_1\}$ and $\mathcal{M}_2 \leftarrow \{\mathcal{R}_b, \mathcal{W}_b, \mathcal{Y}_b, \mathcal{Z}_b, t_2\}$ during the execution of the authentication procedure by using *Execute* query as discussed in Table 3. At last, $\mathcal{A}_d$ can perform the queries *Test* and *Reveal* to verify that the computed $\mathcal{S}_{key}$ between $\mathcal{SM}_a$ and $\mathcal{UC}_b$ is authentic or a randomly generated secret. The authenticated $\mathcal{S}_{key}$ is $h(\mathcal{F}_a \| \mathcal{D}_a \| \mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{PID}_{\mathcal{UC}_b} \| \mathcal{R}_s \| t_3^{new})$, $h(\mathcal{F}_b \| \mathcal{D}_b \| \mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{PID}_{\mathcal{UC}_b} \| \mathcal{R}_s \| t_3^{new})$. It's crucial to emphasize that the security of $\mathcal{S}key$ depends on both temporal secrets, such as $r_{\mathcal{SM}_a}$ and $r_{\mathcal{UC}_b}$ as well as long-term shared secrets $\mathcal{M}_{\mathcal{SM}_a}$ and $\mathcal{M}_{\mathcal{UC}_b}$. It cannot be recognized through eavesdropping on all messages $\mathcal{M}_1$ and $\mathcal{M}_2$. It is not possible for $\mathcal{A}_d$ to determine $\mathcal{S}_{key}$ without this information. In conclusion, EA does not increase the probability of winning $Game_1$. Hence, both $Game_0$ and $Game_1$ become indistinguishable. Therefore, we arrive at the following conclusion:

$$\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d, Game_1} = \mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d, Game_0} \quad (2)$$

- $Game_2$: This game contains the simulation of one-way hash queries. t_1 and $\mathcal{G}_a$ are random in the communication message $\mathcal{M}_1 \leftarrow \{\mathcal{PID}_{\mathcal{SM}_a}, \mathcal{G}_a, \mathcal{H}_a, t_1\}$. Similarly, in another communication message $\mathcal{M}_2 \leftarrow \{\mathcal{R}_b, \mathcal{W}_b, \mathcal{Y}_b, \mathcal{Z}_b, t_2\}$ the components $\mathcal{R}_b$, $\mathcal{W}_b$, and t_2, are random because these contain current timestamps and random numbers. So, no collision occurs when $\mathcal{A}_d$ implements one-way hash queries. Both $Game_1$ and $Game_2$ are practically identical, differing only in the simulation of one-way hash queries within $Game_2$, as inferred from the implications of the birthday paradox. So, we get

$$|\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d, Game_2} - \mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d, Game_1}| \leq \frac{q^2_{hash}}{2|Hash|} \quad (3)$$

- $Game_3$: In this particular game, the $Corrupt(\mathcal{SM}_a)$ query has been performed. Consequently, owing to the execution of this query, $\mathcal{A}_d$ gains access to all the fetched information such as $\mathcal{PID}_{\mathcal{SM}_a}, \mathcal{M}_{\mathcal{SM}_a}, \mathcal{T}_{\mathcal{SM}_a}, t_{\mathcal{SM}_a}$, $\{\mathcal{PID}_{\mathcal{UC}_b} | (b = 1, 2, \ldots, n_{\mathcal{UC}_b})\}$ from the compromised $\mathcal{SM}_a$. Moreover, $\mathcal{A}_d$ will have all the captured messages $\mathcal{M}_1$ and $\mathcal{M}_2$. To derive $\mathcal{S}_{key} \leftarrow h(\mathcal{F}_a \| \mathcal{D}_a \| \mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{PID}_{\mathcal{UC}_b} \| \mathcal{R}_s \| t_3^{new}) \leftarrow h(\mathcal{F}_b \| \mathcal{D}_b \| \mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{PID}_{\mathcal{UC}_b} \| \mathcal{R}_s \| t_3^{new}) \leftarrow \mathcal{S}_{key}$ shared

TABLE 4. Specifications of implementation devices.

