cryptography

MDPI

Article

A Provably Secure Anonymous Authentication Protocol for Consumer and Service Provider Information Transmissions in Smart Grids

Zahraa Abdullah Ali [1], Zaid Ameen Abduljabbar [1,*], Hamid Ali Abed AL-Asadi [1], Vincent Omollo Nyangaresi [2,3], Iman Qays Abduljaleel [1] and Abdulla J. Y. Aldarwish [1,4]

[1] Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq; iman.abduljaleel@uobasrah.edu.iq (I.Q.A.)
[2] Department of Computer Science and Software Engineering, Jaramogi Oginga Odinga University of Science and Technology, Bondo 40601, Kenya
[3] Department of Applied Electronics, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 600124, India
[4] Department of Computer Science, Gujarat University, Ahmedabad 380009, India
* Correspondence: zaid.ameen@uobasrah.edu.iq

Abstract: Smart grids integrate information technology, decision support systems, communication networks, and sensing technologies. All these components cooperate to facilitate dynamic power adjustments based on received client consumption reports. Although this brings forth energy efficiency, the transmission of sensitive data over the public internet exposes these networks to numerous attacks. To this end, numerous security solutions have been presented recently. Most of these techniques deploy conventional cryptographic systems such as public key infrastructure, blockchains, and physically unclonable functions that have either performance or security issues. In this paper, a fairly efficient authentication scheme is developed and analyzed. Its formal security analysis is carried out using the Burrows–Abadi–Needham (BAN) logic, which shows that the session key negotiated is provably secure. We also execute a semantic security analysis of this protocol to demonstrate that it can resist typical smart grid attacks such as privileged insider, guessing, eavesdropping, and ephemeral secret leakages. Moreover, it has the lowest amount of computation costs and relatively lower communication overheads as well as storage costs.

Keywords: attacks; authentication; BAN; protocol; security; smart grids; privacy

Citation: Ali, Z.A.; Abduljabbar, Z.A.; AL-Asadi, H.A.A.; Nyangaresi, V.O.; Abduljaleel, I.Q.; Aldarwish, A.J.Y. A Provably Secure Anonymous Authentication Protocol for Consumer and Service Provider Information Transmissions in Smart Grids. *Cryptography* **2024**, *8*, 20. https://doi.org/10.3390/cryptography8020020

Academic Editor: Josef Pieprzyk

Received: 29 March 2024
Revised: 4 May 2024
Accepted: 7 May 2024
Published: 9 May 2024

1. Introduction

Smart grid (SG) networks incorporate information technology and energy grid so as to manage energy consumptions efficiently. This is normally accomplished by offering bi-directional communication for data exchanges between consumers and power producers [1]. In addition, an SG integrates intelligent sensing, contemporary communication networks, and novel systems that support decision making in conventional grid systems. These technologies enable the effectual distribution of power from the generating stations to the consumer terminals. As explained in [2], SG bi-directional communication is achieved through Advanced Metering Infrastructure (AMI). A typical AMI comprises concentrators, smart meters, and measurement data management systems. On the other hand, a typical SG is made up of control, sensing, and communication systems and actuators [3]. Whereas smart meters (SMs) perform sensing and communication, actuation and control are executed by service providers (SPs). Therefore, SMs are located at consumer premises, where they accurately measure power consumption and transmit these data over to the SP servers. Through effective real-time processing and analyses of consumer data, the generation and distribution of power is dynamically fine-tuned in accordance with user demands. This helps in enhancing the reliability of the power grid system [4].

In spite of the benefits discussed above, the public internet is utilized for the data exchange between the SMs and the SPs [5]. As such, the SG is exposed to security and privacy threats such as eavesdropping, forgery, denial of service (DoS), tampering, and ephemeral secret leakage (EPSL) [6,7]. In addition, the misuse of consumer power consumption reports can lead to privacy leaks. By sending forged and inaccurate data, the SG network can incur additional loads [8]. All these challenges can disrupt the communication process, leading to the degradation of the SG system's performance [9]. As such, security violations and privacy leakages are major issues during smart grid design [10]. This can be attained by perfect data encryption, mutual authentication, as well as session key establishment. In addition, Authenticated Key Exchange (AKE) is crucial for the protection of transmitted data against tampering and interception [6].

The above concerns necessitate the designing of robust, privacy-preserving, secure, and lightweight protocols to safeguard the data exchanged among legitimate SG participants. Since an SG comprises numerous SMs, each SM must be authenticated prior to information exchange. This will help curb threats exampled by impersonation, SM capture, Man-in-the-Middle (MitM), packet replays, de-synchronization, and privileged insider [7]. Upon an effectual mutual authentication process, a common session key should be created between the SM and the SPs to encipher the exchanged data. In addition, data integrity should be upheld, while preventing non-repudiation and side-channeling through a power analysis [11]. Another major concern in an SG network is the limited capabilities of smart meters in terms of communication, energy, and computation. This puts some limitations on the implementation of conventional cryptographic techniques in SG networks. Therefore, ideal SG security approaches should strive to be lightweight in addition to fulfilling numerous security requirements.

1.1. Motivation

It has been shown that a myriad of protocols have been introduced in the smart grid network to preserve its security posture. However, these solutions are based on conventional cryptographic systems such blockchain, public key infrastructure, PUF, and bilinear pairings. All these techniques have many security, performance, or privacy issues and, hence, are not suitable for resource-incapacitated SG devices such as SMs. Attacks such as de-synchronization, impersonation, privacy leaks, replays, and DoS must be prevented, as they adversely interfere with the reliability of smart grids. As such, there is a need for an effective, efficient, and robust security scheme for SGs.

1.2. Threat Model

In this section, we model attacks against our scheme using the most popular Dolev–Yao (DY) and Canetti–Krawczyk models. In these threat models, attacker Ä is capable of the following actions, compromising the private keys belonging to smart meters and service providers:

- Modifying and deleting the contents of intercepted messages;
- Generating and forwarding bogus messages to unsuspecting entities;
- Physically capturing and compromising network entities such as smart meters;
- Retrieving sensitive security tokens stored in the smart meter's memory;
- Deploying extracted smart meter memory content to execute attacks;
- Intercepting derived session keys and other session state parameters.

1.3. Security Requirements

In the face of numerous security threats and privacy leaks, an ideal authentication scheme for smart grid networks should fulfill the following requirements:

Mutual authentication: The identities of all the communicating parties should be reciprocally verified prior to exchanging any network data.

Key agreement: To preserve confidentiality and the integrity of the communication process, a session key should be set up to encrypt all exchanged messages.

Anonymity and untraceability: An attacker should be incapable of discerning the real identity of the communicating entities based on any captured network messages. Additionally, the attacker should be incapable of tracing the communicating parties using these intercepted messages.

Key security: The captured current session key should not facilitate the derivation of past and subsequent session keys.

Formal verification: The derived session key should be mathematically sound.

Resilience against: To offer sufficient security, an ideal authentication protocol needs to withstand attacks such as EPSL, de-synchronization, DoS, eavesdropping, privileged insider, guessing, spoofing, Known Session-Secret Temporary Information (KSSTI), ephemeral secret leakage, physical capture, impersonation, replay, MitM, and forgery.

1.4. Contributions

To address the security, performance, and privacy challenges discussed above, we make the following contributions in our paper.

- We deploy shared keys and pseudo-identities to encipher the communication channel so as to enhance security and privacy preservation.
- To protect against MitM and replay attacks, each entity computes the session keys for traffic protection.
- We deploy BAN logic for the revelation of the probably secure nature of the negotiated session key.

An extensive comparative analysis shows that our protocol withstands the largest number of attacks. In addition, it incurs the lowest computation overheads and relatively lower storage and communication overheads.

The rest of this work is structured as follows: Section 2 discusses the related works in this domain, while our scheme is described in Section 3. On the other hand, Section 4 discusses the security analysis of this protocol, while Section 5 describes its evaluation in terms of performance. Finally, Section 6 presents the conclusions and gives some future research scopes.

2. Related Work

Smart grid security, privacy, and performance have attracted a lot of attention, leading to the introduction of many schemes. For instance, researchers in [10] have presented an identity-based technique, while the authors in [12,13] have developed elliptic curve cryptography (ECC)-based schemes. However, extensive ECC multiplication operations render the schemes in [12,13] inefficient [14]. Therefore, they are not ideal for deployment in computation-limited smart grid components. On the other hand, PUF-based schemes are developed in [15–18]. Although the protocol in [15] withstands modeling attacks, protocols based on PUF have stability issues [19]. In addition, the scheme in [18] offers smart meter physical security but is still vulnerable to EPSL attacks and cannot provide backward key secrecy [17]. To offer smart meter anonymity, a secure scheme is presented in [20]. However, this scheme fails to mutually authenticate the network entities and is prone to DoS attacks [21]. Although the scheme in [22] is anonymity-preserving, it cannot withstand ephemeral secret and session key leakage attacks [23]. In addition, its bilinear pairing operations result in extensive computation overheads [24], similar to the protocols in [23,25].

To reduce the computation overheads associated with bilinear pairings, a scheme based on elliptic curve cryptography is developed in [26]. However, this technique cannot offer anonymity [1] and is defenseless against ephemeral secret leakage attacks [27]. Additionally, it incurs high computation overheads during the generation of security tokens at the Trusted Authority (TA) [1]. On the same breadth, the technique introduced in [28] fails to offer untraceability and identity protection [29]. To deal with these challenges, an anonymous authentication protocol is introduced in [30]. Although identity protection is assured, this technique incurs high computation costs [6]. To offer efficiency in smart grids, lightweight

authentication schemes are developed in [1,6,29,31–34]. However, the schemes in [6,31,32] have not been evaluated against de-synchronization attacks. Similarly, the protocol in [29] has not been evaluated against spoofing and guessing attacks. Although the schemes in [1,33] are resilient against de-synchronization attacks, they have not been evaluated against spoofing attacks. On the other hand, the scheme in [34] cannot withstand de-synchronization attacks [29].

To address the anonymity issues in some of the protocols above, a password-based security technique is introduced in [35]. However, this protocol has incorrect login and authentication phases [36]. Although the scheme in [37,38] overcomes this challenge, it is defenseless against de-synchronization threats. In addition, it fails to provide formal security verification and revocability. On the other hand, the usage of some fixed messages in each session in [39,40] renders said session vulnerable to traceability attacks. The protocol in [41] solves this issue by updating this message for each session. However, the service provider needs to buffer previous data for each SM so as to withstand de-synchronization attacks. Consequently, it incurs heavy storage costs especially in networks with massive SMs.

To enhance security in wireless networks, quantum computing technology has been adopted. For instance, based on quantum information engineering, a technique for local energy distribution to numerous remote nodes is presented in [42], while a verification scheme applicable in a quantum channel is developed in [43]. On the other hand, a blind quantum-based protocol is presented in [44], while a zero-knowledge proof is developed in [45]. However, comparative performance analyses have not been carried out in [42–45]. As explained in [46], blockchain technology can ensure privacy and security devoid of an authorized third party. As such, a blockchain-based protocol is presented in [47]. Although blockchain technology provides traceability, improved security, and immutability, it raises serious issues regarding transparency and privacy [48]. In addition, the blockchain-based protocol in [47] lacks evaluation against threats such as privileged insider and physical capture. To avert the misuse and malicious manipulation of battery equipment and data, a robust security scheme is presented in [49]. Although this technique protects against counterfeiting and possible software backdoors, its comparative security and performance evaluations are missing.

Based on the above discussions, it is clear that many schemes have been developed to address security and privacy issues in the smart grid environment. However, most of them still have challenges in terms of privacy, performance, or security. There is, therefore, a need for the development of novel protocols that can help alleviate these challenges.

3. The Proposed Protocol

The network model of our protocol comprises a utility service provider (USP), a trusted control server (TC), and a smart meter (SM), as evidenced in Figure 1. The TCS executes system initialization and generates the secret values for the SM and the USP during the registration phase.

The SM measures electricity usage on the client end and transmits power consumption reports to the USP over public channels. At the USP, these reports are processed and analyzed to facilitate decision making, which may include dynamic power adjustments. Table 1 describes the symbols used throughout this paper.

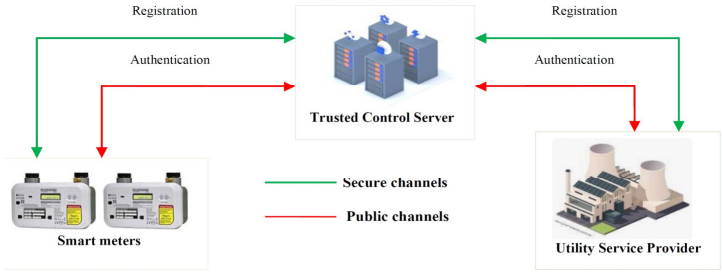

Figure 1. Proposed network model.

