applied sciences

MDPI

Article

Privacy-Preserving Lightweight Authentication Protocol for Demand Response Management in Smart Grid Environment

SungJin Yu [1], KiSung Park [2], JoonYoung Lee [1], YoungHo Park [1,*], YoHan Park [3,*], SangWoo Lee [2] and BoHeung Chung [2]

[1] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea; darkskiln@knu.ac.kr (S.Y.); harry250@naver.com (J.L.)
[2] Electronics and Telecommunications Research Institute, Daejeon 34129, Korea; ks.park@etri.re.kr (K.P.); ttomlee@etri.re.kr (S.L.); bhjung@etri.re.kr (B.C.)
[3] School of Computer Engineering, Keimyung University, Daegu 42601, Korea
* Correspondence: parkyh@knu.ac.kr (Y.P.); yhpark@kmu.ac.kr (Y.P.); Tel.: +82-53-950-7842 (Y.P.); +82-53-580-5229 (Y.P.)

Received: 24 January 2020; Accepted: 27 February 2020; Published: 4 March 2020

Abstract: With the development in wireless communication and low-power device, users can receive various useful services such as electric vehicle (EV) charging, smart building, and smart home services at anytime and anywhere in smart grid (SG) environments. The SG devices send demand of electricity to the remote control center and utility center (UC) to use energy services, and UCs handle it for distributing electricity efficiently. However, in SG environments, the transmitted messages are vulnerable to various attacks because information related to electricity is transmitted over an insecure channel. Thus, secure authentication and key agreement are essential to provide secure energy services for legitimate users. In 2019, Kumar et al. presented a secure authentication protocol for demand response management in the SG system. However, we demonstrate that their protocol is insecure against masquerade, the SG device stolen, and session key disclosure attacks and does not ensure secure mutual authentication. Thus, we propose a privacy-preserving lightweight authentication protocol for demand response management in the SG environments to address the security shortcomings of Kumar et al.'s protocol. The proposed protocol withstands various attacks and ensures secure mutual authentication and anonymity. We also evaluated the security features of the proposed scheme using informal security analysis and proved the session key security of proposed scheme using the ROR model. Furthermore, we showed that the proposed protocol achieves secure mutual authentication between the SG devices and the UC using Burrows–Abadi–Needham (BAN) logic analysis. We also demonstrated that our authentication protocol prevents man-in-the-middle and replay attacks utilizing AVISPA simulation tool and compared the performance analysis with other existing protocols. Therefore, the proposed scheme provides superior safety and efficiency other than existing related protocols and can be suitable for practical SG environments.

Keywords: smart grid; authentication; informal security analysis; BAN logic; ROR model; AVISPA

1. Introduction

In the past few years, with the advances of information and communication technologies, users can easily access any service provided in various smart grid (SG) environments, including smart home, smart building, vehicle-to-grid (V2G) and advanced metering infrastructure (AMI) [1–4]. In particular, smart grid using smart device has attracted growing attention from the academia, industries, and researchers. The SG device (sensing device, smart meter, etc.) is one of the core components, which collects various information (electricity consumption, payment, address, etc.) and transfers it to utility centers (power provider, power distributor, etc.) to provide secure, reliable, and efficient power distribution. According to the report of the U.S. Department of Energy (DoE), since 1988, electricity demand has risen by almost 30%. However, the transmission capacity of electricity has only increased by 15% [5]. Therefore, demand-response management has become an important issue to ensure reliable supply of electricity.

In SG environments, the SG devices are deployed in industries, smart buildings, smart homes, etc. and collect many data in real-time, transferring electricity demands to energy generators. However, energy generators cannot efficiently handle these demands because the data collected by SG devices is very large and is difficult to handle it. To address these problems and maintain the efficient stability of supply, utility centers (UCs) analyze the data collected by SG devices and control fault detection, dynamic pricing, load balancing, leakage power, and demand-response [6]. However, the data transmitted between the UC and the SG devices can be tampered, injected, deleted, and forged by a malicious adversary because they are transmitted over an insecure channel [7]. The result of these situations can generate energy imbalances and gaps between energy demand and response. Therefore, authentication and key agreement mechanisms have become essential security requirements for smooth functioning of the SG operations with respect to demand response and data analytics. The security requirements for the SG system are summarized as follows:

- Secure and efficient authentication and key agreement protocols are essential to ensure secure communication and privacy.
- The proposed authentication and key agreement protocol must withstand various attacks such as replay, masquerade, and off-line identity guessing attacks.
- Authentication and key agreement protocol should consider SG device limitations with respect to power consumption, communication bandwidth, and memory.

In general, for power consumption feedback purposes, a SG relies heavily on the usage of a smart metering infrastructure. For instance, the data of SG device is useful for load forecasting, demand response management, and real-time pricing. However, the recording and transmission of power consumption data may cause serious privacy issues. If fine-grained power consumption data of the SG device is exposed, it can reveal the private information of consumers related to their daily routines or the appliances in the house. In addition, the computation and communication resources at the consumer's side in the SG environments are usually very limited. Therefore, secure and efficient authentication mechanisms for preserving user privacy with low computational costs are essential in resource-constrained SG environments.

In 2019, Kumar et al. [6] proposed an elliptic curve cryptography (ECC)-based authentication protocol for demand response management in SG system. Kumar et al. claimed that their scheme can prevent various attacks. However, this paper shows that their scheme cannot withstand various attacks, including SG device stolen, session key disclosure, and masquerade attacks and cannot ensure secure mutual authentication. Furthermore, their scheme [6] is not suitable for resource-limited smart devices because it uses ECC with high computation and communication overheads. Therefore, we propose a privacy-preserving lightweight authentication scheme for demand response management in SG environments, considering an efficiency of SG devices and improving security level.

1.1. Adversary Model

We adopted the widely known Dolev–Yao (DY) threat model [8] to evaluate the safety of proposed protocol. According to the DY model, a malicious attacker can intercept, delete, modify, and insert the transmitted data over insecure channel. In addition to the capabilities of these attackers, the specific assumptions of the threat model are as follows:

- A malicious adversary can steal or obtain the SG device of a legal user and can extract secret parameters stored in the SG device utilizing power-analysis [9,10]. We also assume that a malicious adversary is able to capture as many SG device as possible.
- A malicious adversary may attempt various attacks, including masquerade, man-in-the-middle (MITM), session key disclosure, and replay attacks [11,12].
- Trusted authority (TA) and UCs are assumed to be fully trusted and semi-trusted entities, respectively, and cannot be compromised by a malicious adversary.

1.2. Contributions

The detailed contributions in this paper are summarized as follows:

- We demonstrate that Kumar et al.'s protocol cannot withstand various attacks such as masquerade, SG devices stolen, and session key disclosure attacks. We also show that their protocol does not ensure secure mutual authentication.
- We present a privacy-preserving lightweight authentication protocol for the SG system using pseudo-identity and secret parameter to enhance the security weaknesses of Kumar et al.'s protocol. The proposed protocol can withstand against masquerade, session key disclosure, replay, and MITM attacks, as well as achieve secure mutual authentication and anonymity. Thus, the proposed protocol is more secure and efficient than Kumar et al.'s protocol because it utilizes only hash and XOR operations.
- We performed the widely known Burrows–Abadi–Needham (BAN) logic analysis [13] to prove that the proposed scheme provides secure mutual authentication. We utilized informal security analysis to prove the safety of the proposed protocol against potential attacks and also proved the session key security of proposed scheme utilizing ROR model [14].
- We performed formal security analysis utilizing the widely adopted Automated Validation of Internet Security Protocols and Applications (AVISPA) tool to evaluate that the proposed scheme is secure against replay and MITM attacks. Moreover, we present the performance analysis of the proposed protocol with existing protocols.

1.3. Organization

The rest of the article is organized as follows. Section 2 presents related works that discuss the SG environments and then Section 3 presents system model for the SG environments. In Sections 4 and 5, we review of Kumar et al.'s scheme and analyze its security problems. In Section 6, we present a privacy-preserving lightweight authentication protocol for demand response management in SG environments to address the security shortcomings of Kumar et al.'s scheme and enhance efficiency. In Section 7, we perform the security analysis of the proposed scheme utilizing informal and formal analysis. Section 8 evaluates the security and performance features of the proposed scheme compared with existing schemes. Finally, we summarize the conclusion in Section 9.

2. Related Works

Many authentication and key agreement schemes for various environments have been presented over the last few years to ensure security and privacy of users [15–17]. In 2014, Rottondi et al. [15] presented the security and privacy scheme in V2G communication. In 2016, Jiang et al. [16] presented an ECC-based three-factor authentication scheme for e-health cloud to ensure privacy of health information. In 2016, Wan et al. [17] presented an efficient privacy-preserving scheme in the SG environments to provide secure communication and guarantee user's anonymity.

Recently, SG has attracted much attention from academia, research institutes, industry, and government [18,19]. In 2016, Tsai and Lo [20] presented identity-based encryption and signature key distribution protocol for the SG. However, in 2016, Odelu et al. [21] showed that Tsai and Lo's scheme [20] does not protect against ephemeral secret leakage attack and cannot ensure the privacy of smart meter. To resolve security drawbacks of Tsai and Lo's scheme, Odelu et al. [21] presented a secure authentication key agreement scheme for SG. In 2015, Doh et al. [22] proposed a secure authentication scheme between smart meter and the utility system to manage information of power consumption. In 2016, Saxena et al. [23] presented an authentication scheme for SG, which performs secure user authentication for SG to provide protection against various attacks. In 2016, He et al. [24] presented ECC based lightweight anonymous key distribution scheme for SG and it was more efficient than Tsai and Lo's scheme [20]. In 2017, Wazid et al. [25] presented secure three-factor remote user authentication scheme for renewable energy in SG system to enhance security level. In 2019, Kumar et al. [6] presented ECC-based authentication protocol for demand response management in SG system. However, as shown below, their scheme cannot prevent a variety of attacks such as SG device stolen, masquerade, and session key disclosure attacks, and it cannot ensure secure mutual authentication. Thus, we present a privacy-preserving lightweight authentication protocol for demand response management in the SG environments to address security problems of Kumar et al.'s scheme.

3. System Model

This section introduces the demand response management for the SG network model. This network model comprises two entities: a SG device and an UC, as shown in Figure 1. A SG device collects electricity data and provides efficient power management services. An UC manages monitoring data, including electricity consumption, load forecasting, demand response, real-time pricing, etc. The UC collects these data and estimates a total electricity capacity of a SG device in the power grid. However, as SG devices are deployed within the SG fields, the recording and transmission of power consumption data may cause serious privacy issues. A SG device usually sends sensitive power consumption reports via communication channel in the SG environments. A malicious adversary can intercept such reports to invade the privacy of users. For instance, it is easy to notice that inhabitants are at home or not by checking the power usage. In addition, privacy-sensitive data, such as usage of appliances, can be released to adversaries [26,27]. Consequently, privacy of users could be violated and sensitive data of users could be used for criminal purposes. Therefore, privacy-preserving authentication protocol in the SG environments should be supported.