Items	Arduino Device	Desktop Machine
Platform	_	Windows
IDE	Arduino IDE	PyCharm
RAM	2 KB (ATmega 328)	16 GB
Processor	Microcontroller:(ATmega 328)	intel Core i7
Clock Speed	16 (MHZ	2.9 (GHZ)
Programming Language	C++	Python
Library	CryptoPP	PyCryptoDome

between $\mathcal{SM}_a$ and $\mathcal{UC}_b$. $\mathcal{A}_d$ needs to compute $\mathcal{F}_a(=\mathcal{F}_b)$ and $\mathcal{D}_a(=\mathcal{D}_b)$. This situation results in the computation of $h(r_{\mathcal{SM}_a}\|t_1)$ and $h(r_{\mathcal{UC}_b}\|t_2)$, which is computationally costly owing to the complexity of ECDDHP within a pol_t. As $Game_2$ and $Game_3$ are identical, except for the incorporation of the $Corrupt(\mathcal{SM}a)$ query and ECDDHP, it consequently follows that

$$|\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d,Game_3} - \mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d,Game_2}| \leq \mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d}(pol_t) \quad (4)$$

Next, all of the related queries to the games mentioned above are performed. Estimating the random bit c_m only remains after the *Test* and *Reveal* queries are executed. So, we have

$$\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d,Game_3} = \frac{1}{2} \quad (5)$$

Using equations (1), (2), and (5), we have the subsequent derivation:

$$\begin{aligned}\frac{1}{2}.\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d}(pol_t) &= |\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d,Game_0} - \frac{1}{2}| \\ &= |\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d,Game_1} - \mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d,Game_3}| \\ &\leq |\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d,Game_1} - \mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d,Game_2}| \\ &+|\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d,Game_2} - \mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d,Game_3}| \quad (6)\end{aligned}$$

Furthermore, equations (3), (4), and (6) give the subsequent conclusion:

$$\frac{1}{2}.\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d}(pol_t) \leq \frac{q^2_{hash}}{2|Hash|} + \mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d}(pol_t) \quad (7)$$

Lastly, multiply both sides of the equation (7) by 2. Then, we get:

$$\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d}(pol_t) \leq \frac{q^2_{hash}}{|Hash|} + 2\mathcal{ADV}^{\mathcal{P}}_{\mathcal{A}_d}(pol_t) \quad (8)$$

B. INFORMAL ANALYSIS

This subsection presents an informal analysis to demonstrate our protocol's stability and security against all well-known threats. The explanation of the informal analysis is elaborated in the below subsections:

1) SMART METER IMPERSONATION ATTACK (1A)

Suppose, $\mathcal{A}_d$ tries to generate the login message $\mathcal{M}_1$. However, to generate $\mathcal{H}_a \leftarrow h(\mathcal{T}_{\mathcal{UC}_b}\|\mathcal{ID}_{\mathcal{SM}_a}\|\mathcal{T}_{\mathcal{SM}_a}\|\mathcal{G}_a\|t_{\mathcal{SM}_a}\|t_1)$, $\mathcal{A}_d$ needs to know the real identity $\mathcal{ID}_{\mathcal{SM}_a}$ of $\mathcal{SM}_a$. It is worth noticing that $\mathcal{A}_d$ is unable to find $\mathcal{ID}_{\mathcal{SM}_a}$ of an $\mathcal{SM}_a$ because $\mathcal{ID}_{\mathcal{SM}_a}$ is not sent in plain text across the channel. Even if $\mathcal{A}_d$ physically captures the $\mathcal{SM}_a$'s memory, he cannot obtain $\mathcal{ID}_{\mathcal{SM}_a}$. As $\mathcal{ID}_{\mathcal{SM}_a}$ is not stored in $\mathcal{SM}_a$'s memory. Therefore, the proposed protocol provides resistance against smart meter impersonation attacks.

2) UTILITY CENTER IMPERSONATION ATTACK (2A)

Suppose that $\mathcal{A}_d$ attempts to generate the message $\mathcal{M}_2$ using valid credentials. Neverthless, $\mathcal{A}_d$ needs to know the $\mathcal{UC}b$'s shared key ($\mathcal{K}_{\mathcal{UC}_b}$) to decrypt $\mathcal{PID}_{\mathcal{SM}_a}$. Therefore, $\mathcal{A}_d$ is unaware of the $\mathcal{UC}b$'s shared key and cannot identify the real identity $\mathcal{ID}_{\mathcal{SM}_a}$. Moreover, $\mathcal{A}_d$ also needs to know all the secret parameters like $\mathcal{D}_b$, $t_{\mathcal{UC}_b}$, $r_{\mathcal{UC}_b}$, and t. As a result, $\mathcal{A}_d$ is unaware of all these secret parameters and cannot generate this response message. Hence, our protocol offers robustness against $\mathcal{UC}_b$ impersonation attacks