Table 1. Notations.

Symbol	Descriptions
TCS	Trusted control server
SM_i	ith smart meter
USP	Utility service provider
K_{TCS}	Master key of the TCS
ID_{TCS}	Unique identifier of the TCS
ID_{SM}	Unique identifier of the SM_i
K_{SM}	SM_i's private key
R_i	Random nonce i
PID_{SM}	SM's pseudo-identity
K_{TSM}	Shared key between TCS and SM
ID_{USP}	Unique identity of the USP
K_{USP}	USP's private key
PID_{USP}	USP's pseudo-identity
K_{UT}	Shared key between USP and TCS
SK_{SU}	Session key between SM_i and USP
h (.)	Hashing function
| |	Concatenation operation
$\oplus$	XOR operation

Our scheme executes five major steps, which encompass system setup, entity registration, mutual authentication, key negotiation, and parameter refresh phases. Algorithm 1 summarizes this protocol, and the sub-sections that follow give the details of these phases.

Algorithm 1 Secure and efficient authentication

Begin

#**********$System\ setup\ phase$ **********#

(1) Generate K_{TCS}, ID_{TCS}, ID_{SM} & K_{SM}

#**********$Registration\ phase$ **********#

(2) Generate R_1 & derive $PID_{SM,}$ then $\overset{Reg-1}{\rightarrow}$ TCS

(3) Generate R_2 & compute K_{TSM}

(4) Store{PID_{SM}, K_{TSM}, R_1}, publish PID_{SM}, then $\overset{Reg-2}{\rightarrow}$ SM_i

(5) Calculate A_1, A_2 & store {A_1, A_2, PID_{SM}}

(6) Generate R_3, select ID_{USP} & K_{USP}, then $\overset{Reg-3}{\rightarrow}$ TCS

(7) Compute K_{UT} & A_3

(8) Store {PID_{USP}, A_3, K_{UT}}, then $\overset{Reg-4}{\rightarrow}$ USP

(9) Calculate A_4, A_5, B_1, B_2 & B_3

(10) Store {A_5, B_1, B_2, B_3}

Algorithm 1 *Cont.*

#********** *Authentication and key negotiation phase* **********#

(11) Input $\{ID_{USP}, K_{USP}\}$, then compute R_3, A_4 & ${B_1}^*$

(12) **If** ${B_1}^*$!= B_1 **then:**

(13) Terminate session

(14) **Else:**

(15) Generate R_4, derive A_3, K_{UT}, B_4, B_5 & C_1, then $\overset{Auth-1}{\rightarrow}$ TCS

(16) Retrive A_3, K_{UT} & derive $({R_4}^* || {PID_{SM}}^*)$, ${C_1}^*$

(17) **If** ${C_1}^*$!= C_1 **then:**

(18) Abort session

(19) **Else:**

(20) Generate R_5 & Fetch K_{TSM}, R_1

(21) Derive C_2, C_3, C_4 & C_5, then $\overset{Auth-2}{\rightarrow}$ SM_i

(22) Calculate R_1, K_{TSM}, ${C_2}^*$ & C_5

(23) **If** ${C_5}^*$!= C_5 **then:**

(24) Stop session

(25) **Else:**

(26) Generate R_6, derive $(h(ID_{USP} || R_4) || h(ID_{TCS} || R_5))$, SK_{SU}, D_1 & D_2, then $\overset{Auth-3}{\rightarrow}$ TCS

(27) Derive $h(ID_{SM} || R_6)$ & ${D_2}^*$

(28) **If** ${D_2}^*$!= D_2 **then:**

(29) Abort session

(30) **Else:**

(31) Derive SK_{SU}, ${PID_{USP}}^*$, ${A_3}^*$, D_3 & D_4

(32) Store $\{PID_{USP}, A_3\}$ with $\{{PID_{USP}}^*, {A_3}^*\}$, then $\overset{Auth-4}{\rightarrow}$ USP

(33) Calculate ${PID_{USP}}^*$, $(h(ID_{TCS} || R_5) || h(ID_{SM} || R_6) || {PID_{USP}}^*)$ & ${D_4}^*$

(34) **If** ${D_4}^*$!= D_4 **then:**

(35) Stop session

(36) **Else:**

(37) Compute SK_{SU}, ${A_3}^*$, ${B_2}^*$ & ${B_3}^*$

(38) Substitute $\{B_2, B_3, PID_{USP}\}$ with $\{{B_2}^*, {B_3}^*, {PID_{USP}}^*\}$

(39) Derive D_5

(40) **If** ${D_5}^*$!= D_5 **then:**

(41) Terminate session

(42) Delete$\{PID_{USP}, A_3\}$ from database

(43) **Endif; Endif;**

(44) **Endif; Endif;**

(45) **Endif;**

End

3.1. System Setup

In this phase, the TCS selects its master key as K_{TCS}. This is followed by the generation of its unique identity ID_{TCS}, the smart meter's unique identity ID_{SM}, as well as the private key of the smart meter, K_{SM}, as shown in Figure 2.

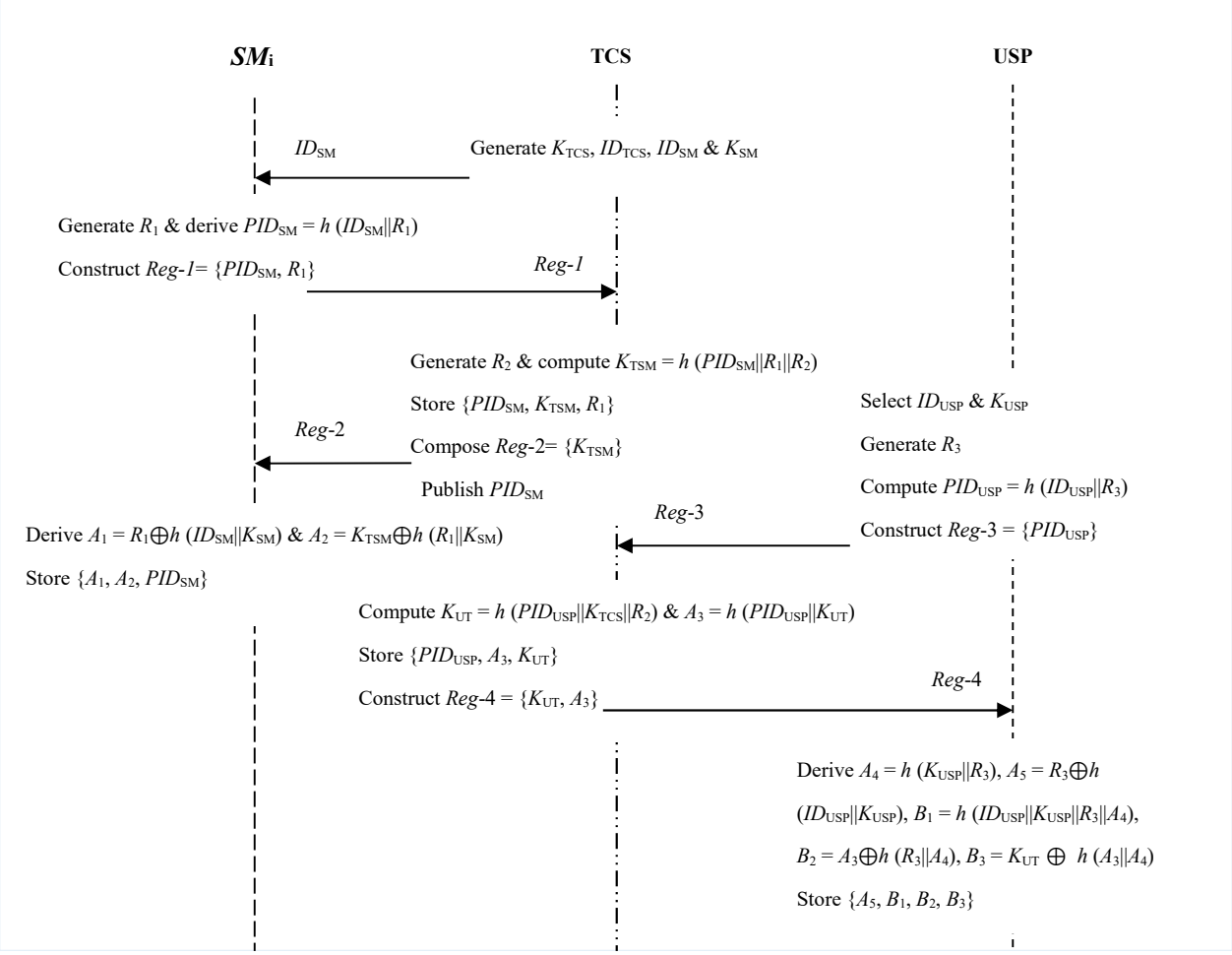

Figure 2. System initialization and registration.

3.2. Registration

In this particular phase, the smart meters are registered at the TCS before they are deployed in the actual field. In addition, the USP is also registered at the TCS prior to exchanging data with the smart meters. The following sub-sections describe this phase in more detail.

3.2.1. Smart Meter Registration

The subsequent three procedures are executed to register the smart meter SM_i to the TCS. To accomplish this, secure communication channels are deployed.

Step 1: The SM_i chooses a random nonce R_1 to derive its pseudo-identity $PID_{SM} = h(ID_{SM} || R_1)$. It then composes registration message $Reg\text{-}1 = \{PID_{SM}, R_1\}$ that is forwarded to the TCS over secure communication media, as shown in Figure 2.

Step 2: When it receives message *Reg*-1, the TCS selects a random nonce R_2 that is deployed to compute the shared key $K_{TSM} = h(PID_{SM} || R_1 || R_2)$. Next, the TCS stores $\{PID_{SM}, K_{TSM}, R_1\}$ in its repository. Next, registration message $Reg\text{-}2 = \{K_{TSM}\}$ is constructed and forwarded to the SM_i, as evidenced in Figure 2. Afterwards, the TCS publishes PID_{SM}.

Step 3: Upon receiving the message *Reg*-2, the smart meter SM_i derives $A_1 = R_1 \oplus h\ (ID_{SM} \| K_{SM})$ and $A_2 = K_{TSM} \oplus h\ (R_1 \| K_{SM})$. Thereafter, it stores $\{A_1, A_2, PID_{SM}\}$ in its memory.

3.2.2. Utility Service Provider Registration

To register to the TCS, the USP needs to execute the following three procedures.

Step 1: The USP chooses its real identity ID_{USP} and secret key K_{USP}. Next, it generates a random nonce R_3 that is used to calculate its pseudo-identity $PID_{USP} = h\ (ID_{USP} \| R_3)$. Thereafter, it constructs registration message *Reg*-3 = $\{PID_{USP}\}$, which is transmitted to the TCS, as depicted in Figure 2.

Step 2: After receiving registration message *Reg*-3, the TCS calculates shared key $K_{UT} = h\ (PID_{USP} \| K_{TCS} \| R_2)$ and $A_3 = h\ (PID_{USP} \| K_{UT})$. Next, it stores $\{PID_{USP}, A_3, K_{UT}\}$ in its database. Finally, registration message *Reg*-4 = $\{K_{UT}, A_3\}$ is composed and sent to the USP.

Step 3: Upon receiving message *Reg*-4, the USP derives $A_4 = h\ (K_{USP} \| R_3)$, $A_5 = R_3 \oplus h\ (ID_{USP} \| K_{USP})$, $B_1 = h\ (ID_{USP} \| K_{USP} \| R_3 \| A_4)$, $B_2 = A_3 \oplus h\ (R_3 \| A_4)$, and $B_3 = K_{UT} \oplus h\ (A_3 \| A_4)$. Next, it stores $\{A_5, B_1, B_2, B_3\}$ in its database.

3.3. Authentication and Key Setup

To securely exchange power consumption reports and adjustment commands, the USP and SM_i must first mutually validate one another. This is followed by the establishment of a session key for message protection over the public internet. The subsequent nine steps are utilized to accomplish these two processes.