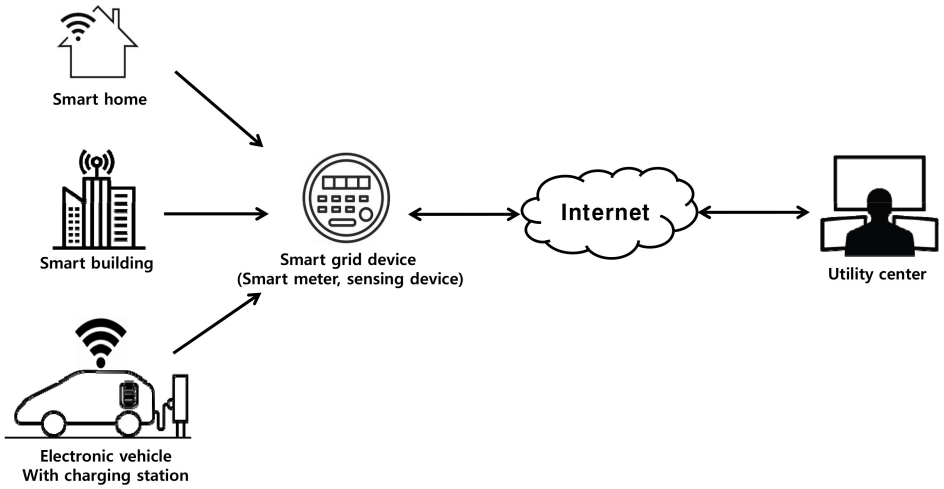

Figure 1. Network model for smart grid environments.

Figure 2 introduces the authentication process of the proposed scheme in the SG environments to provide user privacy, including daily routines and electricity consumption habits. The proposed scheme comprises three parties: trust authority (TA), SG device, and UC. The SG device and the UC first register their identities to TA, and then TA issues credential information for the SG device and the UC. After that, the SG device and the UC perform mutual authentication. After authentication, the SG device and the UC use the session key to exchange power consumption reports and feedbacks, and so on. Consequently, they can communicate safely through the secure channel established by the session key. The meaning of the communication session involves identifying devices in the network and authorizing what each device should carry out in the network. The maintenance of communication session in the proposed scheme may change monthly or yearly, depending on security requirements.

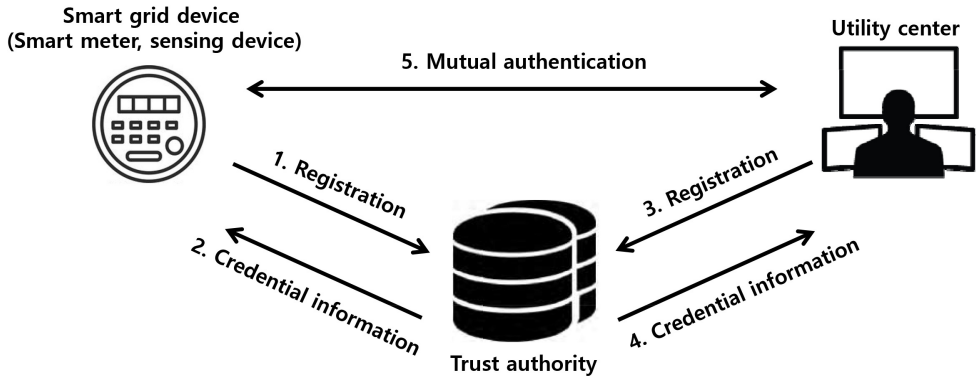

Figure 2. Authentication process of the proposed scheme in smart grid environments.

4. Review of Kumar et al.'s Protocol

This section reviews Kumar et al.'s authentication protocol for SG system. Kumar et al.'s scheme is comprised of five phases: SG device registration, UC registration, authentication, dynamic SG device addition, and dynamic UC additions. Table 1 summarizes the notation used in the protocol.

Table 1. Notations.

Notation	Description
TA	Trusted authority
SD_i	SG device
ID_i	SG's identity
RID_i	SG's pseudo-identity
UC_j	Utility center or remote control center
ID_j	UC's identity
TC_i	Temporal credential
T_i	Timestamp
$E_p(a,b)$	A nonsingular elliptic curve $y^2 = x^3 + ax + b \pmod p$
G	A base point for elliptic curve
$k.G$	An elliptic curve point multiplication
U_i, V_j	The public key for SD_i and UC_j
x	TA's secret key
K_s	TA's master key
SK_{ij}	Session key
$h(\cdot)$	Hash function
$\oplus$	XOR operation
$\|\|$	Concatenation operation

4.1. Smart Grid Device Registration Process

The SG device is called $SD_i (i = 1, 2, \ldots n)$, where n is the number of UC to be deployed initially in SG system. The SD_i must register with TA to receive any services, where n is the number of the SG devices. A trusted authority TA chooses a ID_i and calculates $RID_i = h(ID_i||x)$ and $TC_i = h(x||RTS_i)$, where RTS_i is the registration timestamp of the SG device. After that, the TA pre-loads the data $\{TC_i, RID_i, h(), E_p(a,b), G\}$ into memory before deployment in SG system. Figure 3 describes the SG device registration process of Kumar et al.'s protocol.

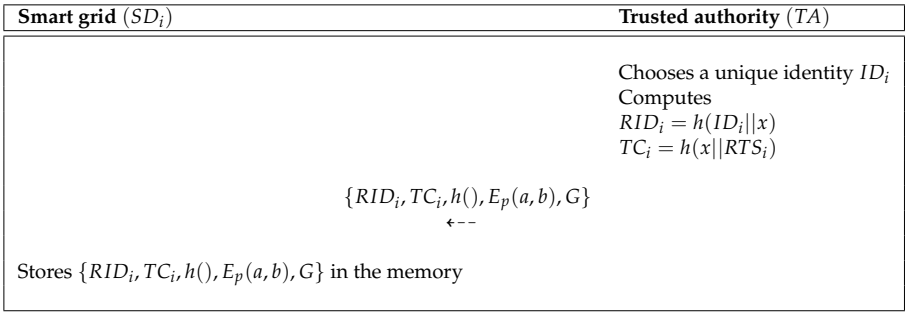

Figure 3. Smart grid device registration process of Kumar et al.'s scheme.

4.2. Utility Center Registration Process

The utility center UC_j must register with TA to deploy the SG environments. The UC_j is called $UC_j (j = 1, 2, \ldots, k)$, where k is the number of UC to be deployed initially in SG system. TA chooses an identity ID_j and calculates $RID_j = h(ID_j||x)$ and $TC_j = h(x||RTS_j)$, where RTS_j is the registration timestamp of the UC. Finally, the TA pre-loads the data $\{RID_j, TC_j, h(), E_p(a,b), G, RID_i | i = 1, 2, \ldots, n\}$ into memory before deployment in SG system. Figure 4 describes the UC registration process of Kumar et al.'s protocol.

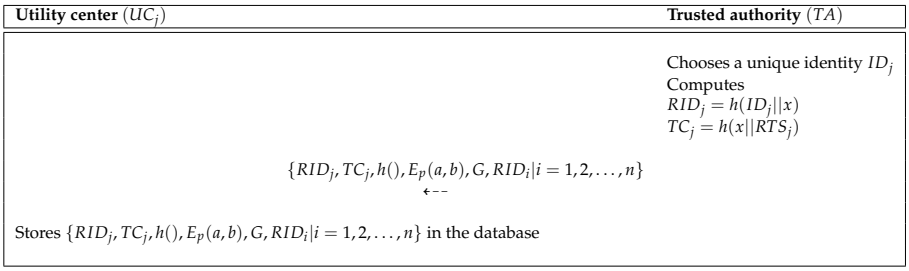

Figure 4. Utility center registration process of Kumar et al.'s scheme.

4.3. Authentication Process

The main goal of this process is to negotiate a session key between SD_i and UC_j. Therefore, the SD_i and UC_j must authenticate each other. Figure 5 describes the authentication process of Kumar et al.'s protocol. The detailed process is described below.

Step 1: SD_i chooses a random number $u \in Z_p^*$ and generates a current timestamp T_1. After that, SD_i computes $U_i = u.G$ and $C_i = h(TC_i||T_1) \oplus h(RID_i||U_i||T_1)$ and sends authentication request message $\{U_i, C_i, T_1\}$ to the UC_j over insecure channel.

Step 2: After receiving the message, UC_j checks $|T_1 - T_1^*| \leq \Delta T$, where ΔT is maximum transmission delay bound and T_1 is current timestamp. If the condition is valid, UC_j computes $D_j = C_i \oplus h(RID_i||U_i||T_1)$ utilizing the corresponding RID_i of SD_i stored in the database.

Step 3: UC_j then generates timestamp T_2 and a random number $v \in Z_p^*$, and calculates $V_j = v.G$, $W_j = v.U_i = (uv).G$, the session key shared with SD_i as $SK_{ij} = h(W_j||D_j||h(RID_j||TC_j||T_2))$, $SKV_{ij} = h(SK_{ij}||RID_i||T_2)$ and $Z_j = h(RID_j||TC_j||T_2) \oplus h(RID_i||U_i||V_j||T_2)$. After that, UC_j sends the authentication message $\{V_j, Z_j, SKV_{ij}, T_2\}$ to the SD_i over insecure channel.

Step 4: After receiving the message, SD_i checks condition $|T_2 - T_2^*| \leq \Delta T$. If it is correct, SD_i further calculates $E_i = Z_j \oplus h(RID_i||U_i||V_j||T_2) = h(RID_j||TC_j||T_2)$, $W_i' = u.V_j = (uv).G$, and session key shared with UC_j as $SK_{ij}' = h(W_i'||h(TC_i||T_1||E_i))(= SK_{ij})$, $SKV_{ij}' = h(SK_{ij}'||RID_i||T_2)$. If the condition $SKV_{ij}' \neq SKV_{ij}$, SD_i aborts communication. Otherwise, SD_i generates a timestamp T_3 and calculates $SKV_{ij}^* = h(SK_{ij}'||RID_i||V_j||T_3)$. After that, SC_i sends acknowledgment message $\{SKV_{ij}^*, T_3\}$ to the UC_j over insecure channel.