3) MAN-IN-THE-MIDDLE ATTACK (3A)

Suppose an $\mathcal{A}_d$ intercepts all communication messages (i.e., $\mathcal{M}_1$ and $\mathcal{M}_2$) among the entities during the authentication phase. In order to alter the message $\mathcal{M}_1$, $\mathcal{A}_d$ has to modify $\mathcal{G}_a$, which requires knowledge of the random number $r_{\mathcal{SM}_a} \in \mathcal{Z}^*_s$. On the other hand, to alter $\mathcal{M}_2$, $\mathcal{A}_d$ needs to change $\mathcal{W}_b$, which requires the knowledge of $\mathcal{D}_b$, $\mathcal{T}_{\mathcal{UC}_b}$, $r_{\mathcal{UC}_b}$, and $\mathcal{M}_{\mathcal{UC}_b}$. As a result, it is clear that $\mathcal{A}_d$ cannot change the message without knowing the shared secret key, random secrets, and current timestamps, respectively. Hence, our protocol provides resistance against MITM attacks.

4) MUTUAL AUTHENTICATION AND SESSION KEY ESTABLISHMENT (4A)

In our protocol, $\mathcal{UC}_b$ verifies $\mathcal{H}_a \stackrel{?}{\leftarrow} h(\mathcal{T}_{\mathcal{UC}_b}\|\mathcal{ID}_{\mathcal{SM}_a}\|\mathcal{T}_{\mathcal{SM}_a}\|\mathcal{G}_a\|t_{\mathcal{SM}_a}\|t_1)$ to authenticate $\mathcal{SM}_a$. In contrast, $\mathcal{SM}_a$ validates $\mathcal{UC}_b$ on $\mathcal{W}_b \stackrel{?}{\leftarrow} h(\mathcal{G}_a\|\mathcal{T}_{\mathcal{UC}_b}\|\mathcal{D}_b\|\mathcal{R}_b\|t_1\|t_2)$. Since both $\mathcal{SM}_a$ and $\mathcal{UC}_b$ authenticate each other before rendering $\mathcal{S}_{key}$. Therefore, mutual authentication is achieved in our protocol. It is to be noted that both participants establish $\mathcal{S}_{key}$ only after successful authentication. Hence, the designed protocol offers mutual authentication and key agreement.

5) SMART METER ANONYMITY AND UNTRACEABILITY (5A)

In the proposed protocol, $\mathcal{A}_d$ cannot render the actual identity $\mathcal{ID}_{\mathcal{SM}_a}$ of $\mathcal{SM}_a$ Moreover, $\mathcal{UC}_b$ does not directly exchange the updated temporary identity $\mathcal{PID}^{new}_{\mathcal{SM}_a}$ to $\mathcal{SM}_a$ via public channel. Since the actual identity is not leaked to any $\mathcal{A}_d$ and no $\mathcal{A}_d$ can identify any particular $\mathcal{SM}_a$ analyzing two distinct sessions. Consequently, our protocol ensures the untraceability and anonymity of the smart meters.

6) PHYSICAL AND CLONING ATTACKS (6A)

An $\mathcal{A}_d$ can physically access $\mathcal{SM}_a$ to compromise/tamper it. However, in the proposed protocol, if $\mathcal{A}_d$ tries to make such an attempt with the memory of $\mathcal{SM}_a$, then it will change the behavior of PUF. As a result, the compromised/tampered $\mathcal{SM}_a$ will become a meaningless device for $\mathcal{A}_d$ as the embedded PUF fails to generate accurate results during the execution of the protocol. Moreover, $\mathcal{UC}_b$ can easily trace such attempts verifying the output of PUF. Hence, our protocol is secure from physical and cloning attacks.

7) OFFERS PERFECT FORWARD SECRECY (7A)

In the proposed protocol, the session key $\mathcal{S}_{key} \longleftarrow h(\mathcal{F}_a \| \mathcal{D}_a \| \mathcal{ID}_{\mathcal{SM}_a} \| \mathcal{PID}_{\mathcal{UC}_b} \| \mathcal{R}_s \| t_3^{new})$ incorporates $\mathcal{D}_a$ and $\mathcal{PID}_{\mathcal{UC}_b}$, which are refreshed and unique for each session. As a result, once a session ends, the previous session key cannot be retrieved, even if the current session key is compromised. This prevents $\mathcal{A}_d$ from recovering any past session keys using the compromised values of $\mathcal{D}_a$ and $\mathcal{PID}_{\mathcal{UC}_b}$, ensuring perfect forward secrecy.