Step 1: The USP operator supplies parameter set $\{ID_{USP}, K_{USP}\}$, after which values $R_3 = A_5 \oplus h\ (ID_{USP} \| K_{USP})$, $A_4 = h\ (K_{USP} \| R_3)$, and $B_1^* = h\ (ID_{USP} \| K_{USP} \| R_3 \| A_4)$ are computed. Next, it confirms if $B_1^* \stackrel{?}{=} B_1$ in a manner such that the communication session is aborted if these two parameters are not identical. Otherwise, the USP randomly generates nonce R_4, which is used to derive $A_3 = B_2 \oplus h\ (R_3 \| A_4)$, $K_{UT} = B_3 \oplus h\ (A_3 \| A_4)$, $B_4 = h\ (PID_{USP} \| A_3 \| K_{UT}) \oplus (R_4 \| PID_{SM})$, $B_5 = h\ (ID_{USP} \| R_4) \oplus h\ (K_{UT} \| R_4)$, and $C_1 = h\ (PID_{USP} \| A_3 \| R_4 \| PID_{SM} \| K_{UT})$. At the end, message *Auth-1* = $\{PID_{USP}, B_4, B_5, C_1\}$ is constructed and transmitted to the TCS, as shown in Figure 3.

Step 2: After receiving message *Auth-1*, TCS retrieves A_3 and K_{UT} corresponding to PID_{USP} and derives $(R_4^* \| PID_{SM}^*) = B_4 \oplus h\ (PID_{USP} \| A_3 \| K_{UT})$ as well as $C_1^* = h\ (PID_{USP} \| A_3 \| R_4^* \| PID_{SM}^* \| K_{UT})$. Thereafter, the TCS validates if $C_1^* \stackrel{?}{=} C_1$ such that the communication session is halted when this check flops. If not, the TCS fetches K_{TSM} and R_1 corresponding to PID_{SM}.

Step 3: The TCS randomly generates number R_5, which is used to calculate $C_2 = h\ (R_4 \| R_5)$, $C_3 = h\ (PID_{SM} \| K_{TSM} \| R_1) \oplus C_2$, $h\ (ID_{USP} \| R_4) = B_5 \oplus h\ (K_{UT} \| R_4)$, $C_4 = (h\ (ID_{USP} \| R_4) \| h\ (ID_{TCS} \| R_5)) \oplus h\ (K_{TSM} \| R_1)$, and $C_5 = h\ (PID_{USP} \| C_2 \| K_{TSM})$. Finally, message *Auth-2* = $\{PID_{USP}, C_3, C_4, C_5\}$ is composed and passed over to the SM_i.

Step 4: After receiving *Auth-2*, SM_i computes $R_1 = A_1 \oplus h\ (ID_{SM} \| K_{SM})$, $K_{TSM} = A_2 \oplus h\ (R_1 \| K_{SM})$, $C_2^* = h\ (PID_{SM} \| K_{TSM} \| R_1) \oplus C_3$, and $C_5 = h\ (PID_{USP} \| C_2^* \| K_{TSM})$. Next, it confirms whether $C_5^* \stackrel{?}{=} C_5$ such that the communication session is abandoned upon validation flop. Otherwise, it chooses a random nonce R_6 and calculates $(h\ (ID_{USP} \| R_4) \| h\ (ID_{TCS} \| R_5)) = C_4 \oplus h\ (K_{TSM} \| R_1)$.

Step 5: The SM_i derives session key $SK_{SU} = h\ (h\ (ID_{USP} \| R_4) \| h\ (ID_{TCS} \| R_5) \| h\ (ID_{SM} \| R_6)$, $D_1 = h\ (PID_{SM} \| K_{TSM} \| R_1) \oplus h\ (ID_{SM} \| R_6)$, and $D_2 = h\ (PID_{USP} \| PID_{SM} \| C_2^* \| h\ (ID_{SM} \| R_6) \| K_{TSM})$. Next, message *Auth-3* = $\{D_1, D_2\}$ is constructed and forwarded to the TCS.

Figure 3. Authentication and key negotiation.

Step 6: Upon receiving message *Auth-3*, the TCS calculates $h\ (ID_{SM}||R_6) = D_1 \oplus h\ (PID_{SM}||K_{TSM}||R_1)$ and $D_2{*} = h\ (PID_{USP}||PID_{SM}||C_2||h\ (ID_{SM}||R_6)||K_{TSM})$. Next, it checks if $D_2{*} \stackrel{?}{=} D_2$ so that the authentication process is terminated upon verification

failure. Otherwise, it computes session key $SK_{SU} = h\ (h\ (ID_{USP} \mid\mid R_4) \mid\mid h\ (ID_{TCS} \mid\mid R_5) \mid\mid h\ (ID_{SM} \mid\mid R_6)$, new pseudo-identity $PID_{USP}{}^* = h\ (PID_{USP} \mid\mid R_4)$, $A_3{}^* = h\ (PID_{USP}{}^* \mid\mid K_{UT})$, $D_3 = h\ (A_3 \mid\mid R_4) \oplus (h\ (ID_{TCS} \mid\mid R_5) \mid\mid h\ (ID_{SM} \mid\mid R_6) \mid\mid PID_{USP}{}^*)$, and $D_4 = h\ (PID_{USP} \mid\mid R_4 \mid\mid h\ (ID_{TCS} \mid\mid R_5) \mid\mid h\ (ID_{SM} \mid\mid R_6) \mid\mid PID_{USP}{}^* \mid\mid K_{UT})$. The TCS stores $\{PID_{USP}, A_3\}$ with $\{PID_{USP}{}^*, A_3{}^*\}$ in its database. At the end, authentication message *Auth-4* = $\{D_3, D_4\}$ is composed and sent over to the USP.

Step 7: Upon receiving *Auth-4*, the USP derives $PID_{USP}{}^* = h\ (PID_{USP} \mid\mid R_4)$, $(h\ (ID_{TCS} \mid\mid R_5) \mid\mid h\ (ID_{SM} \mid\mid R_6) \mid\mid PID_{USP}{}^*) = D_3 \oplus h\ (A_3 \mid\mid R_4)$, and $D_4{}^* = h\ (PID_{USP} \mid\mid R_4 \mid\mid h\ (ID_{TCS} \mid\mid R_5) \mid\mid h\ (ID_{SM} \mid\mid R_6) \mid\mid PID_{USP}{}^* \mid\mid K_{UT})$. It then confirms if $D_4{}^* \stackrel{?}{=} D_4$ such that the authentication is aborted when the verification flops. Otherwise, it derives session key $SK_{SU} = h\ (h\ (ID_{USP} \mid\mid R_4) \mid\mid h\ (ID_{TCS} \mid\mid R_5) \mid\mid h\ (ID_{SM} \mid\mid R_6)$.

Step 8: The USP derives parameters $A_3{}^* = h\ (PID_{USP}{}^* \mid\mid K_{UT})$, $B_2{}^* = A_3{}^* \oplus h\ (R_3 \mid\mid A_4)$, and $B_3{}^* = K_{UT} \oplus h\ (A_3{}^* \mid\mid A_4)$. Next, it replaces $\{B_2, B_3, PID_{USP}\}$ with $\{B_2{}^*, B_3{}^*, PID_{USP}{}^*\}$ in its database. Finally, it derives $D_5 = h\ (SK_{SU} \mid\mid PID_{USP}{}^*)$ and transmits it towards the TCS for the subsequent session.

Step 9: After receiving D_5, the TCS recomputes $D_5{}^* = h\ (SK_{SU} \mid\mid PID_{USP}{}^*)$. Next, it confirms if $D_5{}^* \stackrel{?}{=} D_5$ such that it terminates the session when this validation fails. Otherwise, it deletes parameter set $\{PID_{USP}, A_3\}$ from its database.

3.4. Parameter Update

In this phase, the USP's private key KUSP is updated using the following two steps.

Step 1: The operator supplies their unique identity ID_{USP} as well old secret key $K_{USP}{}^{Old}$. This is followed by the derivation of parameter $R_3 = A_5 \oplus h\ (ID_{USP} \mid\mid K_{USP}{}^{Old})$, $A_4 = h\ (K_{USP}{}^{Old} \mid\mid R_3)$, and $B_1{}^* = h\ (ID_{USP} \mid\mid K_{USP}{}^{Old} \mid\mid R_3 \mid\mid A_4)$. The USP checks if $B_1{}^* \stackrel{?}{=} B_1$ such that this authentication is halted when this check fails. Otherwise, the operator is prompted to input the new secret key $K_{USP}{}^{New}$.

Step 2: The USP derives $A_3 = B_2 \oplus h\ (R_3 \mid\mid A_4)$, $K_{UT} = B_3 \oplus h\ (A_3 \mid\mid A_4)$, $A_4{}^{New} = h\ (K_{USP}{}^{New} \mid\mid R_3)$, $A_5{}^{New} = R_3 \oplus h\ (ID_{USP} \mid\mid K_{USP}{}^{New})$, $B_1{}^{New} = h\ (ID_{USP} \mid\mid K_{USP}{}^{New} \mid\mid R_3 \mid\mid A_4{}^{New})$, $B_2{}^{New} = A_3 \oplus h\ (R_3 \mid\mid A_4{}^{New})$, and $B_3{}^{New} = K_{UT} \oplus h\ (A_3 \mid\mid A_4{}^{New})$. Lastly, it replaces parameter set $\{A_5, B_1, B_2, B_3\}$ with its refreshed equivalents $\{A_5{}^{New}, B_1{}^{New}, B_2{}^{New}, B_3{}^{New}\}$.

4. Security Analysis

In most of the authentication protocols, both formal and informal security analyses are carried out. As such, we present these analyses in this section and provide further details in the sub-sections that follow.

4.1. Formal Security Analysis

To accomplish this analysis, BAN logic is deployed to show that USP and SM_i authenticate each other based on fresh and reliable data. Essentially, this involves the verification of the origin, freshness, and legitimacy of the exchanged messages. The notations in Table 2 are used throughout this formal analysis.

Table 2. BAN logic notations.

Notation	Details
R	Secret key
$A \mid \equiv X$	Entity A believes statement X
$A \mid \sim X$	Entity A once said statement X
$<X>_M$	X is combined with M
$A \lhd X$	Entity A sees statement X
$A \Rightarrow X$	Entity A has jurisdiction over X

Table 2. *Cont.*

Notation	Details
# (*X*)	Message *X* is fresh
$(X)_{\mathrm{R}}$	Message *X* is hashed using key *R*
(*X*, *M*)	X or M is part of formula (X, M)
$A \overset{R}{\leftrightarrow} B$	Entities *A* and *B* share secret key *R*
$\{X\}_{\mathrm{R}}$	Message *X* is enciphered using key *R*
$\mathrm{A} \overset{R}{\rightleftharpoons} \mathrm{B}$	*R* is only known to *A* and *B*

The BAN logic postulates are described using a number of rules that are detailed in Table 3 below.

Table 3. BAN logic rules.

Rule	Details
$\frac{A\mid\equiv \mathrm{A}\overset{R}{\leftrightarrow}\mathrm{B}, \mathrm{A}\lhd\{X\}_R}{A\mid\equiv B\mid\sim X}$	Message Meaning Rule (MMR)
$\frac{A\mid\equiv \#(X), A\mid\equiv B\mid\sim X}{A\mid\equiv B\mid\equiv X}$	Nonce Verification Rule (NVR)
$\frac{A\mid\equiv B\mid\equiv (X,M)}{A\mid\equiv B\mid\equiv X}$	Believe Rule (BR)
$\frac{A\mid\equiv B\Rightarrow X, A\mid\equiv B\mid\equiv X}{A\mid\equiv X}$	Jurisdiction Rule (JR)
$\frac{A\mid\equiv \#(X)}{A\mid\equiv \#(X,M)}$	Freshness rule (FR)

Next, we lay bare that our protocol offers protected mutual validation between the SM_{i} and the USP. In our protocol, four messages are exchanged during the processes of entity verification and session key setup. These particular messages are idealized as follows:

Auth-1. USP $\rightarrow$ TCS: $\{PID_{\mathrm{USP}}, B_4, B_5, C_1\}$
Idealized form: $(PID_{\mathrm{USP}}, A_3, R_4)_{K_{UT}}$
Auth-2. TCS$\rightarrow$ SM_{i}: $\{PID_{\mathrm{USP}}, C_3, C_4, C_5\}$
Idealized form: $(PID_{\mathrm{USP}}, h\,(ID_{\mathrm{USP}} \mid\mid R_4), h\,(ID_{\mathrm{TCS}} \mid\mid R_5), PID_{\mathrm{SM}}, R_1)_{K_{\mathrm{TSM}}}$
Auth-3. SM_{i} $\rightarrow$TCS: $\{D_1, D_2\}$
Idealized form: $(PID_{\mathrm{USP}}, PID_{\mathrm{SM}}, h\,(ID_{\mathrm{USP}} \mid\mid R_4), h\,(ID_{\mathrm{SM}} \mid\mid R_6))_{K_{\mathrm{TSM}}}$
Auth-4. TCS $\rightarrow$ USP: $\{D_3, D_4\}$
Idealized form: $(A_3, h\,(ID_{\mathrm{USP}} \mid\mid R_4), h\,(ID_{\mathrm{TCS}} \mid\mid R_5), h\,(ID_{\mathrm{SM}} \mid\mid R_6))_{K_{\mathrm{UT}}}$

Using the BAN logic analytic procedures, our scheme should uphold the four security goals (GLs) below.