Step 5: After receiving the message, UC_j checks the condition $|T_3 - T_3^*| \leq \Delta T$. If the condition is valid, UC_j computes $SKV_{ij}^{**} = h(SK_{ij}||RID_i||V_j||T_3)$ and checks if $SKV_{ij}^{**} = SKV_{ij}^*$ holds. If the condition is valid, SD_i and UC_j store the common session key $SK_{ij}(= SK_{ij}')$.

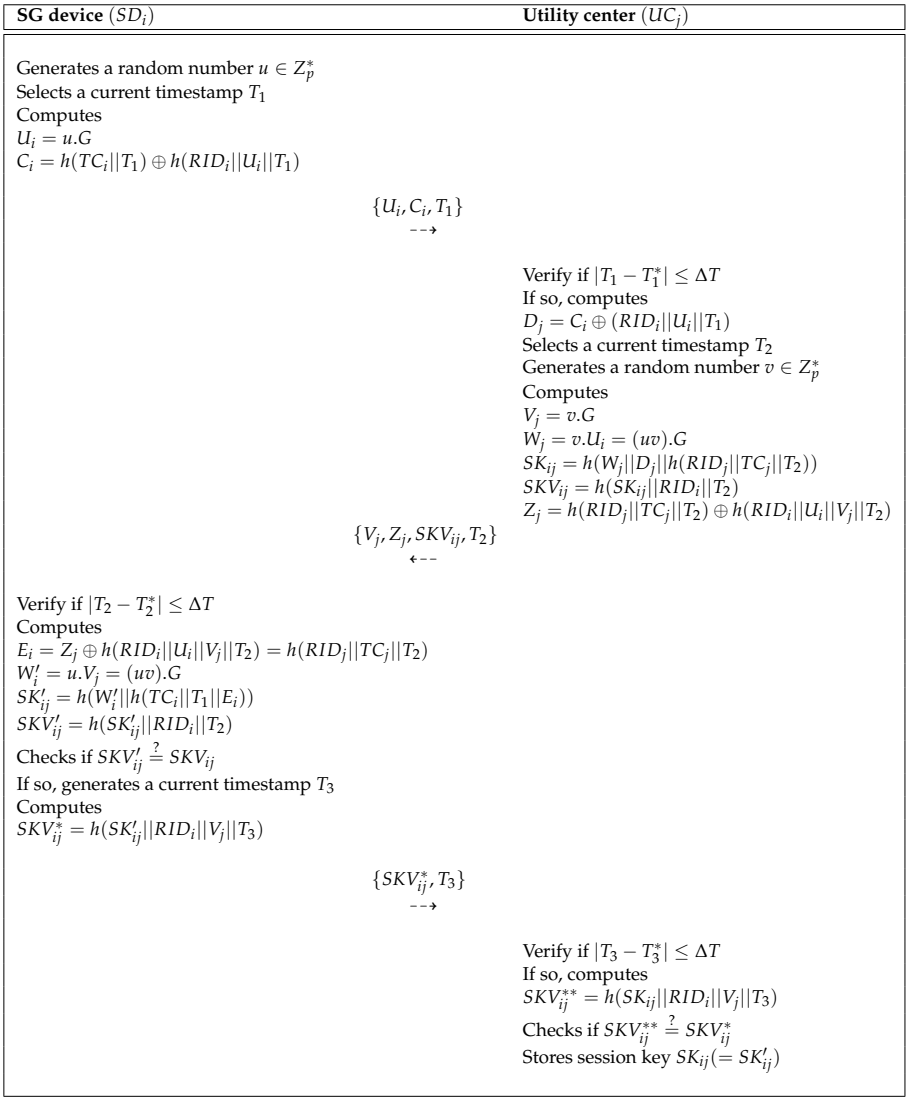

Figure 5. Authentication process of Kumar et al.'s scheme.

4.4. Dynamic Smart Grid Device Addition Process

The main goal of this process is adding a new SG device SD_i^{new} to provide flexibility in the system and the detailed processes are shown below.

Step 1: Trusted authority (TA) selects an identity ID_i^{new} and calculates $RID_i^{new} = h(ID_i^{new}||x)$ and $TC_i^{new} = h(x||RTS_i^{new})$.

Step 2: After that, the TA pre-loads the data $\{RID_i^{new}, TC_i^{new}, h(), E_p(a,b), G\}$ in the memory before it is deployed.

Step 3: TA sends data RID_i^{new} for SD_i^{new} to all UC_j over secure channel. The TA needs to broadcast messages to the deployed UC_j regarding deployment of the SD_i^{new} so that SD_i^{new} and deployed UC_j can establish a common session key after mutual authentication.

4.5. Dynamic Utility Center Addition Process

The main goal of this process is same as the one in Section 4.4 from the point of view of UC and the detailed processes are shown below.

Step 1: The TA selects a identity ID_j^{new} and calculates $RID_j^{new} = h(ID_j^{new}||x)$ and $TC_j^{new} = h(x||RTS_j^{new})$.
Step 2: TA then pre-loads the data $\{RID_j^{new}, TC_j^{new}, RID_i|i = 1, 2, \ldots, n, h(), E_p(a, b), G\}$ in the memory before it is deployed.
Step 3: If a SD_i^{new} is already deployment prior to UC_j^{new}, TA pre-loads RID_i^{new} into the memory of UC_j^{new}.

After finishing this process, TA broadcasts a completion statement to all entities and UC_j^{new} is successfully registered in SG environments.

5. Cryptanalysis of Kumar et al.'s Protocol

This section demonstrates the security drawbacks of Kumar et al.'s protocol, including SG device stolen, masquerade, and session key disclosure attacks, as well as mutual authentication.

5.1. Masquerade Attack

We assume that a malicious adversary U_{ma} can obtain the SG device of legal user SD_i and intercept information transmitted in open channel, and then may attempt to masquerade SD_i. According to Section 1.1, U_{ma} can extract secret information $\{RID_i, TC_i, h(), E_p(a, b), G\}$ using power analysis attack. Finally, U_{ma} performs the masquerade attack as below:

Step 1: U_{ma} generates a random number $u_{ma} \in Z_p^*$ and calculates $U_{ima} = u_{ma}.G$, $C_{ma} = h(TC_i||T_1) \oplus h(RID_i||U_{ima}||T_1)$. After that, U_{ma} sends message $\{U_{ima}, C_{ma}, T_1\}$ to UC_j over insecure channel.
Step 2: After receiving the message from U_{ma}, UC_j checks $|T_1 - T_1^*| \leq \Delta T$. If the condition is valid, UC_j calculates $D_j = C_{ma} \oplus h(RID_i||U_{ima}||T_1)$ and generates a timestamp T_2. UC_j then selects a random number $v \in Z_p^*$ and computes $V_j = v.G$, $W_{ma} = v.U_{ima} = (u_{ma}v).G$, $SK_{ma} = h(W_{ma}||D_j||h(RID_j||TC_j||T_2)$, $SKV_{ma} = h(SK_{ma}||RID_i||T_2)$, and $Z_{ma} = h(RID_j||TC_j||T_2) \oplus h(RID_i||U_{ima}||V_j||T_2)$. After that, UC_j sends the message $\{V_j, Z_{ma}, SKV_{ma}, T_2\}$ to U_{ma}.
Step 3: After receiving the message from UC_j, U_{ma} checks condition $|T_2 - T_2^*| \leq \Delta T$. If the condition is valid, U_{ma} computes $E_i = Z_j \oplus h(RID_i||U_{ma}||V_j||T_2) = h(RID_j||TC_j||T_2)$, $W_i' = u_{ma}.V_j = (u_{ma}v).G$, and $SK_{ma}' = h(W_{ma}'||h(TC_i||T_1)||E_i)$. Then, U_{ma} generates a timestamp T_{3ma} and computes $SKV_{ma}' = h(SK_{ma}'||RID_i||T_2)$. After that, U_{ma} sends message $\{SKV_{ma}^*, T_3\}$ to UC_j over insecure channel.
Step 4: After receiving the message from U_{ma}, UC_j checks condition $|T_{3ma} - T_{3ma}^*| \leq \Delta T$. If the condition is valid, UC_j calculates $SKV_{ma}^{*'} = h(SK_{ma}||RID_i||V_j||T_{3ma})$ and checks if $SKV_{ma}^{*'} = SKV_{ma}^*$ holds. If the condition is valid, U_{ma} and UC_j store session key $SK_{ma}(= SK_{ma}')$.

Therefore, U_{ma} can successfully generate a session key between U_{ma} and UC_j and send a legitimate authentication request message. Consequently, we show that Kumar et al.'s protocol does not withstand masquerade attack.

5.2. Smart Grid Device Stolen Attack

Kumar et al. claimed that their scheme could withstand SG device stolen attack because a malicious attacker U_{ma} cannot calculate the correct $RID_i = h(ID_i||x)$ and $TC_i = h(x||RTS_i)$ without knowing secret key x of the TA. However, according to Section 5.1, we demonstrate that U_{ma} successfully impersonates legitimate user and calculates the session key. Therefore, Kumar et al.'s protocol is insecure against SG device stolen attack.

5.3. Session Key Disclosure Attack

In Kumar et al.'s scheme, they claimed that their scheme was secure against session key disclosure attack, although the secret numbers u and v are compromised to U_{ma}. According to the Kumar et al's scheme, U_{ma} cannot obtain session key SK_{ij} because U_{ma} does not know parameters RID_i and TC_i. However, in Section 5.1, we demonstrate that U_{ma} can successfully generate session key SK_{ij} using parameters obtained from SG devices of a legitimate user. Therefore, once a SG device is compromised, all its previous communications will be breached. Furthermore, since the malicious attacker U_{ma} can capture as many SG devices as possible, the U_{ma} can obtain the session key SK_{ij} of other SG devices. As a result, Kumar et al.'s protocol cannot defend against session key disclosure attack.

5.4. Mutual Authentication

Kumar et al. showed that their scheme could achieve secure mutual authentication between SD_i and UC_j. However, according to Section 5.1, U_{ma} can successfully compute authentication request message $C_i = h(TC_i||T_1) \oplus h(RID_i||U_i||T_1)$ and response message $SKV_{ij}^* = h(SK_{ij}'||RID_i||V_j||T_3)$. Consequently, Kumar et al.'s scheme does not achieve secure mutual authentication.

6. Proposed Scheme

This section proposes a privacy-preserving lightweight authentication scheme for demand response management in the SG environment to overcome various security drawbacks of Kumar et al.'s protocol [6]. In our scheme, the general data flow of the SG system model in public channel is the same as Kumar et al.'s scheme [6]. The proposed scheme is composed of seven process: pre-deployment, SG registration, UC registration, authentication, dynamic SG device addition, and dynamic UC addition.