V. PERFORMANCE COMPARISON

This section evaluates the performance of the proposed protocol and conducts a comparative analysis against several other protocols, including those by Irshad et al. [12], Safkhani et al. [26], Park et al. [27], Kumari and Singh [17], and Rostampour et al. [18].

A. EXPERIMENTAL SETUP

The proposed protocol, as well as the related ones, comprise two fundamental entities, namely

1) $\mathcal{SM}_a$ and
2) $\mathcal{UC}_b$.

The registration phase of an authentication protocol is a singular process, and the dynamic node update process occurs upon request of $\mathcal{SM}_a$. Hence, we have eliminated these two phases when calculating communication and computation costs. However, all the essential cryptographic primitives that are used at $\mathcal{SM}_a$ and $\mathcal{UC}_b$ sides are implemented on an Arduino device and desktop machine, respectively. Table 4 showcases the specifications of both a desktop machine and an Arduino device. Furthermore, Table 5 presents the computational time necessary for various cryptographic operations, such as encryption/decryption, hash method, point multiplication, and point addition, based on their respective implementation environments.

B. COMPUTATION COST

For evaluating the computational overheads of both the proposed and comparable protocols, we have taken into account the cryptographic primitives outlined in Table 5. The computation cost is calculated for each entity, including $\mathcal{SM}_a$ and $\mathcal{UC}_b$. In the proposed protocol, the point multiplication is performed once, while the hash method

TABLE 5. Execution time cryptographic operations.

Operations	*Execution Time*	
	Arduino Device	*Desktop Machine*
$\mathcal{T}_{hash}$	1.184 ms	0.0013 ms
$\mathcal{T}_{se/d}$	0.848 ms	0.0023 ms
$\mathcal{T}_{pmul}$	0.424 ms	0.0017 ms
$\mathcal{T}_{padd}$	0.233 ms	0.0011 ms

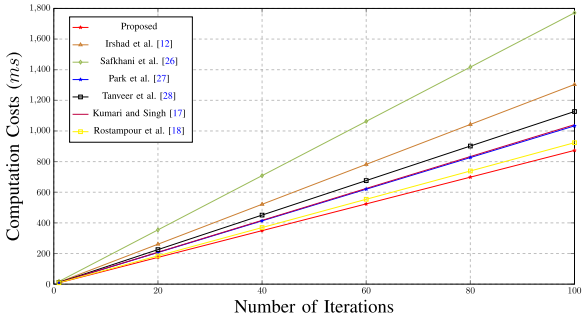

FIGURE 4. Comparison on Computation Costs.

is performed seven times at $\mathcal{SM}_a$ side. Likewise, the symmetric encryption/decryption, hash function, and point multiplication are executed two, ten, and one time, respectively, at $\mathcal{UC}_b$ side. While considering the running time of cryptographic primitives presented in Table 5, the computation cost of $\mathcal{SM}_a$ is 8.712 ms, and the computation cost of $\mathcal{UC}_b$ is 0.0193 ms. Consequently, the accumulative computation overhead of our protocol is 8.7313 ms. The computation cost of comparable protocols such as [12], [17], [18], [26], [27], [28] is computed using a similar methodology and detailed in Fig. 4 and Table 6. It is obvious from the results that the proposed protocol incurs 23.8879% less computation cost as compared to related protocols, which indicates its efficiency.

C. COMMUNICATION COST

The communication cost denotes the number of bits needed to transmit messages between the entities involved in an authentication protocol to complete the authentication procedure. To compute the communication cost of our and related protocols, we consider the following assumptions: XOR operation, password, random number, identity, time stamp, and concatenation take 160 bits. However, point multiplication takes 320 bits. Furthermore, symmetric encryption/decryption, hash function, and private/public keys take 128,256 and 256 bits, respectively. In proposed protocol, $\mathcal{SM}_a$ and $\mathcal{UC}_b$, transmit two communication messages (i.e., $\mathcal{M}_1$ and $\mathcal{M}_2$) with each other. The overall number of bits needed to sent the message $\mathcal{M}_1 \leftarrow \{\mathcal{PID}_{\mathcal{SM}_a}, \mathcal{G}_a, \mathcal{H}_a, t_1\}$ are $(160 + 320 + 320 + 160) = 960$ bits. Likewise, the transmission of $\mathcal{M}_2 \leftarrow \{\mathcal{R}_b, \mathcal{W}_b, \mathcal{Y}_b, \mathcal{Z}_b, t_2\}$ requires $(320 + 320 + 256 + 256 + 160) = 1312$ bits. Thus, the overall communication costs of our protocol are (960

TABLE 6. Computation costs and energy consumption comparison.