GL$_1$: USP$\mid\equiv$ (USP $\overset{SK_{SU}}{\leftrightarrow}$ SM)
GL$_2$: USP$\mid\equiv$ SM$\mid\equiv$ (USP $\overset{SK_{SU}}{\leftrightarrow}$ SM)
GL$_3$: SM$\mid\equiv$ (USP $\overset{SK_{SU}}{\leftrightarrow}$ SM)
GL$_4$: SM$\mid\equiv$ USP$\mid\equiv$ (USP $\overset{SK_{SU}}{\leftrightarrow}$ SM)

To ensure that the BAN logic analysis of our scheme is successfully executed, a number of initial state assumptions ($\mathrm{AS_i}$) are made as follows.

AS$_1$: TCS$\mid\equiv$ (USP $\overset{SK_{SU}}{\leftrightarrow}$ TCS)
AS$_2$: TCS$\mid\equiv$ # (R_4)
AS$_3$: SM$\mid\equiv$ (TCS $\overset{K_{\mathrm{TSM}}}{\leftrightarrow}$ SM)
AS$_4$: SM$\mid\equiv$ # (R_5)
AS$_5$: TCS$\mid\equiv$ (TCS $\overset{K_{\mathrm{TSM}}}{\leftrightarrow}$ SM)
AS$_6$: TCS$\mid\equiv$ # (R_6)
AS$_7$: USP$\mid\equiv$ (USP $\overset{K_{\mathrm{UT}}}{\leftrightarrow}$ TCS)
AS$_8$: USP$\mid\equiv$ # (R_5)

AS$_9$: USP$|\equiv$TCS$|\Rightarrow$ (USP $\overset{h\,(ID_{\mathrm{TCS}}||R_5)||h(ID_{\mathrm{SM}}||R_6)}{\rightleftharpoons}$ SM)

AS$_{10}$: SM$|\equiv$TCS$|\Rightarrow$ (USP $\overset{h\,(ID_{\mathrm{USP}}||R_4)||h(ID_{\mathrm{TCS}}||R_5)}{\rightleftharpoons}$ SM)

AS$_{11}$: USP$|\equiv$SM$|\Rightarrow$(USP $\overset{SK_{SU}}{\leftrightarrow}$ SM)

AS$_{12}$: SM$|\equiv$USP$|\Rightarrow$(USP $\overset{SK_{SU}}{\leftrightarrow}$ SM)

Based on message *Auth-1*, we obtain B$_{\mathrm{L1}}$.

B$_{\mathrm{L1}}$: TCS$\triangleleft(PID_{\mathrm{USP}}, A_3, R_4)_{K_{UT}}$

Deploying B$_{\mathrm{L1}}$ and AS$_1$ with MMR, B$_{\mathrm{L2}}$ is obtained.

B$_{\mathrm{L2}}$: TCS$|\equiv$USP$|\sim(PID_{\mathrm{USP}}, A_3, R_4)_{K_{UT}}$

Applying FR to B$_{\mathrm{L2}}$ and AS$_2$ yields B$_{\mathrm{L3}}$.

B$_{\mathrm{L3}}$: TCS$|\equiv\#(PID_{\mathrm{USP}}, A_3, R_4)_{K_{UT}}$

Using NVR on both B$_{\mathrm{L2}}$ and B$_{\mathrm{L3}}$, we obtain B$_{\mathrm{L4}}$.

B$_{\mathrm{L4}}$: TCS$|\equiv$USP$|\equiv(PID_{\mathrm{USP}}, A_3, R_4)_{K_{UT}}$

From message *Auth-2*, we can obtain B$_{\mathrm{L5}}$.

B$_{\mathrm{L5}}$: SM$\triangleleft(PID_{\mathrm{USP}}, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{TCS}}||R_5), PID_{\mathrm{SM}}, R_1)_{K_{\mathrm{TSM}}}$

The application of MMR on both B$_{\mathrm{L5}}$ and AS$_3$ results in B$_{\mathrm{L6}}$.

B$_{\mathrm{L6}}$: SM$|\equiv$TCS$|\sim(PID_{\mathrm{USP}}, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{TCS}}||R_5), PID_{\mathrm{SM}}, R_1)_{K_{\mathrm{TSM}}}$

To obtain B$_{\mathrm{L7}}$, FR is used on B$_{\mathrm{L6}}$ and AS$_4$.

B$_{\mathrm{L7}}$: SM$|\equiv\#(PID_{\mathrm{USP}}, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{TCS}}||R_5), PID_{\mathrm{SM}}, R_1)_{K_{\mathrm{TSM}}}$

On the other hand, NVR is applied to both B$_{\mathrm{L6}}$ and B$_{\mathrm{L7}}$ to obtain B$_{\mathrm{L8}}$.

B$_{\mathrm{L8}}$: SM$|\equiv$TCS$|\equiv(PID_{\mathrm{USP}}, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{TCS}}||R_5), PID_{\mathrm{SM}}, R_1)_{K_{\mathrm{TSM}}}$

Based on message *Auth-3*, we can obtain B$_{\mathrm{L9}}$.

B$_{\mathrm{L9}}$: TCS$\triangleleft(PID_{\mathrm{USP}}, PID_{\mathrm{SM}}, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{SM}}||R_6))_{K_{\mathrm{TSM}}}$

Applying MMR on B$_{\mathrm{L9}}$ and AS$_5$ yields B$_{\mathrm{L10}}$.

B$_{\mathrm{L10}}$: TCS$|\equiv$SM$|\sim(PID_{\mathrm{USP}}, PID_{\mathrm{SM}}, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{SM}}||R_6))_{K_{\mathrm{TSM}}}$

Using FR on B$_{\mathrm{L10}}$ and AS$_6$ results in B$_{\mathrm{L11}}$.

B$_{\mathrm{L11}}$: TCS$|\equiv\#(PID_{\mathrm{USP}}, PID_{\mathrm{SM}}, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{SM}}||R_6))_{K_{\mathrm{TSM}}}$

On the other hand, NVR is used on both BL$_{10}$ and B$_{\mathrm{L11}}$ to obtain B$_{\mathrm{L12}}$.

B$_{\mathrm{L12}}$: TCS$|\equiv$SM$|\equiv(PID_{\mathrm{USP}}, PID_{\mathrm{SM}}, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{SM}}||R_6))_{K_{\mathrm{TSM}}}$

From message *Auth-4*, we can obtain B$_{\mathrm{L13}}$.

B$_{\mathrm{L13}}$: USP$\triangleleft(A_3, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{TCS}}||R_5), h\,(ID_{\mathrm{SM}}||R_6))_{K_{\mathrm{UT}}}$

The application of MMR on B$_{\mathrm{L13}}$ and AS$_7$ yields B$_{\mathrm{L14}}$.

B$_{\mathrm{L14}}$: USP$|\equiv$TCS$|\sim(A_3, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{TCS}}||R_5), h\,(ID_{\mathrm{SM}}||R_6))_{K_{\mathrm{UT}}}$

To obtain B$_{\mathrm{L15}}$, FR is used in both B$_{\mathrm{L14}}$ and AS$_8$.

B$_{\mathrm{L15}}$: USP$|\equiv\#(A_3, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{TCS}}||R_5), h\,(ID_{\mathrm{SM}}||R_6))_{K_{\mathrm{UT}}}$

However, using NVR on B$_{\mathrm{L14}}$ and B$_{\mathrm{L15}}$ yields B$_{\mathrm{L16}}$.

B$_{\mathrm{L16}}$: USP$|\equiv$TCS$|\equiv(A_3, h\,(ID_{\mathrm{USP}}||R_4), h\,(ID_{\mathrm{TCS}}||R_5), h\,(ID_{\mathrm{SM}}||R_6))_{K_{\mathrm{UT}}}$

Since the session key is $SK_{\mathrm{SU}} = h\,(h\,(ID_{\mathrm{USP}}||R_4)||h\,(ID_{\mathrm{TCS}}||R_5)||h\,(ID_{\mathrm{SM}}||R_6)$, B$_{\mathrm{L17}}$ can be obtained from B$_{\mathrm{L12}}$, B$_{\mathrm{L16}}$, and AS$_9$.

B$_{\mathrm{L17}}$: USP$|\equiv$SM$|\equiv$(USP $\overset{SK_{SU}}{\leftrightarrow}$ SM); hence, **GL$_2$** is obtained.

From B$_{\mathrm{L4}}$, B$_{\mathrm{L8}}$, and AS$_{10}$, we obtain B$_{\mathrm{L18}}$.

B$_{\mathrm{L18}}$: SM$|\equiv$USP$|\equiv$(USP $\overset{SK_{SU}}{\leftrightarrow}$ SM); thus, **GL$_4$** is obtained.

Based on B$_{\mathrm{L17}}$ and AS$_{11}$, we can obtain B$_{\mathrm{L19}}$.

B$_{\mathrm{L19}}$: USP$|\equiv$(USP $\overset{SK_{SU}}{\leftrightarrow}$ SM), achieving **GL$_1$**.

Using B$_{\mathrm{L18}}$ and AS$_{12}$, we can obtain B$_{\mathrm{L20}}$.

B$_{\mathrm{L20}}$: SM$|\equiv$(USP $\overset{SK_{SU}}{\leftrightarrow}$ SM), attaining **GL$_3$**.

The effectual attainment of all the formulated security objectives implies that the USP, TCS, and SM have executed secure mutual authentication and can now proceed to exchange data.

4.2. Informal Security Analysis

In this sub-section, both the Dolev–Yao (DY) and Canetti–Krawczyk (CK) threat models are deployed to show the robustness of our protocol against typical smart grid attacks. Essentially, we make some assumptions about the attacker's capabilities and then show how our protocol counters the attacker's capabilities in both the DY and CK models. These attack capabilities are well articulated in [50].

Theorem 1. *Our scheme offers anonymity and untraceability.*

Proof. Let us assume that an adversary Ä has eavesdropped on Auth-1 = {PID_{USP}, B_4, B_5, C_1}, Auth-2 = {PID_{USP}, C_3, C_4, C_5}, Auth-3 = {D_1, D_2}, and Auth-4 = {D_3, D_4}. Here, B_4 = h (PID_{USP} | | A_3 | | K_{UT})⊕(R_4 | | PID_{SM}), B_5 = h (ID_{USP} | | R_4)⊕h (K_{UT} | | R_4), C_1 = h (PID_{USP} | | A_3 | | R_4 | | PID_{SM} | | K_{UT}), C_3 = h (PID_{SM} | | K_{TSM} | | R_1)⊕C_2, C_4 = (h (ID_{USP} | | R_4) | | h (ID_{TCS} | | R_5))⊕h (K_{TSM} | | R_1), C_5 = h (PID_{USP} | | C_2 | | K_{TSM}), D_1 = h (PID_{SM} | | K_{TSM} | | R_1)⊕ h (ID_{SM} | | R_6), D_2 = h (PID_{USP} | | PID_{SM} | | C_2* | | h (ID_{SM} | | R_6) | | K_{TSM}), D_3 = h (A_3 | | R_4)⊕(h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6) | | PID_{USP}*), and D_4 = h (PID_{USP} | | R_4 | | h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6) | | PID_{USP}* | | K_{UT}). The goal is to obtain the real identities of the USP, TCS, and SM_i that can facilitate the tracking of these entities. Evidently, these identities are encapsulated in other parameters (such as nonces R_1, R_4, R_5, and R_6) before being hashed. Towards the end of each session, secret parameter PID_{USP} is updated as PID_{USP}* = h (PID_{USP} | | R_4). As such, all the messages are dynamic for each session. □

Theorem 2. *Spoofing and impersonation attacks are thwarted.*

Proof. The main objective of these attacks is to spoof exchanged messages so as to masquerade oneself as a legitimate network entity. The following three cases demonstrate the resilience of our scheme against these threats. □

Case 1: Suppose that Ä wants to impersonate the USP through the interception of message Auth-1 = {PID_{USP}, B_4, B_5, C_1} sent from the USP towards the TCS over public channels. Here, B_4 = h (PID_{USP} | | A_3 | | K_{UT})⊕(R_4 | | PID_{SM}), B_5 = h (ID_{USP} | | R_4)⊕h (K_{UT} | | R_4), and C_1 = h (PID_{USP} | | A_3 | | R_4 | | PID_{SM} | | K_{UT}). However, Ä is unable to derive these parameters without knowledge of USP's real identity (ID_{USP}), the shared key between USP and TCS (K_{UT}), and random nonce R_4, among other values.