6.1. Pre-Deployment Process

In this section, the SG devices SD_i and UC_j must register with TA before its deployment in SG environments. TA firstly selects unique identities ID_i and ID_j of SD_i and UC_j, respectively. Then, TA stores the credential information $\{ID_i\}$ in the memory of SD_i and stores the credential information $\{ID_j\}$ in the database of UC_j prior to its deployment in the SG environments.

6.2. Smart Grid Device Registration Process

The SD_i must register with trusted authority TA to receive the power management services. Figure 6 describes the SG device registration process of proposed scheme and the steps of this process are given below.

Step 1: TA generates a random number x_i, a_i for SD_i. After that, TA computes $RID_i = h(ID_i||a_i)$, $X_i = h(RID_i||K_s||x_i)$, $A_i = X_i \oplus h(RID_i||a_i)$, and $B_i = h(RID_i||X_i)$ and stores $\{x_i, RID_i\}$ in secure database. Finally, TA sends $\{A_i, B_i, a_i\}$ to SD_i.

Step 2: After receiving the message, SD_i computes $C_i = h(ID_i||B_i) \oplus a_i$ and stores $\{A_i, B_i, C_i\}$ in the memory.

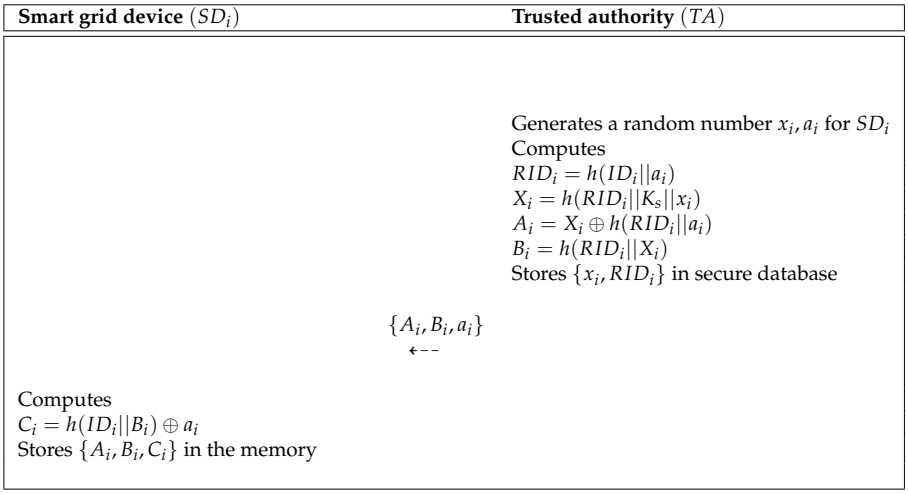

Figure 6. Smart grid device registration process of the proposed scheme.

6.3. Utility Center Registration Process

The UC_j must register with TA in order to provide power management services. Figure 7 describes the UC registration process of proposed scheme and the steps of this process are given below.

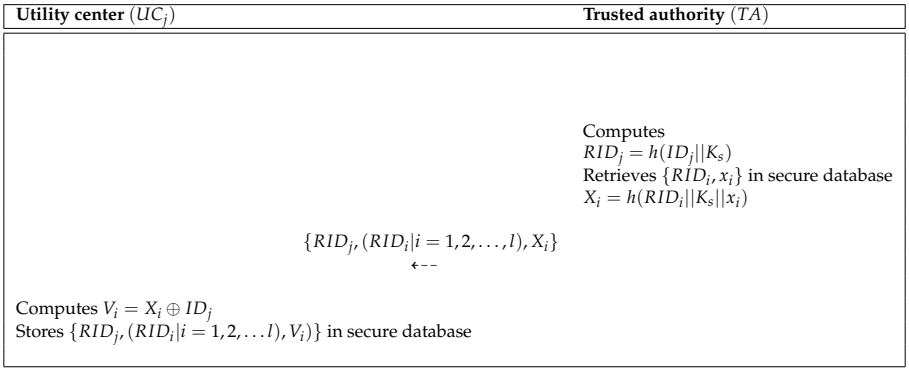

Figure 7. Utility center registration process of the proposed scheme.

Step 1: TA computes $RID_j = h(ID_j||K_s)$ and retrieves $\{RID_i, x_i\}$ in secure database. Then, TA computes $X_i = h(RID_i||K_s||x_i)$ and sends $\{RID_j, (RID_i|i = 1, 2 \ldots, l), X_i\}$ to UC_j.

Step 2: After receiving the message, UC_j computes $V_i = X_i \oplus ID_j$ and stores $\{RID_j, (RID_i|i = 1, 2 \ldots, l), V_i\}$ in the database.

6.4. Authentication Process

In authentication process, the proposed scheme provides the user's privacy by using pseudo-identity and secret parameters in the SG environments. Before the starting session, SD_i request an authentication request to UC_j in order to ensure secure communication and establish the session key SK_{ij}. Figure 8 describes the authentication process of proposed scheme and the steps of this process are given below.

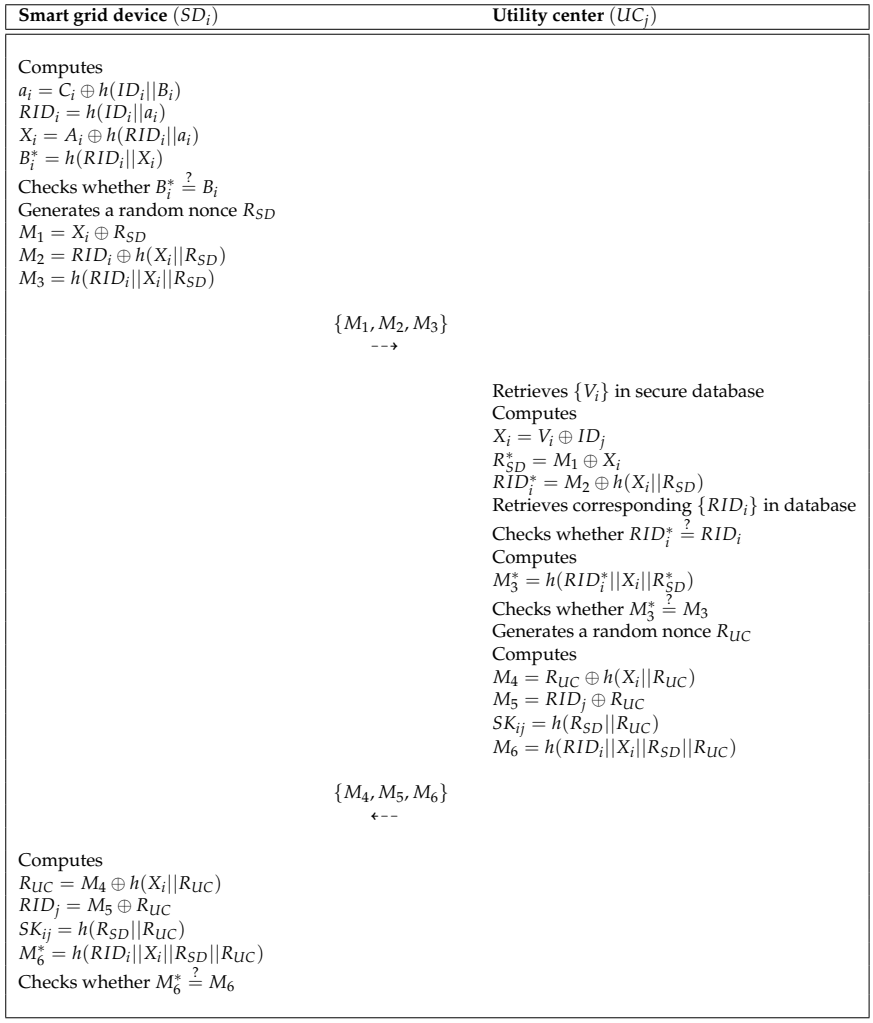

Figure 8. Authentication process of the proposed scheme.

Step 1: SD_i computes $a_i = C_i \oplus h(ID_i||B_i)$, $RID_i = h(ID_i||a_i)$, $X_i = A_i \oplus h(RID_i||a_i)$, and $B_i^* = h(RID_i||X_i)$. Then, SD_i checks whether $B_i^* \stackrel{?}{=} B_i$. If the condition $B_i^* \stackrel{?}{=} B_i$ is valid, SD_i generates a random nonce R_{SD} and computes $M_1 = X_i \oplus R_{SD}$, $M_2 = RID_i \oplus h(X_i||R_{SD})$, and $M_3 = h(RID_i||X_i||R_{SD})$. After that, SD_i sends authentication request message $\{M_1, M_2, M_3\}$ to UC_j over insecure channel.

Step 2: After receiving the message from SD_i, UC_j retrieves $\{V_i\}$ in database and calculates $X_i = V_i \oplus ID_j$, $R_{SD}^* = M_1 \oplus X_i$, and $RID_i^* = M_2 \oplus h(X_i||R_{SD})$. Then, UC_j retrieves corresponding $\{RID_i\}$ in database and checks whether $RID_i^* \stackrel{?}{=} RID_i$. If the condition $RID_i^* \stackrel{?}{=} RID_i$ is valid, UC_j calculates $M_3^* = h(RID_i^*||X_i||R_{SD}^*)$ and checks whether $M_3^* \stackrel{?}{=} M_3$. If the condition $M_3^* \stackrel{?}{=} M_3$ is correct, UC_j generates a random nonce R_{UC} and computes $M_4 = R_{UC} \oplus h(X_i||R_{UC})$, $M_5 = RID_j \oplus R_{UC}$, $SK_{ij} = h(R_{SD}||R_{UC})$ and $M_6 = h(RID_i||X_i||R_{SD}||R_{UC})$. Finally, UC_j sends authentication message $\{M_4, M_5, M_6\}$ to SD_i over insecure channel.

Step 3: After receiving the message from UC_j, SD_i computes $R_{UC} = M_4 \oplus h(X_i||R_{UC})$, $RID_j = M_5 \oplus R_{UC}$, $SK_{ij} = h(R_{SD}||R_{UC})$, and $M_6^* = h(RID_i||X_i||R_{SD}||R_{UC})$. After that,

SD_i checks whether $M_6^* \stackrel{?}{=} M_6$. If the condition $M_6^* \stackrel{?}{=} M_6$ is correct, the SD_i and UC_j achieve mutual authentication successfully.