Protocols	$\mathcal{SM}_a$	$\mathcal{UC}_b$	Total Cost	Energy Consumption
Proposed	$7\mathcal{T}_{hash}$ + $1\mathcal{T}_{pmul} \approx 8.712$ ms	$10\mathcal{T}_{hash}$ + $1\mathcal{T}_{pmul}$ + $2\mathcal{T}_{se/d} \approx 0.0193$ ms	8.7313 ms	94.9965 mJ
Irshad et al. [12]	$11\mathcal{T}_{hash} \approx 13.024$ ms	$8\mathcal{T}_{hash} \approx 0.0104$ ms	13.0344 ms	141.8142 mJ
Safkhani et al. [26]	$11\mathcal{T}_{hash}$ + $7\mathcal{T}_{pmul}$ + $2\mathcal{T}_{se/d} \approx 17.688$ ms	$9\mathcal{T}_{hash}$ + $5\mathcal{T}_{pmul}$ + $2\mathcal{T}_{se/d} \approx 0.0248$ ms	17.7128 ms	192.7152 mJ
Park et al. [27]	$8\mathcal{T}_{hash}$ + $2\mathcal{T}_{pmul} \approx 10.32$ ms	$7\mathcal{T}_{hash}$ + $2\mathcal{T}_{pmul} \approx 0.0125$ ms	10.3325 ms	112.4176 mJ
Tanveer et al. [28]	$7\mathcal{T}_{hash}$ + $1\mathcal{T}_{pmul}$ + $3\mathcal{T}_{se/d} \approx 11.256$ ms	$6\mathcal{T}_{hash}$ + $3\mathcal{T}_{se/d} \approx 0.0147$ ms	11.2707 ms	122.6252 mJ
Kumari and Singh [17]	$6\mathcal{T}_{hash}$ + $6\mathcal{T}_{pmul}$ + $5\mathcal{T}_{padd} \approx 10.389$ ms	$6\mathcal{T}_{hash}$ + $6\mathcal{T}_{pmul}$ + $5\mathcal{T}_{padd} \approx 0.0218$ ms	10.4108 ms	113.2695 mJ
Rostampour et al. [18]	$5\mathcal{T}_{hash}$ + $6\mathcal{T}_{pmul} \approx 9.224$ ms	$4\mathcal{T}_{hash}$ + $3\mathcal{T}_{pmul} \approx 0.0103$ ms	9.2343 ms	100.9641 mJ

TABLE 7. Communication costs comparison.

Protocols	$\mathcal{SM}_a$	$\mathcal{UC}_b$	Total Cost
Proposed	960 bits	1312 bits	2272 bits
Irshad et al. [12]	1024 bits	1024 bits	2048 bits
Safkhani et al. [26]	832 bits	416 bits	1248 bits
Park et al. [27]	1280 bits	1120 bits	2400 bits
Tanveer et al. [28]	1248 bits	672 bits	1920 bits
Kumari and Sing [17]	1472 bits	800 bits	2272 bits
Rostampour [18]	992 bits	800 bits	1772 bits

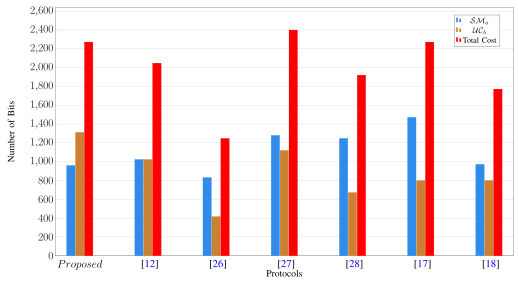

FIGURE 5. Comparison on Communication Costs.

+ 1312) = 2272 bits. The communication overhead of comparable protocols such as [12], [17], [18], [26], [27], [28] is computed using a similar methodology and detailed in Table 7. The graphical representation in Fig. 5 illustrates the communication costs of our protocol alongside other protocols [12], [17], [18], [26], [27], [28].