Case 2: Let us assume that Ä has intercepted messages Auth-2 = {PID_{USP}, C_3, C_4, C_5} and Auth-4 = {D_3, D_4} transmitted from the TCS towards the SM_i and TCS, respectively. Here, C_3 = h (PID_{SM} | | K_{TSM} | | R_1)⊕C_2, C_4 = (h (ID_{USP} | | R_4) | | h (ID_{TCS} | | R_5))⊕h (K_{TSM} | | R_1), C_5 = h (PID_{USP} | | C_2 | | K_{TSM}), D_1 = h (PID_{SM} | | K_{TSM} | | R_1)⊕ h (ID_{SM} | | R_6), and D_2 = h (PID_{USP} | | PID_{SM} | | C_2* | | h (ID_{SM} | | R_6) | | K_{TSM}). Afterwards, an attempt is made to construct bogus messages {$PID_{USP}{}^b$, $C_3{}^b$, $C_4{}^b$, $C_5{}^b$} and {$D_3{}^b$, $D_4{}^b$}. However, without TCS' real identity (ID_{TCS}), random nonces (R_1, R_4, R_5, and R_6), and shared key K_{TSM}, among other parameters, the derivation of these messages flops.

Case 3: Suppose that Ä has captured message Auth-3 = {D_1, D_2} sent from SM_i towards TCS over public channels. Here, D_1 = h (PID_{SM} | | K_{TSM} | | R_1)⊕ h (ID_{SM} | | R_6) and D_2 = h (PID_{USP} | | PID_{SM} | | C_2* | | h (ID_{SM} | | R_6) | | K_{TSM}). Similar to Case 2 above, Ä cannot construct valid message Auth-3 without knowledge of SM_i's real identity (ID_{SM}), shared key (K_{TSM}), and random nonces (R_1 and R_6).

Theorem 3. *Strong mutual entity verification is executed.*

Proof. In the proposed approach, all the network parties mutually authenticate one another. For instance, upon receiving message Auth-1 = {PID_{USP}, B_4, B_5, C_1} from the USP, the TCS computes C_1* = h (PID_{USP} | | A_3 | | R_4* | | PID_{SM}* | | K_{UT}) and validates USP by

checking if $C_1^* \stackrel{?}{=} C_1$. Conversely, upon receiving Auth-2 = {PID_{USP}, C_3, C_4, C_5} from the TCS, the SM_i computes C_5 = h (PID_{USP} | | C_2^* | | K_{TSM}) and verifies the TCS by confirming whether $C_5^* \stackrel{?}{=} C$. Similarly, the TCS receives message Auth-3 = {D_1, D_2} from SM_i, derives D_2^* = h (PID_{USP} | | PID_{SM} | | C_2 | | h (ID_{SM} | | R_6) | | K_{TSM}), and authenticates SM_i by checking if $D_2^* \stackrel{?}{=} D_2$. In contrast, the USP obtains message Auth-4 = {D_3, D_4} from the TCS, computes D_4^* = h (PID_{USP} | | R_4 | | h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6) | | PID_{USP}^* | | K_{UT}), and validates the TCS by confirming if $D_4^* \stackrel{?}{=} D_4$. □

Theorem 4. *The communicating entities negotiate session keys.*

Proof. In our protocol, the TCS, SM_i, and USP autonomously calculate the session key SK_{SU} = h (h (ID_{USP} | | R_4) | | h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6). For instance, after receiving message Auth-2 from the TCS, the SM_i computes the session key as SK_{SU} = h (h (ID_{USP} | | R_4) | | h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6), together with parameters D_1 and D_2. However, upon receiving Auth-3 from the SM_i, the TCS computes the session key as SK_{SU} = h (h (ID_{USP} | | R_4) | | h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6), together with values PID_{USP}^*, A_3^*, D_3, and D_4. Similarly, the USP receives message Auth-4 from the TCS and computes the session key as SK_{SU} = h (h (ID_{USP} | | R_4) | | h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6), together with values A_3^*, B_2^*, and B_3^*. □

Theorem 5. *Our scheme can withstand forgery and eavesdropping attacks.*

Proof. Let us assume that adversary Ä wants to forge session key SK_{SU} = h (h (ID_{USP} | | R_4) | | h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6). Evidently, Ä must have access to identities ID_{USP}, ID_{TCS}, and ID_{SM}. In addition, random nonces R_4, R_5, and R_6 must be obtained by Ä. However, these identities and nonces cannot be obtained by eavesdropping messages Auth-1 = {PID_{USP}, B_4, B_5, C_1}, Auth-2 = {PID_{USP}, C_3, C_4, C_5}, Auth-3 = {D_1, D_2}, and Auth-4 = {D_3, D_4} exchanged over public channels. Let us assume that Ä has captured long-term secret keys K_{TCS}, K_{UT}, K_{TSM}, and K_{SM}. However, none of these keys is incorporated in the negotiated session key SK_{SU}. As such, the session keys derived in our protocol are secured. □

Theorem 6. *MitM and replay attacks are thwarted.*

Proof. Suppose that Ä has the ability of intercepting and modifying authentication messages Auth-1 = {PID_{USP}, B_4, B_5, C_1}, Auth-2 = {PID_{USP}, C_3, C_4, C_5}, Auth-3 = {D_1, D_2}, and Auth-4 = {D_3, D_4} exchanged over insecure public channels. Here, B_4 = h (PID_{USP} | | A_3 | | K_{UT})⊕(R_4 | | PID_{SM}), B_5 = h (ID_{USP} | | R_4)⊕h (K_{UT} | | R_4), C_1 = h (PID_{USP} | | A_3 | | R_4 | | PID_{SM} | | K_{UT}), C_3 = h (PID_{SM} | | K_{TSM} | | R_1)⊕C_2, C_4 = (h(ID_{USP} | | R_4) | | h (ID_{TCS} | | R_5))⊕h (K_{TSM} | | R_1), C_5 = h (PID_{USP} | | C_2 | | K_{TSM}), D_1 = h (PID_{SM} | | K_{TSM} | | R_1)⊕h (ID_{SM} | | R_6), D_2 = h (PID_{USP} | | PID_{SM} | | C_2^* | | h (ID_{SM} | | R_6) | | K_{TSM}), D_3 = h (A_3 | | R_4)⊕(h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6) | | PID_{USP}^*), and D_4 = h (PID_{USP} | | R_4 | | h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6) | | PID_{USP}^* | | K_{UT}). It is clear that all these messages incorporate random nonces such as R_1, R_4, R_5, and R_6. In addition, any successful modification of these messages requires knowledge of identities (ID_{USP}, ID_{TCS}, ID_{SM}) and shared keys (K_{UT}, K_{TSM}), all of which are unavailable to Ä. □

Theorem 7. *Privileged insider attacks are effectively prevented.*

Proof. Let us assume that some privileged insider Ä has accessed USP's pseudo-identity (PID_{USP}) during the registration phase. In addition, Ä has access to {A_5, B_1, B_2, B_3} stored in the USP's database. With all these parameters, Ä makes some attempts in deriving session key SK_{SU} = h (h (ID_{USP} | | R_4) | | h (ID_{TCS} | | R_5) | | h (ID_{SM} | | R_6). However, Ä does not know real identities (ID_{USP}, ID_{TCS}, ID_{SM}) and random nonces (R_4, R_5, R_6). Therefore, this attack will fail. □

Theorem 8. *The proposed scheme can resist de-synchronization and backdoor-based DoS attacks.*

Proof. The objective of these threats is to alter and block exchanged messages so as to interfere with future mutual verification processes among the USP, TCS, and SM_i. This can be occasioned by some SG and SM firmware-containing backdoors. Suppose that Ä wants to de-synchronize the next authentication session by modifying Auth-1, Auth-2, and Auth-3. However, Theorem 6 demonstrates the difficulty in modifying these messages devoid of random nonces, real identities, and shared keys. Let us assume that Ä wants to block all the transmitted messages so as to interfere with the synchronization procedures among the USP, TCS, and SM_i. To achieve this, USP's pseudo-identity PID_{USP}, incorporated in all four authentication messages, is utilized. However, in Step 7 above, our scheme refreshes this parameter as $PID_{USP}^* = h(PID_{USP} || R_4)$ and includes it in parameters $D_3 = h(A_3 || R_4) \oplus (h(ID_{TCS} || R_5) || h(ID_{SM} || R_6) || PID_{USP}^*)$ and $D_4 = h(PID_{USP} || R_4 || h(ID_{TCS} || R_5) || h(ID_{SM} || R_6) || PID_{USP}^* || K_{UT})$. Thereafter, authentication message Auth-4 = $\{D_3, D_4\}$ is relayed to the USP. Provided that $PID_{USP}^*$is valid, it then passes the $D_4^* \stackrel{?}{=} D_4$ check. Otherwise message Auth-4 is rejected at the USP. Upon the successful verification of PID_{USP}^*, the USP derives and sends $D_5 = h(SK_{SU} || PID_{USP}^*)$ to the TCS for further validation through the $D_5^* \stackrel{?}{=} D_5$ check. It is only after the successful verification of PID_{USP}^* that TCS deletes parameter set $\{PID_{USP}, A_3\}$ from its database. Otherwise, the TCS continues to store these two values to stay in sync with the USP. □

Theorem 9. *Offline guessing attacks are resisted.*

Proof. The assumption made in these attacks is that Ä is able to obtain $\{A_5, B_1, B_2, B_3\}$ from the USP's database. Here, $A_3 = h(PID_{USP} || K_{UT})$, $A_4 = h(K_{USP} || R_3)$, $A_5 = R_3 \oplus h(ID_{USP} || K_{USP})$, $B_1 = h(ID_{USP} || K_{USP} || R_3 || A_4)$, $B_2 = A_3 \oplus h(R_3 || A_4)$, and $B_3 = K_{UT} \oplus h(A_3 || A_4)$. It is clear that these messages are encapsulated with random nonce, ID_{USP}, and K_{USP}. In accordance with Theorem 5, Ä cannot easily ascertain identity ID_{USP} and random nonces. Since K_{USP} is the USP's private key, it is not available to Ä and cannot be eavesdropped over public channels. □

Theorem 10. *Our scheme is robust against KSSTI and ephemeral secret leakage attacks.*

Proof. The purpose of this attack is to enable adversary Ä to access session-specific tokens such as nonces R_1, R_2, R_3, R_4, R_5, and R_6. Thereafter, Ä attempts some KSSTI under the CK-adversarial model. This might include an attempt to derive the session key $SK_{SU} = h(h(ID_{USP} || R_4) || h(ID_{TCS} || R_5) || h(ID_{SM} || R_6)$. However, even with these ephemerals, Ä cannot derive SK_{SU}. This is because the real identities of the SM_i, TCS, and USP (ID_{USP}, ID_{TCS}, ID_{SM}) are required. Based on Theorem 5, Ä cannot easily ascertain these identities, and, hence, this attack flops. □

Theorem 11. *The proposed protocol can withstand physical attacks.*

Proof. The assumption made here is that adversary Ä has physically obtained the SM_i upon which the stored values $\{A_1, A_2, PID_{SM}\}$ in its memory are extracted via a power analysis. Here, $A_1 = R_1 \oplus h(ID_{SM} || K_{SM})$, $A_2 = K_{TSM} \oplus h(R_1 || K_{SM})$, and $PID_{SM} = h(ID_{SM} || R_1)$. The next objective is to ascertain SM_i's identity (ID_{SM}), shared key (K_{TSM}), and SM's private key (K_{SM}). However, these values are masked with random nonces before being hashed. Since reversing the one-way hashing function is computationally cumbersome, our scheme is robust against physical attacks. □

5. Performance Evaluations

Storage, computation, supported security, and privacy features, as well as communication complexities are most often utilized as metrics to evaluate authentication protocols. As such, we deploy such metrics in our comparative performance evaluations as detailed below.