6.5. Dynamic Smart Grid Device Addition Process

When new SG device SD_i wants to register with the SG environments, the following steps must be performed and detailed steps are as follows. The main goal of this process is adding a new SG device to provide flexibility in SG environments. The detailed steps of this process are given below.

Step 1: First, TA chooses a new ID_i^{new} to the SD_i over secure channel. After receiving the message, SD_i sends RID_i to the TA over secure channel. Then, TA generates a random number a_i^{new}, x_i^{new}.

Step 2: After that, TA computes $RID_i^{new} = h(ID_i^{new}||a_i^{new})$, $X_i^{new} = h(RID_i^{new}||K_s||x_i^{new})$, $A_i^{new} = X_i^{new} \oplus h(RID_i^{new}||a_i^{new})$, and $B_i^{new} = h(RID_i^{new}||X_i^{new})$. Finally, TA stores $\{x_i^{new}, RID_i^{new}\}$ in secure database and sends its to the SD_i over secure channel.

Step 3: After receiving the message, SD_i computes $C_i^{new} = h(ID_i^{new}||B_i) \oplus a_i^{new}$ and stores $\{A_i^{new}, B_i^{new}\}$ in the memory.

6.6. Dynamic Utility Center Addition Process

The following steps are required to deploy new UC_j^{new} and the detailed steps are given below.

Step 1: The TA chooses a new ID_j^{new} and sends $\{ID_j^{new}\}$ to UC_j over secure channel. After receiving the message, UC_j sends RID_j to the TA over secure channel. After that, TA computes $RID_j^{new} = h(ID_j^{new}||K_s)$ and retrieves $\{RID_i^{new}, x_i\}$ in the database.

Step 2: Then, TA computes $X_i^{new} = h(RID_i^{new}||K_s||x_i^{new})$ and sends $\{RID_j^{new}, (RID_i^{new}|i = 1, 2, \ldots, l), X_i^{new}\}$ to the UC_j.

Step 3: After receiving the message, UC_j computes $V_i^{new} = X_i^{new} \oplus ID_j^{new}$ and stores $\{RID_j^{new}, (RID_i^{new}|i = 1, 2, \ldots, l), V_i^{new}\}$ in secure database.

7. Security Analysis

In this phase, we demonstrate that the proposed scheme has the ability to resist various attacks using informal security analysis and the formal security verification tool Automated Validation of Internet Security Protocols and Applications (AVISAP). We also analyze that our proposed scheme provides session key security and secure mutual authentication using Real-or-Random (ROR) model [14] and Burrows–Abadi–Needham (BAN) logic [13]. ROR model, BAN logic, and AVISPA analysis techniques are also widely accepted to evaluate the security of protocol.

7.1. Informal Security Analysis

We performed informal security analysis to demonstrate the safety of the proposed scheme. Our protocol can defend against various attacks such as session key disclosure, SG device stolen, masquerade, and replay attacks, as well as ensure secure mutual authentication and anonymity.

7.1.1. Masquerade Attack

According to Section 1.1, a malicious adversary U_{ma} can obtain SG device of legitimate user and can intercept transmitted data over insecure channel. If U_{ma} tries impersonate a legitimate user, U_{ma} must correctly generate an authentication request and response messages. However, U_{ma} cannot generate the authentication request message $\{M_1, M_2, M_3\}$ and authentication message $\{M_4, M_5, M_6\}$ without the correct random nonces R_{SD} and R_{UC}. Furthermore, U_{ma} cannot generate a session key $SK_{ij} = h(R_{SD}||R_{UC})$

because secret parameter X_i is not available to U_{ma}. Therefore, the proposed scheme is secure against masquerade attack.

7.1.2. Smart Grid Device Stolen Attack

We assume that a malicious adversary U_{ma} obtains SG device of a legitimate user and extracts secret information $\{A_i, B_i, C_i\}$ stored in the memory using power analysis attack [9]. However, U_{ma} cannot obtain sensitive information of a legitimate user because all information stored in the memory is masked by XOR operation and hash function. Therefore, our protocol prevents SG device stolen attack because U_{ma} cannot know the user's real identity ID_i, a_i, and secret parameter X_i.

7.1.3. Replay Attack

Our protocol withstands replay attack because all transmitted messages are changed in every session. Assuming that U_{ma} tries to impersonate legal user by resending information transmitted in a previous authentication process, U_{ma} cannot use the previous messages because SD_i and UC_j check whether $M_3^* \stackrel{?}{=} M_3$ and $M_6^* \stackrel{?}{=} M_6$, respectively. Thus, our protocol is secure against replay attack.

7.1.4. Session key disclosure attack

In the proposed scheme, U_{ma} cannot calculate $SK_{ij} = h(R_{SD}||R_{UC})$ because U_{ma} cannot compute authentication request message $\{M_1, M_2, M_3\}$ without knowing random nonce R_{SD} and secret parameter X_i. Therefore, our protocol can withstand session key disclosure attack.

7.1.5. Insider attack

This type of attack happens when the administrator of authentication server exploits data stored in the database to legalize his authentication process on behalf of the user. Even if it is assumed that a malicious adversary U_{ma} can obtain RID_i, RID_j, V_i stored in memory of UC_j, U_{ma} cannot obtain sensitive information such as user's real identity ID_i and X_i without knowing random nonce R_{SD} and ID_j. Thus, our protocol is secure against insider attack.

7.1.6. Mutual Authentication

After receiving the authentication request message $\{M_1, M_2, M_3\}$ from the SD_i, UC_j checks whether $M_3^* \stackrel{?}{=} M_3$. If $M_3^* \stackrel{?}{=} M_3$ is valid, UC_j authenticates SD_i successfully. After receiving the authentication message $\{M_4, M_5, M_6\}$ from the UC_j, SD_i also checks whether $M_6^* \stackrel{?}{=} M_6$, and then SD_i authenticates UC_j. Therefore, our protocol ensures secure mutual authentication between SD_i and UC_j because U_{ma} cannot generate correct authentication messages.

7.1.7. Anonymity

U_{ma} does not obtain a legitimate user's real identity ID_i because it is masked by one-way hash function and XOR operation such as $RID_i = h(ID_i||a_i)$. Therefore, our protocol ensures anonymity because U_{ma} cannot know the user's real identity without random nonce a_i and R_{SD}.

7.2. Security Features

In Table 2, we evaluate the security features of the proposed scheme with existing schemes [6,20,21,28]. The schemes in [20,28] cannot withstand session key disclosure attack and those in [20,21,28] provide dynamic node addition phase. The scheme in [6] cannot withstand various types of attacks and cannot

ensure secure mutual authentication and anonymity. Consequently, the proposed scheme ensures better security functionality than all previous schemes.

Table 2. A comparative summary: security features.

Security Feature	Wu–Zhou [28]	Tsai–Lo [20]	Odelu et al. [21]	Kumar et al. [6]	Ours
Masquerade attack	○	○	○	×	○
Smart grid device stolen attack	○	○	○	×	○
Replay attack	○	○	○	○	○
Session key disclosure attack	×	×	○	×	○
Man-in-the-middle attack	○	○	○	○	○
Mutual authentication	○	○	○	×	○
Anonymity	×	○	○	○	○
Dynamic node addition phase	×	×	×	○	○

○: security feature is satisfied; ×: security feature is not satisfied.

7.3. Formal Security Analysis Using BAN Logic

We performed BAN logic [13] analysis to verify that our protocol provides secure mutual authentication. Table 3 shows the notation used for BAN logic analysis and we then defines the goals, idealized forms, and assumptions before performing BAN logic analysis.

Table 3. Notations used for BAN logic.

Notation	Description
$Q\mid\equiv M$	Q **believes** statement M
$\#M$	Statement M is **fresh**
$Q \lhd M$	Q **sees** statement M
$Q\mid\sim M$	Q once **said** M
$Q \Rightarrow M$	Q **controls** statement M
$< M >_N$	Formula M is **combined** with formula N
$\{M\}_K$	Formula M is **encrypted** by key K
SK	**Session key** used in the current authentication session
$Q \overset{K}{\leftrightarrow} W$	Q and W communicate utilizing K as the **shared key**

7.3.1. BAN Logic Rule

The rules of BAN logic are as follows.

- Message meaning rule:

$$\frac{Q \mid\equiv Q \overset{K}{\leftrightarrow} W, \quad Q \lhd \{M\}_K}{Q \mid\equiv W \mid\sim M}$$

- Nonce verification rule:

$$\frac{Q \mid\equiv \#(M), \quad Q \mid\equiv W \mid\sim M}{Q \mid\equiv W \mid\equiv M}$$

- Jurisdiction rule:

$$\frac{Q \mid\equiv W \mid\Longrightarrow M, \quad Q \mid\equiv W \mid\equiv M}{Q \mid\equiv M}$$

- Freshness rule:

$$\frac{Q \mid \equiv \#(M)}{Q \mid \equiv \#(M, N)}$$

- Belief rule:

$$\frac{Q \mid \equiv (M, N)}{Q \mid \equiv M}$$

7.3.2. Goals

The goals for BAN logic analysis are as follows.

Goal 1: $UC_j \mid\equiv (UC_j \overset{SK}{\longleftrightarrow} SD_i)$
Goal 2: $UC_j \mid\equiv SD_i \mid\equiv (UC_j \overset{SK}{\longleftrightarrow} SD_i)$
Goal 3: $SD_i \mid\equiv (UC_j \overset{SK}{\longleftrightarrow} SD_i)$
Goal 4: $SD_i \mid\equiv UC_j \mid\equiv (UC_j \overset{SK}{\longleftrightarrow} SD_i)$

7.3.3. Idealized Forms

The idealized forms are formulated as follows:

$\boldsymbol{Msg_1}$: $SD_i \rightarrow UC_j$: $(RID_i, R_{SD})_{X_i}$
$\boldsymbol{Msg_2}$: $UC_j \rightarrow SD_i$: $(RID_i, RID_j, R_{UC})_{X_i}$

7.3.4. Assumptions

We define initial assumptions to perform the BAN logic analysis.

$\boldsymbol{A_1}$: $UC_j \mid\equiv \#(R_{SD})$
$\boldsymbol{A_2}$: $SD_i \mid\equiv \#(R_{UC})$
$\boldsymbol{A_3}$: $UC_j \mid\equiv (UC_j \overset{X_i}{\longleftrightarrow} SD_i)$
$\boldsymbol{A_4}$: $SD_i \mid\equiv (UC_j \overset{X_i}{\longleftrightarrow} SD_i)$
$\boldsymbol{A_5}$: $UC_j \mid\equiv SD_i \Rightarrow (R_{SD})$
$\boldsymbol{A_6}$: $SD_i \mid\equiv UC_j \Rightarrow (R_{UC})$
$\boldsymbol{A_7}$: $UC_j \mid\equiv SD_i \Rightarrow (UC_j \overset{SK}{\longleftrightarrow} SD_j)$
$\boldsymbol{A_8}$: $SD_i \mid\equiv UC_j \Rightarrow (UC_j \overset{SK}{\longleftrightarrow} SD_j)$

7.3.5. Proof Using BAN Logic

We performed the BAN logic analysis for our protocol and the detailed proofs are below.