D. ENERGY CONSUMPTION

Throughout the implementation phase, the system utilizes a certain amount of battery power to process and transmit data among its participants. This energy consumption is quantified as $EC = CE \times CP$ within the wireless communication channel. Here, CE denotes the necessary execution time, and CP represents the maximum CPU power utilized, fixed at 10.88 W for wireless data transmission. It's worth noting that the energy consumed is directly proportional to the execution time. Thus, if the data transmission protocol requires less computational time, it will consequently consume less battery power. As indicated in Table 6, the proposed protocol necessitates only 8.7313 ms for execution, resulting in an energy consumption of 94.9965 mJ. Table 6 provides a comparison of energy consumption among the proposed and related protocols.

TABLE 8. Comparison on security features.

Protocols	*1A*	*2A*	*3A*	*4A*	*5A*	*6A*
Proposed	✓	✓	✓	✓	✓	✓
Irshad et al. [12]	✓	✓	$\mathcal{N}/A$	✓	✓	✗
Safkhani et al. [26]	✓	✓	✓	✓	✓	$\mathcal{N}/A$
Park et al. [27]	✓	✓	✓	✓	✓	$\mathcal{N}/A$
Tanveer et al. [28]	✓	✓	✓	$\mathcal{N}/A$	✓	✗
Kumari and Singh [17]	✓	✓	✓	✓	✓	✗
Rostampour et al. [18]	✓	✓	✓	✓		✗

✓: Provides; ✗: Not Provides; $\mathcal{N}/A$: Not Applicable

E. SECURITY FEATURES

Table 8 depicts a comparative analysis of the security attributes between our and relevant protocols [12], [17], [18], [26], [27], [28]. Our protocol ensures all the important security attributes. On the other hand, the relevant protocols of [12], [17], [18], [26], [27], [28] exhibit vulnerabilities against multiple security threats, including MITM attacks, impersonation attacks, PI attacks, ephemeral secret leakage (ESL) attack, and physical attacks.

Upon reviewing Section V, it becomes evident that the proposed protocol outperforms the related protocols of [12], [17], [18], [26], [27], [28] regarding computational overheads and security attributes. Despite the higher communication cost of our protocol compared to some related ones, it presents a trade-off between lightweightness and the security aspects of an authentication protocol.

VI. CONCLUSION

Our paper unveils an anonymous and lightweight authentication protocol devised to secure communication in the SPG system. Our protocol ingeniously resists both cyber and physical threats by leveraging PUF. The proposed protocol establishes a session key after completing the secure authentication of legitimate entities. We employ the ROR model to formally analyze our protocol, substantiating its robustness regarding privacy and security. Additionally, the informal analysis provides compelling evidence of the protocol's efficacy in thwarting potential security attacks. Our protocol offers superior functionality and exhibits advanced security features that are indispensable to the SPG system. Moreover, the performance comparison demonstrates that the proposed protocol corroborates efficiency, surpassing competing protocols in terms of communication, computation cost, and security features. Despite these advantageous features, it is crucial to acknowledge the proposed protocol's

limitations, particularly its insufficient security against post-quantum attacks. We are committed to addressing this issue and ensuring long-term security. In future, we plan to focus on developing a robust lattice-based authentication protocol capable of withstanding post-quantum attacks, thereby enhancing the security of the SPG system.

REFERENCES

[1] S. Garg, K. Kaur, G. Kaddoum, J. J. Rodrigues, and M. Guizani, "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3548–3557, May 2019.

[2] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi, and M. Shafiq, "A provably secure and efficient authenticated key agreement scheme for energy Internet-based Vehicle-to-grid technology framework," *IEEE Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4425–4435, Jul./Aug. 2020.

[3] M. A. Rahman, M. H. Manshaei, E. Al-Shaer, and M. Shehab, "Secure and private data aggregation for energy consumption scheduling in smart grids," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 2, pp. 221–234, Mar./Apr. 2015.

[4] M. Badra and S. Zeadally, "Design and performance analysis of a virtual ring architecture for smart grid privacy," *IEEE Trans. Inf. forensics security*, vol. 9, no. 2, pp. 321–329, Feb. 2014.

[5] A. A. Khan, V. Kumar, M. Ahmad, and S. Rana, "LAKAF: Lightweight authentication and key agreement framework for smart grid network," *J. Syst. Archit.*, vol. 116, Jun. 2021, Art. no. 102053.

[6] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Gener. Comput. Syst.*, vol. 81, pp. 557–565, Apr. 2018.

[7] X. Li, F. Wu, S. Kumari, L. Xu, A. K. Sangaiah, and K.-K. R. Choo, "A provably secure and anonymous message authentication scheme for smart grids," *J. Parallel Distrib. Comput.*, vol. 132, pp. 242–249, Oct. 2019.