5.1. Computation Overheads

During the mutual verification and key setup phase, our scheme executes only one-way hashing (T_H) operations. Specifically, $7T_H$ and $16T_H$ operations are executed on the smart meter and utility service provider sides, respectively. The time complexities of the diverse cryptographic functions in the smart meter are computed on a 1 GB RAM, 1.2 GHz CPU, Quad-core Raspberry Pi-3, while the USP cryptographic primitives are computed on an 8 GB RAM, Core i7-6700 laptop equipped with a 3.40 GHz CPU. Under these two environments, the execution durations are presented in Table 4.

Table 4. Execution durations.

Scheme	Costs (ms)	
	SM	USP
Bilinear pairing operations, T_{BP}	95.72100	9.52800
ECC point addition, T_{ECA}	0.13400	0.00700
One-way hash function, T_H	0.34500	0.03900
ECC point multiplication, T_{PM}	2.70000	0.70500
Symmetric encryption, T_{SE}	0.41000	0.00460
Symmetric decryption, T_{SD}	0.41000	0.00460
Esch256 one-way hash function, T_{HE}	0.33000	0.03200
Physically unclonable function, T_{PUF}	0.00049	-
Counter-mode encryption with authentication tag, T_{CO}	0.34900	0.04100
Bio-metric key generation and reproduction, T_{REP}	2.70000	0.70500
Modular exponential, T_E	30.7920	0.31200
Scalar multiplication, T_{SM}	2.70000	0.70500

Using the execution durations in Table 4 as a basis, the total computation complexity of our scheme is 2.805 ms. Table 5 details the derivation and comparison of the computation complexities of other peer approaches.

Table 5. Computation complexities.

Scheme	SM	USP	Total (ms)
Baghestani et al. [1]	$5T_H + 2T_{PM}$	$11T_H + 2T_{PM}$	$17T_H + 4T_{PM} \approx 8.964$
Xia et al. [6]	$19T_{PM}$	$17T_{PM}$	$10T_H + 8T_{PM} \approx 63.285$
Mohammadali et al. [10]	$3T_H + 2T_{PM}$	$4T_H + 3T_{PM}$	$7T_H + 5T_{PM} \approx 8.706$
Kumar et al. [13]	$5T_H + 2T_{PM}$	$6T_H + 2T_{PM}$	$11T_H + 4T_{PM} \approx 8.769$
Tsai & Lo [22]	$5T_H + 4T_{PM} + T_E$	$2T_{BP} + 3T_{PM} + T_E + 5T_H$	$2T_{BP} + 7T_{PM} + 2T_E + 10T_H \approx 237.381$
Tanveer & Alasmary [29]	$2T_{HE} + 2T_{CO} + T_{REP} + T_{PUF}$	$5T_{HE} + 2T_{CO}$	$7T_{HE} + 4T_{CO} + T_{REP} + T_{PUF} \approx 4.300$
Chaudhry et al. [31]	$4T_H + 2T_{SE} + 3T_{PM}$	$6T_H + 2T_{SE} + 4T_{PM}$	$10T_H + 4T_{SE} + 7T_{PM} \approx 13.363$
Taqi & Jalili [32]	$4T_H + T_{SE} + T_{SD} + 3T_{PM}$	$3T_H + T_{SE} + T_{SD} + 3T_{PM}$	$7T_H + 2T_{SE} + 2T_{SD} + 6T_{PM} \approx 12.5412$
Chen et al. [33]	$7T_H + T_{SD}$	$9T_H + 2T_{SE} + T_{SD}$	$16T_H + 2T_{SE} + 2T_{SD} \approx 3.1898$
Park et al. [47]	$5T_H + 2T_{SM}$	$6T_H + 2T_{SM}$	$11T_H + 4T_{SM} \approx 8.769$
Proposed	$7T_H$	$16T_H$	$16T_H + 7T_H \approx 3.0390$

As demonstrated in Figure 4, the technique in [22] has the longest execution time of 237.381 ms. This can be explained by the computationally extensive bilinear pairings in [22]. This is followed by the protocols in [6], [31], [32], [1], [13], [47], [10], [29], and [33] respectively. Conversely, our protocol incurs the least computation complexities.

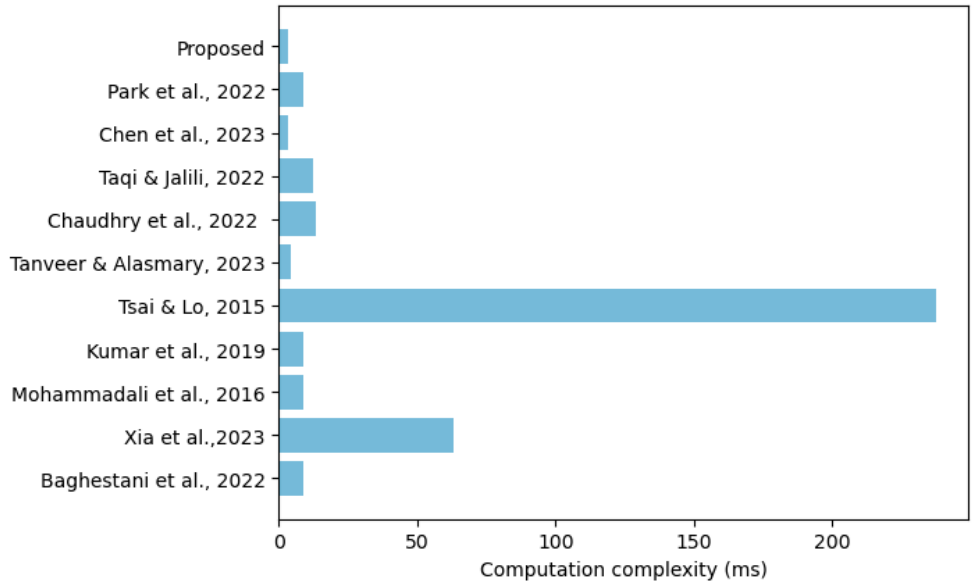

Figure 4. Computational complexities [1,6,10,13,22,29,31–33,47].

Even though the approach in [33] has a relatively lower execution time, it cannot withstand guessing, KSSTI, eavesdropping, ephemeral secret leakage, spoofing, and physical capture attacks. In the SG environment, the majority of components does not have a high computation power; hence, our protocol is the most suitable for deployment.

5.2. Communication Overheads

In our scheme, messages *Auth-1*, *Auth-2*, *Auth-3*, and *Auth-4* are exchanged during the verification and key setup phase. The specific details of these messages are as follows.

Auth-1 = $\{PID_{USP}, B_4, B_5, C_1\}$

Auth-2 = $\{PID_{USP}, C_3, C_4, C_5\}$

Auth-3 = $\{D_1, D_2\}$

Auth-4 = $\{D_3, D_4\}$

Here, $PID_{USP} = h(ID_{USP}||R_3)$, $B_4 = h(PID_{USP}||A_3||K_{UT}) \oplus (R_4||PID_{SM})$, $B_5 = h(ID_{USP}||R_4) \oplus h(K_{UT}||R_4)$, $C_1 = h(PID_{USP}||A_3||R_4||PID_{SM}||K_{UT})$, $C_3 = h(PID_{SM}||K_{TSM}||R_1) \oplus C_2$, $C_4 = (h(ID_{USP}||R_4)||h(ID_{TCS}||R_5)) \oplus h(K_{TSM}||R_1)$, $C_5 = h(PID_{USP}||C_2||K_{TSM})$, $D_1 = h(PID_{SM}||K_{TSM}||R_1) \oplus h(ID_{SM}||R_6)$, $D_2 = h(PID_{USP}||PID_{SM}||C_2^*||h(ID_{SM}||R_6)||K_{TSM})$, $D_3 = h(A_3||R_4) \oplus (h(ID_{TCS}||R_5)||h(ID_{SM}||R_6)||PID_{USP}^*)$, and $D_4 = h(PID_{USP}||R_4||h(ID_{TCS}||R_5)||h(ID_{SM}||R_6)||PID_{USP}^*||K_{UT})$.

Using the values in [23,33,39], the hashing, symmetric encryption, point multiplication, timestamps, and symmetric decryption output lengths are 160 bits, 128 bits, 320 bits, 32 bits, and 128 bits, correspondingly. As such, *Auth-1* = 160 + 160 + 160 + 160 = 640 bits; *Auth-2* = {160 + 160 + 160 + 160} = 640 bits; *Auth-3* = {160 + 160} = 320 bits; and *Auth-4* = {160 + 160} = 320 bits. Consequently, the overall communication complexity of our technique is 1920 bits. Table 6 presents the comparative analysis of the incurred communication complexities of our protocol together with those of its peer approaches.

Table 6. Communication complexities.

Scheme	Messages Exchanged	Total (Bits)
Baghestani et al. [1]	SM $\overset{864}{\rightarrow}$ USP $\overset{832}{\leftrightarrow}$ SM	1696
Xia et al. [6]	SM $\overset{1664}{\rightarrow}$ USP $\overset{1152}{\leftrightarrow}$ SM	2816
Mohammadali et al. [10]	SM $\overset{768}{\rightarrow}$ USP $\overset{608}{\leftrightarrow}$ SM $\overset{160}{\leftrightarrow}$ USP	1536
Kumar et al. [13]	SM $\overset{512}{\rightarrow}$ USP $\overset{672}{\leftrightarrow}$ SM $\overset{192}{\leftrightarrow}$ USP	1376

Table 6. *Cont.*

Scheme	Messages Exchanged	Total (Bits)
Tsai & Lo [22]	SM $\overset{480}{\rightarrow}$ USP $\overset{480}{\leftrightarrow}$ SM $\overset{320}{\leftrightarrow}$ USP	1280
Tanveer & Alasmary [29]	USP $\overset{544}{\rightarrow}$ TCS $\overset{662}{\leftrightarrow}$SM	1206
Chaudhry et al. [31]	SM $\overset{768}{\rightarrow}$ USP $\overset{768}{\leftrightarrow}$SM	1536
Taqi & Jalili [32]	SM $\overset{512}{\rightarrow}$ USP $\overset{896}{\leftrightarrow}$SM $\overset{576}{\leftrightarrow}$ USP	1984
Chen et al. [33]	SM $\overset{864}{\rightarrow}$USP $\overset{704}{\leftrightarrow}$SM $\overset{160}{\rightarrow}$ USP $\overset{160}{\leftrightarrow}$ SM	1888
Park et al. [47]	SM $\overset{512}{\rightarrow}$ USP $\overset{672}{\leftrightarrow}$SM $\overset{192}{\leftrightarrow}$USP	1376
Proposed	USP $\overset{640}{\rightarrow}$ TCS $\overset{640}{\leftrightarrow}$SM $\overset{320}{\leftrightarrow}$TCS $\overset{320}{\leftrightarrow}$USP	1920

As evidenced in Figure 5, the technique in [6] exhibits the largest communication overheads of 2816 bits. This is followed by the protocols in [32], our proposed scheme, [33], [1], [10], [31], [13], [47], [22], and [29], in this order. Even though the technique in [29] incurs the lowest communication overheads, its design does not consider guessing, eavesdropping, and spoofing attacks. Similarly, the security scheme in [22] is defenseless against privileged insider, de-synchronization, DoS, guessing, spoofing, KSSTI, eavesdropping, EPSL, physical capture, and forgery attacks. In the same breadth, the protocol in [47] is not analyzed against attacks such as de-synchronization, privileged insider, DoS, guessing, eavesdropping, physical capture, ephemeral secret leakage, spoofing, replay, and forgery. In addition, it does not offer anonymity. On its part, the approach in [13] fails to provide session key agreement and mutual authentication. In addition, it is not analyzed against de-synchronization, DoS, privileged insider, guessing, KSSTI, eavesdropping, spoofing, and forgery attacks. Concerning the protocol in [33], it is defenseless against guessing, KSSTI, eavesdropping, EPSL, spoofing, and physical capture attacks. Likewise, the protocol in [1] cannot withstand privileged insider, physical capture, guessing, KSSTI, eavesdropping, spoofing, and forgery attacks. Regarding the protocol in [10], it cannot protect against DoS, spoofing, privileged insider, guessing, KSSTI, eavesdropping, EPSL, physical capture, and forgery.