Step 1: According to Msg_1, we obtain

$$S_1 : UC_j \lhd (RID_i, R_{SD})_{X_i}$$

Step 2: Using the message meaning rule with S_1 and A_3, we can obtain

$$S_2 : UC_j \mid\equiv SD_i \mid\sim (RID_i, R_{SD})_{X_i}$$

Step 3: Using the freshness rule with A_1, we can obtain

$$S_3 : UC_j \mid\equiv SD_i \mid\equiv \#(RID_i, R_{SD})_{X_i}$$

Step 4: From the nonce verification rule with S_2 and S_3, we can obtain

$$S_4 : UC_j \mid\equiv SD_i \mid\equiv (RID_i, R_{SD})_{X_i}$$

Step 5: Using the belief rule with S_4, we can obtain

$$S_5 : UC_j \mid\equiv SD_i \mid\equiv (R_{SD})$$

Step 6: Because of $SK = h(R_{SD}||R_{UC})$, from the S_5 and A_2 we can obtain

$$S_6 : UC_j \mid\equiv SD_i \mid\equiv (UC_j \overset{SK}{\longleftrightarrow} SD_i) \quad \textbf{(Goal 2)}$$

Step 7: From the jurisdiction rule with S_6 and A_7 we can obtain

$$S_7 : UC_j \mid\equiv (UC_j \overset{SK}{\longleftrightarrow} SD_i) \quad \textbf{(Goal 1)}$$

Step 8: According to Msg_2, we can obtain

$$S_8 : SD_i \lhd (RID_i, RID_j, R_{UC})_{X_i}$$

Step 9: Using the message meaning rule with S_8 and A_4, we can obtain

$$S_8 : SD_i \mid\equiv UC_j \mid\sim (RID_i, RID_j, R_{UC})_{X_i}$$

Step 10: Using the freshness rule with A_2, we can obtain

$$S_{10} : SD_i \mid\equiv UC_j \mid\equiv \#(RID_i, RID_j, R_{UC})_{X_i}$$

Step 11: Using the nonce verification rule with S_9 and S_{10}, we can obtain

$$S_{11} : SD_i \mid\equiv UC_j \mid\equiv (RID_i, RID_j, R_{UC})_{X_i}$$

Step 12: Using the belief rule with S_{11}, we can obtain

$$S_{12} : SD_i \mid\equiv UC_j \mid\equiv (R_{UC})$$

Step 13: Because of $SK = h(R_{SD}||R_{UC})$, from the S_{12} and A_1 we can obtain

$$S_{13} : SD_i \mid\equiv UC_j \mid\equiv (UC_j \overset{SK}{\longleftrightarrow} SD_i) \quad \textbf{(Goal 4)}$$

Step 14: Using the jurisdiction rule with S_{13} and A_8 we can obtain

$$S_7 : SD_i \mid\equiv (UC_j \overset{SK}{\longleftrightarrow} SD_i) \quad \textbf{(Goal 3)}$$

Based on Goals 1–4, we proved that proposed protocol ensures secure mutual authentication between SD_i and UC_j.

7.4. Formal Security Analysis Using ROR Model

ROR model [14] is the formal security analysis to verify session key (SK) security of protocol from active/passive attacker U_A. We first discuss the ROR model before performing the proof of SK security for the proposed protocol.

In our protocol, there are two participants SG device $P_{SD_i}^{t_1}$ and UC $P_{UC_j}^{t_2}$, where $P_{SD_i}^{t_1}$ and $P_{UC_j}^{t_2}$ are instances t_1^{th} of SD_i and t_2^{th} of UC_j, respectively. Table 4 defines queries for ROR model to perform security analysis, including *Execute*, *CorruptSD*, *Reveal*, *Send*, and *Test* queries. *Hash* is also a random oracle, which is a collision-resistant hash function. We uses Zipf's law [29] to prove SK security of the proposed protocol, which has been widely applied to verify recent authentication schemes [30,31].

Table 4. Queries of ROR model.

Query	Description
$Execute(P_{SD_i}^{t_1}, P_{UC_j}^{t_2})$	This query denotes that U_A can eavesdrop transmitted messages between SD and UC over insecure channel. This query is modeled as an eavesdropping attack.
$CorruptSD(P_{SD_i}^{t_1})$	This corrupt SG device query means that U_A can extract sensitive information stored in the SG device utilizing power-analysis attack. This query is modeled as an active attack.
$Send(P^t, M)$	This query denotes that U_A can transmit message M to P^t and can also receive the corresponding message from P^t. This query is modeled as an active attack.
$Test(P^t)$	This query means that an unbiased coin c is first flipped before the experiment begins and its output is used as a decider. U_A execute this query and if session key SK_{ij} between SD and UC is fresh, P^t returns SK_{ij} if $c = 1$ or a random number when $c = 0$. Otherwise, it returns the null value $\perp$.
$Reveal(P^t)$	The query means that U_A can compromise SK_{ij} between P^t and its partner in the current session.

Theorem 1. *If Adv_{U_A} denotes the advantage function of a malicious attacker U_A in violating SK security of the proposed authentication scheme, then*

$$Adv_{U_A} \leq \frac{q_h^2}{|Hash|} + 2\{C \cdot q_{send}^s\}$$

where Hash, q_{send} and q_h are the number of Hash query, the number of Send query, and the range space of the hash function $h(.)$, respectively, and s and C are the Zipf's parameters [29].

Proof. Similarly, we adopt the proof as presented in [32,33]. A sequence of four games is denoted by GM_i, where $i \in [0, 3]$ are defined for demonstrating the SK security of the proposed authentication scheme. We denote that $Succ_i$ is the probability a malicious attacker U_A wins the game GM_i. The detailed descriptions of these four games are shown in Game 0–3. □

- **Game** GM_0: This game is the initial game in which U_A selects the random bit c. In addition, this game denotes actual attack of U_A for the protocol and c is guessed at the beginning of G_0. According to this game, we can get,

$$Adv_{U_A} = |2 \cdot Pr[Succ_0] - 1| \quad (1)$$

- **Game** GM_1: This game denotes that U_A performs an eavesdropping attack, in which it intercepts all transmitted messages $\{M_1, M_2, M_3\}$ and $\{M_4, M_5, M_6\}$ during authentication process utilizing *Execute* query. Once the game ends, U_A sends *Test* and *Reveal* queries. The output of the *Test* and

Reveal queries decide if U_A obtains random numbers and shared session key $SK_{ij} = h(R_{SD}||R_{UC})$ between SD and UC. To derive SK_{ij}, U_A needs secret information R_{SD}, R_{UC}, and X_i. Thus, GM_0 and GM_1 are indistinguishable because the winning probability of U_A is not increased. We then get,

$$Pr[Succ_1] = Pr[Succ_0] \tag{2}$$

- **Game** GM_2: In this game, the *Hash* and *Send* queries are simulated. This game is modeled as an active attack, in which a malicious attacker U_A eavesdrops all transmitted messages $\{M_1, M_2, M_3\}$ and $\{M_4, M_5, M_6\}$ during authentication process. All transmitted messages in authentication process are protected by utilizing the collision-resistant one-way hash function $h(.)$. In addition, random numbers R_{SD} and R_{UC} are used in the messages $\{M_1, M_2, M_3\}$ and $\{M_4, M_5, M_6\}$. However, R_{SD} and R_{UC} are not derived from all transmitted messages due to the collision-resistant one-way hash function $h(.)$. U_A makes and sends *Hash* query, and then we can get the result using birthday paradox.

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_h^2}{2|Hash|} \tag{3}$$

- **Game** GM_3: In this the final game, the *CorruptSD* query is simulated. Hence, a malicious attacker U_A can extract the credential informations $\{A_i, B_i, C_i\}$ from memory of the SG device using power-analysis attack. Note that $A_i = X_i \oplus h(RID_i||a_i)$, $B_i = h(RID_i||X_i)$ and $C_i = h(ID_i||B_i) \oplus a_i$. It is computationally infeasible for U_A to derive identity ID_i of SD_i correctly via the *Send* queries without TA's master key K_s and secret parameter X_i. As a result, GM_2 and GM_3 are indistinguishable if identity guessing attack is not implemented. Consequently, utilizing Zipf's law [29], we can get the result as below:

$$|Pr[Succ_3] - Pr[Succ_2]| \leq C \cdot q_{send}^s \tag{4}$$

As all the games are executed, U_A can only guess the exact bit c. Thus, we can get as below:

$$Pr[Succ_3] = \frac{1}{2} \tag{5}$$

Using Equations (1), (2), and (5), we can get the result as below:

$$\begin{aligned} \frac{1}{2}Adv_{U_A} &= |Pr[Succ_0] - \frac{1}{2}| \\ &= |Pr[Succ_1] - \frac{1}{2}| \\ &= |Pr[Succ_1] - Pr[Succ_3]| \end{aligned} \tag{6}$$

Using Equations (4)–(6), we obtain the result utilizing the triangular inequality as below:

$$\begin{aligned} \frac{1}{2}Adv_{U_A} &= |Pr[Succ_1] - Pr[Succ_3]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| \\ &\quad + |Pr[Succ_2] - Pr[Succ_3]| \\ &\leq \frac{q_h^2}{2|Hash|} + max\{C \cdot q_{send}^s\} \end{aligned} \tag{7}$$

Finally, we obtain the required result by multiplying both sides of Equation (7) by a factor of 2.

$$Adv_{U_A} \leq \frac{q_h^2}{|Hash|} + 2max\{C' \cdot q_{send}^{s'}\}$$

7.5. Formal Security Analysis Using AVISPA

AVISPA is a widely used simulation tool for checking whether authentication protocol is secure against replay and MITM attacks. To perform AVISPA simulation, the session and environment of security protocol must be defined using the High-Level Protocol Specification Language (HLPSL). We define three basic roles in HLPSL implementation for the proposed protocol: the SG device SD, the utility server UC, and the trusted authority TA. The *session* and *environments* are shown in Figure 9.

```
role session(SD, UC, TA: agent, SKsdta,SKucta: symmetric_key, H:
hash_func)
def=
local SN1, SN2, SN3, RV1, RV2, RV3: channel(dy)
composition
device(SD,UC,TA, SKsdta, H, SN1, RV1)
/\ center(SD,UC,TA, SKucta, H, SN2, RV2)
/\ authority(SD,UC,TA, SKsdta, SKucta, H, SN3, RV3)
end role

role environment()
def=
const sd, uc, ta : agent,
sksdta,skucta: symmetric_key,
h: hash_func,
idi, idj: text,
uc_sd_ruc, sd_uc_rsd: protocol_id,
sp1,sp2,sp3,sp4: protocol_id

intruder_knowledge = {sd,uc,ta,h,idi,idj}
composition
session(sd,uc,ta,sksdta,skucta,h)/\session(i,uc,ta,sksdta,skucta,h)
/\session(sd,i,ta,sksdta,skucta,h)
/\session(sd,uc,i,sksdta,skucta,h)

end role

goal
secrecy_of sp1, sp2, sp3, sp4
authentication_on uc_sd_ruc, sd_uc_rsd

end goal

environment()
```

Figure 9. Role specification of environment and session.