[8] N. Kumar, G. S. Aujla, A. K. Das, and M. Conti, "ECCAuth: A secure authentication protocol for demand response management in a smart grid system," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6572–6582, Dec. 2019.

[9] S. Yu et al., "Privacy-preserving lightweight authentication protocol for demand response management in smart grid environment," *Appl. Sci.*, vol. 10, no. 5, p. 1758, 2020.

[10] A. A. Khan, V. Kumar, M. Ahmad, S. Rana, and D. Mishra, "PALK: Password-based anonymous lightweight key agreement framework for smart grid," *Int. J. Elect. Power Energy Syst.*, vol. 121, Oct. 2020, Art. no. 106121.

[11] J. Srinivas, A. K. Das, X. Li, M. K. Khan, and M. Jo, "Designing anonymous signature-based authenticated key exchange scheme for Internet of Things-enabled smart grid systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4425–4436, Jul. 2020.

[12] A. Irshad, S. A. Chaudhry, M. Alazab, A. Kanwal, M. S. Zia, and Y. B. Zikria, "A secure demand response management authentication scheme for smart grid," *Sustain. Energy Technol. Assess.*, vol. 48, Dec. 2021, Art. no. 101571.

[13] S. A. M. Taqi and S. Jalili, "LSPA-SGs: A lightweight and secure protocol for authentication and key agreement based elliptic curve cryptography in smart grids," *Energy Rep.*, vol. 8, pp. 153–164, Nov. 2022.

[14] H. M. S. Badar, K. Mahmood, W. Akram, Z. Ghaffar, M. Umar, and A. K. Das, "Secure authentication protocol for home area network in smart grid-based smart cities," *Comput. Elect. Eng.*, vol. 108, May 2023, Art. no. 108721.

[15] C.-D. Lee, J.-H. Li, and T.-H. Chen, "A blockchain-enabled authentication and conserved data aggregation scheme for secure smart grids," *IEEE Access*, vol. 11, pp. 85202–85213, 2023.

[16] A. Tomar, S. Tripathi, and K. Arivarasan, "A blockchain-based certificateless aggregate signature scheme for fog-enabled smart grid environment," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 4, pp. 1892–1905, Dec. 2023.

[17] D. Kumari and K. Singh, "Lightweight secure authentication and key agreement technique for smart grid," *Peer-to-Peer Netw. Appl.*, vol. 17, no. 1, pp. 451–478, 2024.

[18] S. Rostampour et al., "Using a privacy-enhanced authentication process to secure IoT-based smart grid infrastructures," *J. Supercomput.*, vol. 80, no. 2, pp. 1668–1693, 2024.

[19] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. Shen, "Towards a light-weight message authentication mechanism tailored for smart grid communications," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2011, pp. 1018–1023.

[20] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[21] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[22] T. Li, C. Gao, L. Jiang, W. Pedrycz, and J. Shen, "Publicly verifiable privacy-preserving aggregation and its application in IoT," *J. Netw. Comput. Appl.*, vol. 126, pp. 39–44, Jan. 2019.

[23] C.-P. Schnorr, "Efficient signature generation by smart cards," *J. Cryptol.*, vol. 4, no. 3, pp. 161–174, 1991.

[24] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2002, pp. 337–351.

[25] C.-C. Chang and H.-D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Trans. wireless Commun.*, vol. 15, no. 1, pp. 357–366, Jan. 2016.

[26] M. Safkhani, S. Kumari, M. Shojafar, and S. Kumar, "An authentication and key agreement scheme for smart grid," *Peer-to-Peer Netw. Appl.*, vol. 15, no. 3, pp. 1595–1616, 2022.

[27] K. Park, J. Lee, A. K. Das, and Y. Park, "BPPS: Blockchain-enabled privacy-preserving scheme for demand-response management in smart grid environments," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1719–1729, Mar./Apr. 2022.

[28] M. Tanveer, H. Shah, A. Alkhayyat, S. A. Chaudhry, and M. Ahmad, "ARAP-SG: Anonymous and reliable authentication protocol for smart grids," *IEEE Access*, vol. 9, pp. 143366–143377, 2021.

AMINA ZAHOOR received the M.C.S. and M.S. degrees in computer science from COMSATS University Islamabad (Sahiwal Campus), Pakistan, in 2018 and 2022, respectively. She is currently pursuing the Ph.D. degree. Her research interests include lightweight cryptography and authenticated key agreement protocols. She was awarded the Campus and the Institute position holder certificate for her academic excellence.