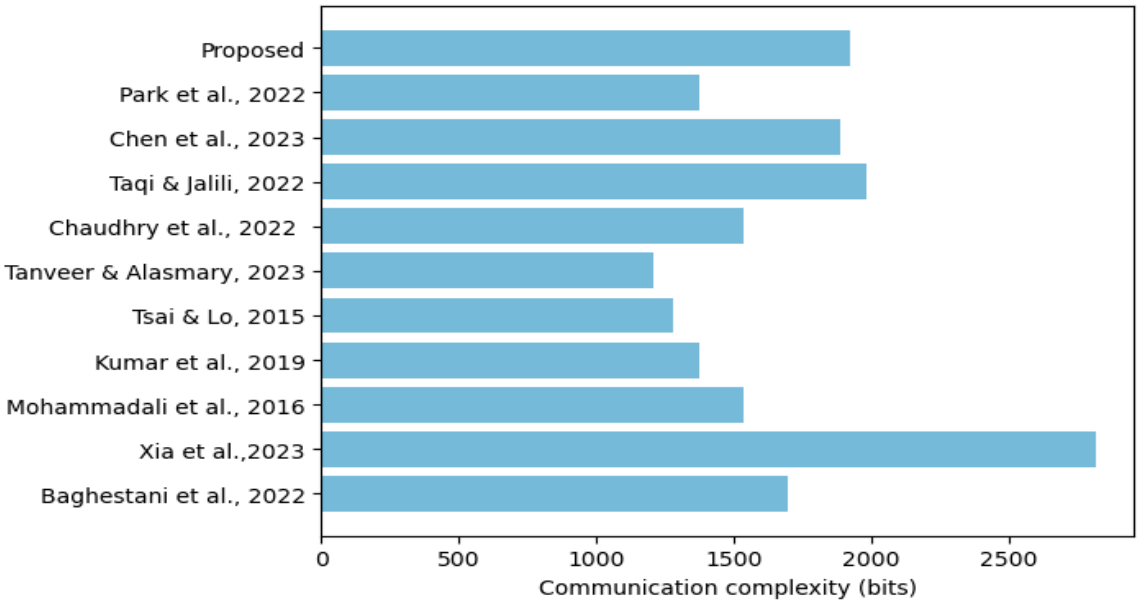

Figure 5. Communication complexities [1,6,10,13,22,29,31–33,47].

In addition, it cannot offer entity untraceability and anonymity. Finally, the scheme in [31] is not robust against spoofing, de-synchronization, DoS, privileged insider, guessing,

eavesdropping, ESPL, and forgery attacks. Evidently, our protocol provides a good balance between security and communication complexity.

5.3. Storage Overheads

In our scheme, value sets $\{A_5, B_1, B_2, B_3\}$ and $\{A_1, A_2, PID_{SM}\}$ are stored in the USP database and smart meter memory, respectively. Here, $A_5 = B_1 = B_2 = B_3 = A_1 = A_2 = PID_{SM}$ = 160 bits. Consequently, the cumulative storage complexity in our scheme is 1120 bits, or 140 bytes. Table 7 shows the derivation of the storage complexities of our scheme as well as those ones of its peers.

Table 7. Storage overheads.

Scheme	Stored Parameters	Total (Bits)
Baghestani et al. [1]	**SM:** {H_1, H_2, n, E, P, F_P, SM_{sj}, x_j, y_j} **USP:** {SM_{IDj},M_k}	2432
Xia et al. [6]	**SM:** {x_S, R_2} **USP:** {x_C}	896
Mohammadali et al. [10]	**SM:** {S_M, R_M, y_M, r_M} **USP:** {y_{AHE}, r_{AHE}}	1600
Kumar et al. [13]	**SM:** {RID_i, TC_i, h (·),E_p (a, b),G} **USP:** {RID_j, TC_j, {RID_i | i = 1, 2, . . ., l}, h (·),E_p (a, b),G}	2240
Tsai & Lo [22]	**SM:** {G_1, G_2, P, e,H, H_1, H_2, H_3, H_4, q, Ppub, g} **USP:** {G_1, G_2, P, e, H, H_1, H_2, H_3, H_4, q, Ppub, g}, K_j, H_1 (SID_j)P + Ppub	6112
Tanveer & Alasmary [29]	**SM:** {CH_{SMi}, TID_{SMi}, RN_r, HD} **USP:** {SID_i, B_i, RN_r}	1056
Chaudhry et al. [31]	**SM:** {E, P, F_p, n, $SMpr_j$, σ_j, $idST_j$, ST_j, H (.), $SMID_j$, $Pidst_j$} **USP:** {M_k}	2176
Taqi & Jalili [32]	**SM:** {a_i, A_i} **USP:** {a_j, A_j}	896
Chen et al. [33]	**SM:** {ID_i, N_1, X_i} **USP:** {S_i}	832
Park et al. [47]	**SM:** {PIDi, LS_{SMi}, H, E(a, b), G} **USP:** {$PCUID_j$, H, E(a,b), G, $PID_{i=1...l}$}	2240
Proposed	**SM:** {A_1, A_2, PID_{SM}} **USP:** {A_5, B_1, B_2, B_3}	1120

The specific details of the various parameters stored in the related schemes are described in Table 8.

As revealed in Figure 6, the approach in [22] incurs the highest storage complexity of 6112 bits. This is followed by the protocols in [1], [47], [13], [31], [10], the proposed scheme, [29], [6], [32], and [33] respectively. The high storage cost in [22] is due to the numerous security tokens that have to be stored in the end devices.

Table 8. Details of stored parameters.

Symbol	Details
x_S, SM_{sj}, $SMpr_j$, S_M	*SM*'s private keys
R_M,	*SM*'s public key
R_2	Keying parameter based on smart meter's public key
x_C, K_j	*USP*'s private keys
H_1, H_2, H, H (..), h (.), H_1, H_2, H_3, H_4	One-way hash functions
n, E, P	Elliptic curve E and a point P of order n
F_P	Finite field
x_j, y_j, X_i, LS_{SMi}, σ_j, ST_j, A_i, A_j, SID_i, B_i, y_M, y_{AHE}, g	Derived intermediary parameters
SM_{IDj}, ID_i, $SMID_j$	*SM*'s unique identity
$idST_j$	Unique identifier for *SM*
SID_j	*USP*'s unique identity
M_k	Master key
N_1, a_i, a_j, RN_r, r_M, r_{AHE}	Random numbers
S_i	SM's unique identification stored in the table
PIDi, $Pidst_j$, TID_{SMi}, RID_i	Pseudo-identities for *SM*
$PCUID_j$, RID_j	Pseudo-identities for *USP*
TC_i	SM's temporal credential
TC_j	USP's temporal credential
$E(a, b)$, G, E_p (a, b)	Elliptic curve with base point G.
P, G_1, G_2	Generator of G_1, cyclic additive group, and cyclic multiplicative group, respectively
q	Prime order of G_1 and G_2
e	Pairing operation
$Ppub$	Public key of the trust anchor
CH_{SMi}	Registration authority (*RA*) challenge parameter
HD	Helper data

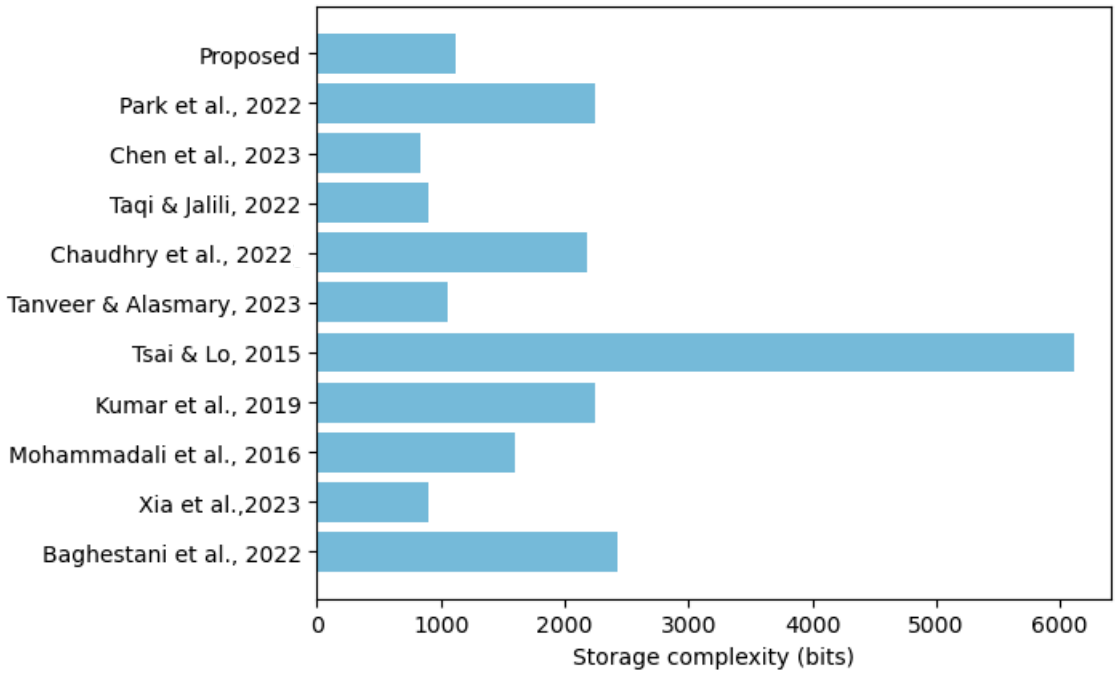

Figure 6. Storage complexities [1,6,10,13,22,29,31–33,47].

Although the protocols in [6,29,32,33] have slightly lower storage complexities compared to our scheme, they are susceptible to numerous threats, as shown in Table 9. Since smart devices such as SMs in the grid system have limited storage, our scheme is ideal for implementation in this environment.

Table 9. Supported functionalities.

	[10]	[13]	[22]	[29]	[6]	[1]	[31]	[32]	[33]	[47]	Proposed
Functionality											
Session key agreement	√	×	√	√	√	√	√	√	√	√	√
Anonymity and untraceability	×	√	√	√	×	√	√	√	√	×	√
Key security	√	√	√	√	√	√	√	√	√	√	√
Mutual authentication	√	×	√	√	√	√	√	√	√	√	√
Formal verification	√	√	√	√	×	√	√	√	√	√	√
Resilience against											
De-synchronization	√	×	×	√	×	√	×	×	√	×	√
Backdoor-based DoS	×	×	×	√	×	√	×	√	√	×	√
Privileged insider	×	×	×	√	×	×	×	×	√	×	√
Guessing	×	×	×	×	×	×	×	√	×	×	√
KSSTI	×	×	×	√	×	×	√	×	×	√	√
Eavesdropping	×	×	×	×	×	×	×	×	×	×	√
Ephemeral secret leakage	×	√	×	√	×	√	×	×	×	×	√
Spoofing	×	×	×	×	×	×	×	×	×	×	√
Physical capture	×	√	×	√	×	×	√	√	×	×	√
Impersonation	√	√	√	√	√	√	√	×	√	√	√
Replay	√	√	√	√	√	√	√	√	√	×	√
MitM	√	√	√	√	√	√	√	×	√	√	√
Forgery	×	×	×	√	×	×	×	×	×	×	√

√ Feature supported; × Feature not supported or not considered.

5.4. Supported Functionalities

The protocol developed in this paper offers a wide range of salient security and privacy features and is robust against several attacks. Table 9 provides a comparative evaluation of the security characteristics of our scheme as well as its resilience to attacks.

As revealed in Table 9, the scheme in [6] supports only six features and, hence, is the least secure. This is followed by the protocol in [47], which supports seven features. In contrast, the schemes in [10,13,22] support eight features and, hence, have been rated third. This is followed by the protocols in [32], [31], [1], [33], and [29], which offer support for 9, 10, 11, 11, and 15 characteristics, correspondingly.

Conversely, our scheme supports all 18 security and privacy features. Using the 15 features provided in [29] as a basis, our scheme offers a 20% improvement in smart grid networks' security posture.

6. Conclusions

The consumer consumption report and power adjustments data exchanged between SMs and SPs are exposed to many privacy and security threats. This is due to the utilization of insecure communication channels for the message communication procedures. Such attacks include ephemeral secret leakage, denial of service, eavesdropping, tampering, and forgery. To address this challenge, many security solutions have been developed recently. Nevertheless, the majority of these solutions has been shown to be inefficient or have some susceptibilities that render them inappropriate for smart meters. In this paper, a security protocol that is provably secure has been developed. It has also been demonstrated to be resilient against attacks such as privileged insider, de-synchronization, DoS, guessing, KSSTI, eavesdropping, EPSL, spoofing, physical capture, impersonation, replay, MitM, and forgery. In addition, it provides security functionalities such as anonymity, strong authentication, session key agreement, session key security, and untraceability. In terms of performance, it incurs the least computational costs and relatively lower storage and

communication costs. Future work will feature the development of novel approaches that can further reduce the incurred storage and communication overheads.

Author Contributions: All the authors have contributed equally to this article. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon request.