7.5.1. Detailed Specification of Roles

First, SD receives the initial messages and makes a state value from 0 to 1. SD generates a random number a_i, calculates RID_i, and then SD sends a registration request message $\{RID_i, a_i\}$ to TA over secure channel and changes the state value from 1 to 2. In transition 2, SD receives the secret parameters $\{A_i, B_i\}$ from TA over secure channel. In login and authentication process, SD generates a random number R_{SD} and computes an authentication request message $\{M_1, M_2, M_3\}$. Then, SD sends $\{M_1, M_2, M_3\}$ to utility center UC and updates the state value from 2 to 3. In the last transition, SD receives a authentication message $\{M_4, M_5, M_6\}$ from the UC, computes the session key SK_{ij}, and declares a request function $request(SD, UC, uc\_sd\_ruc, Ruc')$, which means that uc_sd_ruc denotes a strong authentication factor. As

a result, *SD* authenticates *UC* successfully. The specification of a SG device (*SD*) is shown in Figure 10. In Figures 11 and 12, the role specifications of *UC* and *TA* are similarly defined with *SD*.

```
role device(SD, UC, TA : agent, SKsdta : symmetric_key, H: hash_func, SND,
RCV : channel(dy))

played_by SD
def=
 local State: nat,
 IDi,PWi, Ai, PIDi, RPWi, RIDi, Aii, Bi, IDj, RIDj, Ks, Xi, Xii : text,
    Rsd,Ruc,M1,M2,M3,M4,M5,M6,SKij : text
 const sp1, sp2, sp3, sp4, uc_sd_ruc, sd_uc_rsd: protocol_id
init State := 0
transition
1. State = 0 /\ RCV(start) =|>
State' := 1  /\ Ai' := new()
        /\ PIDi' := H(IDi.Ai')
        /\ RPWi' := H(PWi.Ai')
        /\ SND({PIDi'.RPWi'}_SKsdta)
          /\ secret({PIDi'}, sp1, {SD,TA})/\ secret({PWi,Ai'},sp2,{SD})

2. State = 1
/\ RCV({xor(H(H(IDi.Ai').Ks.Xi'),H(H(H(IDi.Ai').H(PWi.Ai')).H(PWi.Ai'))).
H(H(PWi.Ai').H(H(IDi.Ai').Ks.Xi'))}_SKsdta) =|>
State' := 2 /\ Rsd' := new()
       /\ M1' := xor(H(H(IDi.Ai').Ks.Xi'),Rsd')
       /\ M2' := xor(H(H(IDi.Ai').H(PWi.Ai')),H(H(H(IDi.Ai').Ks.Xi').Rsd'))
       /\ M3' := H(H(H(IDi.Ai').H(PWi.Ai')).H(H(IDi.Ai').Ks.Xi').Rsd')
       /\ SND (M1'.M2'.M3')
         /\witness(SD,UC, sd_uc_rsd, Rsd')

3. State = 2 /\ RCV(xor(Ruc',
H(H(H(IDi.Ai').Ks.Xi').Ruc').xor(H(IDj.Ks),Ruc').H(H(H(IDi.Ai').H(PWi.Ai')
).H(H(IDi.Ai').Ks.Xi').Rsd'.Ruc'))) =|>
State' := 3 /\ SKij' := H(Rsd'.Ruc')
          /\request(SD,UC,uc_sd_ruc,Ruc')
end role
```

Figure 10. Role specification of smart grid device.

```
role center(SD, UC, TA : agent, SKucta : symmetric_key, H: hash_func, SND,
RCV : channel(dy))

played_by UC
def=
 local State: nat,
 IDi,PWi, Ai, PIDi, RPWi, RIDi, Aii, Bi, IDj, RIDj, Ks, Xi, Xii : text,
    Rsd,Ruc,M1,M2,M3,M4,M5,M6,SKij : text
 const sp1, sp2, sp3, sp4, uc_sd_ruc, sd_uc_rsd: protocol_id
init State := 0
transition

1. State = 0 /\ RCV(start) =|>
  State' := 1 /\ secret({IDj}, sp4, {UC,TA})
    /\SND({IDj}_SKucta)

2. State = 1
/\ RCV({H(IDj.Ks).H(H(IDi.Ai').H(PWi.Ai')).H(H(IDi.Ai').Ks.Xi')}_SKucta)
=|>
 State' := 2
3. State = 2
/\ RCV(xor(H(H(IDi.Ai').Ks.Xi'),Rsd').xor(H(H(IDi.Ai').H(PWi.Ai')),H(H(H(I
Di.Ai').Ks.Xi').Rsd')).H(H(H(IDi.Ai').H(PWi.Ai')).H(H(IDi.Ai').Ks.Xi').Rsd'))
=|>
 State' := 3 /\ Ruc' := new()
       /\ M4' := xor(Ruc', H(H(H(IDi.Ai').Ks.Xi').Ruc'))
       /\ M5' := xor(H(IDj.Ks),Ruc')
       /\ SKij' := H(Rsd'.Ruc')
       /\ M6' := H(H(H(IDi.Ai').H(PWi.Ai')).H(H(IDi.Ai').Ks.Xi').Rsd'.Ruc')
       /\ SND(M4'.M5'.M6')
       /\ witness(UC,SD,uc_sd_ruc,Ruc')
       /\ request(UC,SD,sd_uc_rsd,Rsd')
end role
```

Figure 11. Role specification of utility center.

```
role authority(SD, UC, TA : agent, SKsdta, SKucta : symmetric_key, H:
hash_func, SND, RCV : channel(dy))

played_by TA
def=
 local State: nat,
 IDi,PWi, Ai, PIDi, RPWi, RIDi, Aii, Bi, IDj, RIDj, Ks, Xi, Xii : text,
    Rsd,M1,M2,M3,M4,M5,M6 : text
 const sp1, sp2, sp3, sp4, uc_sd_ruc, sd_uc_rsd: protocol_id
init State := 0
transition
1. State = 0 /\ RCV({H(IDi.Ai').H(PWi.Ai')}_SKsdta) =|>
State' := 1 /\ Xi' := new()
       /\ RIDi' := H(H(IDi.Ai').H(PWi.Ai'))
       /\ Xii' := H(H(IDi.Ai').Ks.Xi')
       /\ Aii' := xor(Xii',H(RIDi'.H(PWi.Ai')))
       /\ Bi' := H(H(PWi.Ai').Xii')
       /\ secret({Ks}, sp3, {TA})
          /\SND({Aii'.Bi'}_SKsdta)


2. State = 1 /\ RCV({IDj}_SKucta) =|>
State' :=2 /\ RIDj' := H(IDj.Ks)
      /\ Xi' := new() /\ Ai' := new()
      /\ Xii' := H(H(IDi.Ai').Ks.Xi')
      /\ SND({RIDj'.H(H(IDi.Ai').H(PWi.Ai')).H(H(IDi.Ai').Ks.Xi')}_SKucta)

end role
```

Figure 12. Role specification of trusted authority.