KHALID MAHMOOD (Senior Member, IEEE) received the Ph.D. degree in computer science from International Islamic University, Islamabad, Pakistan, in 2018. He is currently working as an Assistant Professor with the Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan. Earlier, he also served as a Faculty Member with COMSATS University Islamabad, which is ranked one in the IT category. He is an Approved Supervisor with Higher Education Commission, Pakistan. He is the Founder of Network Security Research Group. His research interests include the design and development of lightweight authenticated and key agreement solutions for diverse infrastructures like smart grid, the Internet of Drones, the Internet of Things, vehicular ad hoc networks, mobile edge computing, and blockchain. In 2017, considering his research, the Pakistan Council for Science and Technology granted him the Prestigious Young Productive Scientist Award while affirming he was among the Top Productive Computer Scientists in Pakistan. He is a member of ACM Professional

MUHAMMAD ASAD SALEEM received the M.C.S. and M.S. degrees (Hons.) in computer science from COMSATS University Islamabad (Sahiwal Campus), Pakistan, in 2018 and 2020, respectively, and the Ph.D. degree in computer science and technology from the University of Electronic Science and Technology of China (UESTC). He has authored 27 articles in reputed international journals and conferences, significantly impacting computer science. His research has been published in high-level transactions and journals, including the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and IEEE INTERNET OF THINGS JOURNAL. His research interests include the design of authentication protocols, information security, network security, and applied cryptography. He received the Academic Achievement Award from the School of Computer and Engineering, UESTC from 2022 to 2023. His academic excellence has been recognized with both Campus and Institute Gold Medals.

HAFIZ MUHAMMAD SANAULLAH BADAR received the Ph.D. degree in computer science. He is an Assistant Professor with Emerson University Multan. He has contributed significantly to top-tier conferences and journals, showcasing proficiency in the field. Actively collaborating in research groups, he contributes to advancements in his areas of specialization. With over 8 years of teaching and research experience, his expertise lies in cyber security, security issues in smart grid, artificial intelligence, and machine learning. Furthermore, he has reviewed multiple research works and served as a Technical Session Chair and Organizing Chair for the conference.

TUAN-VINH LE (Member, IEEE) received the Ph.D. degree from the Graduate Institute of Management, Chang Gung University, Taiwan, in July 2021. From August 2021 to July 2022, he was an Assistant Professor with the Department of Information Management, Chihlee University of Technology, Taiwan. He is currently a Tenure-Track Assistant Professor with the Bachelor's Program in Artificial Intelligence and Information Security, College of Science and Engineering, Fu Jen Catholic University, Taiwan, where the Chief of internationalization of the college. He has published his papers in multiple journals, including the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *Mathematics*, *Bioengineering*, IEEE ACCESS, *Sensors*, and *Journal of Internet Technology*. His current research interests include information security, communication system security, cryptography, blockchain, mobile sequencing applications, smart healthcare, and smart grid. He is a Frequent Reviewer and an Editor of multiple SCI-indexed journals. In addition, he is a Committee Member of some international conferences and an instructor certified by the EC-Council.

ASHOK KUMAR DAS (Senior Member, IEEE) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering from IIT Kharagpur, India. He is currently a Full Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. He was also a Visiting Faculty Member of the Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA, USA. He has authored over 420 papers in international journals and conferences in the above areas, including over 355 reputed journal papers. His Google Scholar H-index is 87, and his i10-index is 264, with over 22 000 citations. His research interests include cryptography, system and network security, blockchain, security in the Internet of Things, Internet of Vehicles, Internet of Drones, smart grids, smart city, cloud/fog computing, intrusion detection, AI/ML security, and post-quantum cryptography. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He has been listed in the Web of Science (Clarivate^{TM}) Highly Cited Researcher in recognition of his exceptional research performance in 2022 and 2023. He is/was on the editorial board of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE SYSTEMS JOURNAL, *Journal of Network and Computer Applications* (Elsevier), *Computer Communications* (Elsevier), *Journal of Cloud Computing* (Springer), *Cyber Security and Applications* (Elsevier), *IET Communications*, *KSII Transactions on Internet and Information Systems*, and *International Journal of Internet Technology and Secured Transactions* (Inderscience). He also served as one of the Technical Program Committee Chairs of the first International Congress on Blockchain and Applications, Avila, Spain, in June 2019, International Conference on Applied Soft Computing and Communication Networks, Chennai, India, in October 2020, and second International Congress on Blockchain and Applications, L'Aquila, Italy, in October 2020.