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

References

1. Baghestani, S.H.; Moazami, F.; Tahavori, M. Lightweight authenticated key agreement for smart metering in smart grid. *IEEE Syst. J.* **2022**, *16*, 4983–4991. [CrossRef]
2. Sun, C.-C.; Hahn, A.; Liu, C.-C. Cyber security of a power grid: State-of-the-art. *Int. J. Electr. Power Energy Syst.* **2018**, *99*, 45–56. [CrossRef]
3. Salem, F.M.; Ibrahim, E.; Elghandour, O. A lightweight authenticated key establishment scheme for secure smart grid communications. *Int. J. Saf. Secur. Eng.* **2020**, *10*, 549–558. [CrossRef]
4. Numan, M.; Baig, M.F.; Yousif, M. Reliability evaluation of energy storage systems combined with other grid flexibility options: A review. *J. Energy Storage* **2023**, *63*, 107022. [CrossRef]
5. Nyangaresi, V.O.; Abduljabbar, Z.A.; Al Sibahee, M.A.; Abood, E.W.; Abduljaleel, I.Q. Dynamic ephemeral and session key generation protocol for next generation smart grids. In Proceedings of the International Conference on Ad Hoc Networks, Virtual Event, 6–7 December 2021; pp. 188–204. [CrossRef]
6. Xia, Z.; Liu, T.; Wang, J.; Chen, S. A secure and efficient authenticated key exchange scheme for smart grid. *Heliyon* **2023**, *9*, e17240. [CrossRef]
7. Huseinović, A.; Mrdović, S.; Bicakci, K.; Uludag, S. A survey of denial-of-service attacks and solutions in the smart grid. *IEEE Access* **2020**, *8*, 177447–177470. [CrossRef]
8. Zhu, L.; Li, M.; Zhang, Z.; Xu, C.; Zhang, R.; Du, X.; Guizani, N. Privacy-preserving authentication and data aggregation for fog-based smart grid. *IEEE Commun. Mag.* **2019**, *57*, 80–85. [CrossRef]
9. Kumar, P.; Lin, Y.; Bai, G.; Paverd, A.; Dong, J.S.; Martin, A. Smart grid metering networks: A survey on security, privacy and open research issues. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2886–2927. [CrossRef]
10. Mohammadali, A.; Haghighi, M.S.; Tadayon, M.H.; Mohammadi-Nodooshan, A. A novel identity-based key establishment method for advanced metering infrastructure in smart grid. *IEEE Trans. Smart Grid* **2016**, *9*, 2834–2842. [CrossRef]
11. Tanveer, M.; Ahmad, M.; Khalifa, H.S.; Alkhayyat, A.; Abd El-Latif, A.A. A new anonymous authentication framework for secure smart grids applications. *J. Inf. Secur. Appl.* **2022**, *71*, 103336. [CrossRef]
12. Abbasinezhad-Mood, D.; Nikooghadam, M. An anonymous ECC-based self-certified key distribution scheme for the smart grid. *IEEE Trans. Ind. Electron.* **2018**, *65*, 7996–8004. [CrossRef]
13. Kumar, N.; Aujla, G.S.; Das, A.K.; Conti, M. ECCAuth: A secure authentication protocol for demand response management in a smart grid system. *IEEE Trans. Ind. Inform.* **2019**, *15*, 6572–6582. [CrossRef]
14. Zhang, L.; Zhu, Y.; Ren, W.; Wang, Y.; Choo, K.-K.R.; Xiong, N.N. An energy-efficient authentication scheme based on Chebyshev chaotic map for smart grid environments. *IEEE Internet Things J.* **2021**, *8*, 17120–17130. [CrossRef]
15. Gope, P.; Sikdar, B. A privacy-aware reconfigurable authenticated key exchange scheme for secure communication in smart grids. *IEEE Trans. Smart Grid* **2021**, *12*, 5335–5348. [CrossRef]
16. Kaveh, M.; Mosavi, M.R. A lightweight mutual authentication for smart grid neighborhood area network communications based on physically unclonable function. *IEEE Syst. J.* **2020**, *14*, 4535–4544. [CrossRef]
17. Tahavori, M.; Moazami, F. Lightweight and secure PUF-based authenticated key agreement scheme for smart grid. *Peer-To-Peer Netw. Appl.* **2020**, *13*, 1616–1628. [CrossRef]
18. Gope, P.; Sikdar, B. Privacy-aware authenticated key agreement scheme for secure smart grid communication. *IEEE Trans. Smart Grid* **2018**, *10*, 3953–3962. [CrossRef]
19. Nyangaresi, V.O.; Petrovic, N. Efficient PUF based authentication protocol for internet of drones. In Proceedings of the 2021 International Telecommunications Conference (ITC-Egypt), Alexandria, Egypt, 13–15 July 2021; pp. 1–4. [CrossRef]
20. Li, X.; Wu, F.; Kumari, S.; Xu, L.; Sangaiah, A.K.; Choo, K.-K.R. A provably secure and anonymous message authentication scheme for smart grids. *J. Parallel Distrib. Comput.* **2019**, *132*, 242–249. [CrossRef]
21. Wu, L.; Wang, J.; Zeadally, S.; He, D. Anonymous and efficient message authentication scheme for smart grid. *Secur. Commun. Netw.* **2019**, *2019*, 4836016. [CrossRef]
22. Tsai, J.-L.; Lo, N.-W. Secure anonymous key distribution scheme for smart grid. *IEEE Trans. Smart Grid* **2015**, *7*, 906–914. [CrossRef]
23. Odelu, V.; Das, A.K.; Wazid, M.; Conti, M. Provably secure authenticated key agreement scheme for smart grid. *IEEE Trans. Smart Grid* **2016**, *9*, 1900–1910. [CrossRef]

24. Abduljabbar, Z.A.; Nyangaresi, V.O.; Jasim, H.M.; Ma, J.; Hussain, M.A.; Hussien, Z.A.; Aldarwish, A.J. Elliptic curve cryptography-based scheme for secure signaling and data exchanges in precision agriculture. *Sustainability* **2023**, *15*, 10264. [CrossRef]
25. Deng, L.; Gao, R. Certificateless two-party authenticated key agreement scheme for smart grid. *Inf. Sci.* **2021**, *543*, 143–156. [CrossRef]
26. Mahmood, K.; Chaudhry, S.A.; Naqvi, H.; Kumari, S.; Li, X.; Sangaiah, A.K. An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. *Future Gener. Comput. Syst.* **2018**, *81*, 557–565. [CrossRef]
27. Abbasinezhad-Mood, D.; Nikooghadam, M. Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications. *Future Gener. Comput. Syst.* **2018**, *84*, 47–57. [CrossRef]
28. Banerjee, S.; Odelu, V.; Das, A.K.; Chattopadhyay, S.; Park, Y. An efficient, anonymous and robust authentication scheme for smart home environments. *Sensors* **2020**, *20*, 1215. [CrossRef] [PubMed]
29. Tanveer, M.; Alasmary, H. LACP-SG: Lightweight authentication protocol for smart grids. *Sensors* **2023**, *23*, 2309. [CrossRef] [PubMed]
30. Srinivas, J.; Das, A.K.; Li, X.; Khan, M.K.; Jo, M. Designing anonymous signature-based authenticated key exchange scheme for Internet of Things-enabled smart grid systems. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4425–4436. [CrossRef]
31. Chaudhry, S.A.; Yahya, K.; Garg, S.; Kaddoum, G.; Hassan, M.M.; Zikria, Y.B. LAS-SG: An elliptic curve-based lightweight authentication scheme for smart grid environments. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1504–1511. [CrossRef]
32. Taqi, S.A.M.; Jalili, S. LSPA-SGs: A lightweight and secure protocol for authentication and key agreement based Elliptic Curve Cryptography in smart grids. *Energy Rep.* **2022**, *8*, 153–164. [CrossRef]
33. Chen, C.; Guo, H.; Wu, Y.; Shen, B.; Ding, M.; Liu, J. A Lightweight Authentication and Key Agreement Protocol for IoT-Enabled Smart Grid System. *Sensors* **2023**, *23*, 3991. [CrossRef] [PubMed]
34. Abdi Nasib Far, H.; Bayat, M.; Kumar Das, A.; Fotouhi, M.; Pournaghi, S.M.; Doostari, M.-A. LAPTAS: Lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT. *Wirel. Netw.* **2021**, *27*, 1389–1412. [CrossRef]
35. Khan, A.A.; Kumar, V.; Ahmad, M.; Rana, S.; Mishra, D. PALK: Password-based anonymous lightweight key agreement framework for smart grid. *Int. J. Electr. Power Energy Syst.* **2020**, *121*, 106121. [CrossRef]
36. Chaudhry, S.A. Correcting "PALK: Password-based anonymous lightweight key agreement framework for smart grid". *Int. J. Electr. Power Energy Syst.* **2021**, *125*, 106529. [CrossRef]
37. Wazid, M.; Das, A.K.; Kumar, N.; Alazab, M. Designing Authenticated Key Management Scheme in 6G-Enabled Network in a Box Deployed for Industrial Applications. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7174–7184. [CrossRef]
38. Nyangaresi, V.O.; Abduljabbar, Z.A.; Abduljabbar, Z.A. Authentication and Key Agreement Protocol for Secure Traffic Signaling in 5G Networks. In Proceedings of the IEEE 2nd International Conference on Signal, Control and Communication (SCC), Hammamet, Tunisia, 20–22 December 2021; pp. 188–193. [CrossRef]
39. Esfahani, A.; Mantas, G.; Matischek, R.; Saghezchi, F.B.; Rodriguez, J.; Bicaku, A.; Maksuti, S.; Tauber, M.G.; Schmittner, C.; Bastos, J. A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment. *IEEE Internet Things J.* **2019**, *6*, 288–296. [CrossRef]
40. Nyangaresi, V.O.; Abood, E.W.; Abduljabbar, Z.A.; Al Sibahe, M.A. Energy Efficient WSN Sink-Cloud Server Authentication Protocol. In Proceedings of the 5th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 22–23 October 2021; pp. 1–6. [CrossRef]
41. Zhang, L.; Zhao, L.; Yin, S.; Chi, C.-H.; Liu, R.; Zhang, Y. A lightweight authentication scheme with privacy protection for smart grid communications. *Future Gener. Comput. Syst.* **2019**, *100*, 770–778. [CrossRef]
42. Ikeda, K. Long-range quantum energy teleportation and distribution on a hyperbolic quantum network. *IET Quantum Commun.* **2023**, 1–8. [CrossRef]
43. Ikeda, K.; Lowe, A. Quantum protocol for decision making and verifying truthfulness among N-quantum parties: Solution and extension of the quantum coin flipping game. *IET Quantum Commun.* **2023**, *4*, 218–227. [CrossRef]
44. Broadbent, A.; Fitzsimons, J.; Kashefi, E. Universal blind quantum computation. In Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science, Atlanta, GA, USA, 25–27 October 2009; pp. 517–526. [CrossRef]
45. Hiroka, T.; Morimae, T.; Nishimaki, R.; Yamakawa, T. Certified everlasting zero-knowledge proof for QMA. In *Annual International Cryptology Conference*; Springer Nature: Cham, Switzerland, 2022; pp. 239–268. [CrossRef]
46. Ikeda, K. Security and privacy of blockchain and quantum computation. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 2018; Volume 111, pp. 199–228. [CrossRef]
47. Park, K.; Lee, J.; Das, A.K.; Park, Y. BPPS: Blockchain-enabled privacy-preserving scheme for demand-response management in smart grid environments. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 1719–1729. [CrossRef]
48. Zhou, L.; Diro, A.; Saini, A.; Kaisar, S.; Hiep, P.C. Leveraging zero knowledge proofs for blockchain-based identity sharing: A survey of advancements, challenges and opportunities. *J. Inf. Secur. Appl.* **2024**, *80*, 103678. [CrossRef]

49. Crocetti, L.; Di Rienzo, R.; Verani, A.; Baronti, F.; Roncella, R.; Saletti, R. A novel and robust security approach for authentication, integrity, and confidentiality of Lithium-ion Battery Management Systems. In Proceedings of the 2023 IEEE 3rd International Conference on Industrial Electronics for Sustainable Energy Systems (IESES), Shanghai, China, 26–28 July 2023; pp. 1–6. [CrossRef]
50. Al Sibahee, M.A.; Nyangaresi, V.O.; Ma, J.; Abduljabbar, Z.A. Stochastic Security Ephemeral Generation Protocol for 5G Enabled Internet of Things. In *International Conference on Internet of Things as a Service*; Springer: Cham, Switzerland, 2022; pp. 3–18. [CrossRef]