7.5.2. Results of AVISPA Analysis

We utilized CL-based Attack Searcher (CL-AtSe) and On-the-fly-Model-Checker (OFMC) back-ends to the verify security of our protocol. The HLPSL code was translated into intermediate format, and then converted to output format using the back-ends. Figure 13 shows the results of simulation using two back-ends. The result of CL-AtSe back-end shows that two states were analyzed and the translation time was 0.10 s. The result of OFMC back-end shows it visited node 1040 nodes with nine plies depth. According to the results of simulation, the proposed protocol is secure against replay and MITM attacks.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/sge.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 6.03s
  visitedNodes: 1040 nodes
  depth: 9 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/sge.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 2 states
  Reachable  : 0 states
  Translation: 0.10 seconds
  Computation: 0.00 seconds
```

Figure 13. AVISPA simulation result using OFMC and CL-AtSe.

8. Performance Analysis

This section compares performances and security feature of proposed scheme with existing schemes [6,20,21,28].

8.1. Computation Overhead

We compared the computation costs of the proposed scheme with existing schemes [6,20,21,28]. We define the parameters based on the work of Kumar et al.'s scheme [6]. T_{cert_ver}, T_{cert}, T_h T_s, T_e, T_m, T_{eca}, T_{ecm}, and T_b denote public key certificate verification, public key certificate generation, one-way hash function, symmetric encryption/decryption, modular exponentiation, multiplication, ECC point addition, ECC point multiplication, and bilinear pairing, respectively. Based on the works in [21,34], we present the execution time for various cryptographic operations in Table 5 and assume $\{T_s \approx T_h, T_m \approx T_e\}$ is negligible because it requires very low execution time. We also assume $T_{eca} \ll T_e$ and $T_{eca} \approx T_h$.

Table 5. Various cryptographic operations based on execution time [21,34].

Entity	T_b	T_{ecm}	T_{mp}	T_e	T_h
Pentium IV	3.16 ms	1.17 ms	1.17 ms	<1 ms	0.01 ms
HiPerSmart Card	0.38 s	0.13 s	0.13 s	<0.1 s	0.001 s

In authentication process, total computation overheads of proposed scheme and Kumar et al.'s scheme are $16T_h$ and $12T_h + 4T_{ecm}$, respectively. Based on the works in [21,34], the total computational overheads of our scheme is 0.011 s and 0.05 ms, which is implemented on HiPerSmart card and Pentium IV platform, respectively. Therefore, we provide better efficiency than existing schemes because our protocol utilizes only hash function and XOR operation. Table 6 shows the analysis result of computation overhead compared to existing schemes.

Table 6. A comparative summary: computation overheads.

Schemes	Total Computation Cost
Wu–Zhou [28]	$7T_{mp} + T_m + 5T_h + T_s + T_{cert} + T_{cert\_ver} \approx 528.91$ ms
Tsai–Lo [20]	$7T_{mp} + 2T_e + 2T_b + 10T_h \approx 635.88$ ms
Odelu et al. [21]	$5T_{mp} + 2T_e + 2T_b + 12T_h \approx 505.72$ ms
Kumar et al. [6]	$12T_h + 4T_{ecm} \approx 268.40$ ms
Ours	$16T_h \approx 11.05$ ms

8.2. Communication Overhead

We first define that timestamp, identity, hash, random number, and ECC cryptosystem are 32, 160, 160, 160, and 320 bits, respectively. In our protocol, transmitted messages $\{M_1, M_2, M_3\}$ and $\{M_4, M_5, M_6\}$ require (160 + 160 + 160 =) 480 and (160 + 160 + 160 =) 480 bits, respectively. As a result, the proposed scheme has more efficient than related schemes [6,20,21,28] because the total communication overhead of proposed protocol is very low compared with the others. Table 7 shows the analysis result of communication overhead compared to existing schemes.

Table 7. A comparative summary: communication overheads.

Schemes	Communication Cost	Number of Messages
Wu–Zhou [28]	3648 bits	4 messages
Tsai–Lo [20]	1408 bits	3 messages
Odelu et al. [21]	1920 bits	3 messages
Kumar et al. [6]	1376 bits	3 messages
Ours	960 bits	2 messages

8.3. Storage Overhead

We first define that identity, hash, timestamp, random number, and public key cryptosystem are 20, 20, 4, 20, and 40 bytes, respectively. In our protocol, stored messages $\{A_i, B_i, C_i\}$ and $\{RID_i, RID_j, X_i\}$ require (20 + 20 + 20 =) 60 and (20 + 20 + 20 =) 60 bytes, respectively. Although the proposed scheme storage overhead of somewhat higher than Kumar et al.'s scheme, it provides better efficiency and security than the other related schemes [6,20,21,28]. Table 8 shows the analysis result of storage overhead compared to existing schemes.

Table 8. A comparative summary: storage overheads.

Schemes	Stored Message (Smart Device)	Stored Message (Utility Center/Service Provider)
Wu–Zhou [28]	-	-
Tsai–Lo [20]	$K_i \approx 40$ bytes	$K_j \approx 40$ bytes
Odelu et al. [21]	$s_i, R_i \approx 80$ bytes	$k_j, K_j \approx 80$ bytes
Kumar et al. [6]	$RID_i, TC_i \approx 40$ bytes	$RID_j, TC_j \approx 40$ bytes
Ours	$A_i, B_i, C_i \approx 60$ bytes	$RID_i, RID_j, X_i \approx 80$ bytes

9. Conclusions

This study demonstrated that Kumar et al.'s scheme cannot defend against various potential attacks such as masquerade, SG device stolen, and session key disclosure attacks. We also showed that Kumar et al.'s scheme does not ensure mutual authentication. To overcome these security shortcomings of Kumar et al.'s scheme, we present a privacy-preserving lightweight authentication protocol for demand response management in the SG environments. Our protocol prevents against various attacks, including masquerade, replay, SG device stolen, and session key disclosure attacks and achieves secure mutual authentication and anonymity. We proved that our protocol ensures secure mutual authentication between SD_i and UC_j using BAN logic, and then we showed that the proposed protocol withstands various potential attacks using informal security analysis and ROR model. We also demonstrated that our scheme was secure against replay and MITM attacks using AVISPA simulation tool. Furthermore, we compared communication overheads, computation overheads, and storage overheads with existing schemes. Therefore, our protocol is applicable for practical SG environments because it is more secure and efficient than other existing schemes.

Author Contributions: Conceptualization, S.Y. and K.P.; Formal analysis, S.Y., K.P. and J.L.; Methodology, J.L., S.L. and B.C.; Software, J.L., S.L. and B.C.; Supervision, Y.P. (YoungHo Park) and Y.P. (YoHan Park); Validation, Y.P. (YoungHo Park) and Y.P. (YoHan Park); Writing—original draft, S.Y.; Writing—review & editing, Y.P. (YoungHo Park) and K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A3B07050409) and in part by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MIST) (No. 2018-0-00312, Developing technologies to predict, detect, respond, and automatically diagnose security threats to automotive Ethernet-based vehicle).

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

References

1. Park, Y.H.; Park, Y.H. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123. [CrossRef] [PubMed]
2. Tonyali, S.; Akkaya, K.; Saputro, N.; Uluagac, A.S.; Nojoumian, M. Privacy–preserving protocols for secure and reliable data aggregation in IoT–enabled Smart Metering systems. *Future Gener. Comput. Syst.* **2018**, *78*, 547–557. [CrossRef]
3. Braeken, A.; Kumar, P.; Martin, A. Efficient and Privacy-Preserving Data Aggregation and Dynamic Billing in Smart Grid Metering Networks. *Energies* **2018**, *11*, 2085. [CrossRef]
4. Kumar, P.; Gurtov, A.; Sain, M.; Martin, A.; Ha, P.H. Lightweight authentication and key agreement for smart metering in smart energy networks. *IEEE Trans. Smart Grid* **2019**, *10*, 4349–4359. [CrossRef]
5. Department of Energy. Exploring the Imperative of Revitalizing America's Electric Infrastructure. February 2017. Available online: https://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DOE_SG_Book_Single_Pages.pdf (accessed on 3 February 2020).
6. Kumar, N.; Aujla, G.S.; Das, A.K.; Conti, M. ECCAuth: Secure authentication protocol for demand response management in smart grid systems. *IEEE Trans. Ind. Inform.* **2019**, *15*, 6572–6582. [CrossRef]
7. Desai, S.; Alhadad, R.; Chilamkurti, N.; Mahmood, A. A survey of privacy preserving schemes in IoE enabled smart grid advanced metering infrastructure. *Clust. Comput.* **2019**, *22*, 43–69. [CrossRef]
8. Dolev, D.; Yao, A.C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
9. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology—CRYPTO*; Lecture Notes in Computer Science; Springer: Santa Barbara, CA, USA, 1999; pp. 388–397.
10. Messerges, T.S.; Dabbish, E.A.; Sloan, R.H. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* **2012**, *51*, 541–552. [CrossRef]
11. Lee, J.Y.; Yu, S.J.; Park, K.S.; Park, Y.H.; Park, Y.H. Secure three-factor authentication protocol for multi-gateway IoT environments. *Sensors* **2019**, *19*, 2358. [CrossRef]
12. Yu, S.J.; Park, K.S.; Park, Y.H. A secure lightweight three–factor authentication scheme for IoT in cloud computing environment. *Sensors* **2019**, *19*, 3598. [CrossRef]
13. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
14. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password based authenticated key exchange in the three-party setting. In Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005; pp. 65–84.
15. Rottondi, C.; Fontana, S.; Verticale, G. Enabling privacy in vehicle-to-grid interactions for battery recharging. *Energies* **2014**, *7*, 2780–2798. [CrossRef]
16. Jiang, Q.; Khan, M.K.; Lu, X.; Ma, J.; He, D. A privacy preserving three–factor authentication protocol for e-Health clouds. *J. Supercomput.* **2016**, *72*, 3826–3849. [CrossRef]
17. Wan, Z.; Zhu, W.T.; Wang, G. PRAC: Efficient privacy protection for vehicle-to-grid communications in the smart grid. *Comput. Secur.* **2016**, *62*, 246–256. [CrossRef]
18. Jo, H.J.; Kim, I.S.; Lee, D.H. Efficient and privacy-preserving metering protocols for smart grid systems. *IEEE Trans. Smart Grid* **2016**, *7*, 1732–1742. [CrossRef]
19. Mahmood, K.; Chaudhry, S.A.; Naqvi, H.; Kumari, S.; Li, X.; Sangaiahm, A.K. An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. *Future Gener. Comput. Syst.* **2018**, *81*, 557–565. [CrossRef]
20. Tsai, J.L.; Lo, N.W. Secure anonymous key distribution scheme for smart grid. *IEEE Trans. Smart Grid* **2016**, *7*, 906–914. [CrossRef]
21. 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]

22. Doh, I.; Lim, J.; Chae, K. Secure authentication for structured smart grid system. In Proceedings of the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'15), Fukuoka, Japan, 8–10 July 2015; pp. 200–204.
23. Saxena, N.; Choi, B.J.; Lu, R. Authentication and authorization scheme for various user roles and devices in smart grid. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 907–921. [CrossRef]
24. He, D.; Wang, H.; Khan, M.K.; Wang, L. Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography. *IET Commun.* **2016**, *10*, 1795–1802. [CrossRef]
25. Wazid, M.; Das, A.K.; Kumar, N.; Rodrigues, J.P.C. Secure three-factor user authentication scheme for renewable energy based smart grid environment. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3144–3153. [CrossRef]
26. Weaver, K. A Perspective on How Smart Meters Invade Individual Privacy. 2014. Available online: https://skyvisionsolutions.files.wordpress.com/2014/08/utility-smart-meters-invade-privacy-22-aug-2014.pdf (accessed on 3 February 2020).
27. Finster, S.; Baumgart, I. Privacy-aware smart metering: A survey. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 1088–1101. [CrossRef]
28. Wu, D.; Zhou, C. Fault-tolerant and scalable key management for smart grid. *IEEE Trans. Smart Grid* **2011**, *2*, 375–381. [CrossRef]
29. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
30. Park, K.S.; Park, Y.H.; Park, Y.H.; Das, A.K. 2PAKEP: Provably secure and efficient two-party authenticated key exchange protocol for mobile environment. *IEEE Access* **2018**, *6*, 30225–30241. [CrossRef]
31. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V. Design of secure key management and user authentication scheme for fog computing services. *Future Gener. Comput. Syst.* **2019**, *91*, 475–492. [CrossRef]
32. Das, A.K.; Wazid, M.; Kumar, N.; Khan, M.K.; Choo, K.K.R.; Park, Y.H. Design of secure and lightweight authentication protocol for wearable devices environment. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 1310–1322. [CrossRef]
33. Srinivas, J.; Das, A.K.; Kumar, N.; Rodrigues, J.J. TCALAS: Temporal credential based anonymous lightweight authentication scheme for internet of drones environment. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6903–6916. [CrossRef]
34. Tseng, Y.M.; Huang, S.S.; Tsai, T.T.; Ke, J.H. List-free id-based mutual authentication and key agreement protocol for multi-server architectures. *IEEE Trans. Emerg. Top. Comput.* **2016**, *4*, 102–112. [CrossRef]