Received May 20, 2019, accepted June 3, 2019, date of publication June 6, 2019, date of current version June 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2921399

A Dynamic Privacy-Preserving Key Management Protocol for V2G in Social Internet of Things

KISUNG PARK[1], YOUNGHO PARK[1], (Member, IEEE), ASHOK KUMAR DAS[2], (Senior Member, IEEE), SUNGJIN YU[1], JOONYOUNG LEE[1], AND YOHAN PARK[3]

[1]School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea
[2]Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[3]Division of IT Convergence, Korea Nazarene University, Cheonan 31172, South Korea

Corresponding author: Youngho Park (parkyh@knu.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT, and Future Planning under Grant 2017R1A2B1002147, and in part by the BK21 Plus Project supported by the Ministry of Education, South Korea, under Grant 21A20131600011.

ABSTRACT With the smart grid (SG) and the social Internet of Things (SIoT), an electric vehicle operator can use reliable, flexible, and efficient charging service with vehicle-to-grid (V2G). However, open channels can be vulnerable to various attacks by a malicious adversary. Therefore, secure mutual authentication for V2G has become essential, and numerous related protocols have been proposed. In 2018, Shen *et al.* proposed a privacy-preserving and lightweight key agreement protocol for V2G in SIoT to ensure security. However, we demonstrate that their protocol does not withstand impersonation, privileged-insider, and offline password guessing attacks, and it does not also guarantee secure mutual authentication, session key security, and perfect forward secrecy. Therefore, this paper proposes a dynamic privacy-preserving and lightweight key agreement protocol for V2G in SIoT to resolve the security weaknesses of Shen *et al.*'s protocol. The proposed protocol resists several attacks including impersonation, offline password guessing, man-in-the-middle, replay, and trace attacks, ensures anonymity, perfect forward secrecy, session key security, and secure mutual authentication. We evaluate the security of the proposed protocol using formal security analysis under the broadly-accepted real-or-random (ROR) model, secure mutual authentication proof using the widely-accepted Burrows-Abadi-Needham (BAN) logic, informal (non-mathematical) security analysis, and also the formal security verification using the broadly-accepted automated validation of Internet security protocols and applications (AVISPA) tool. We then compare computation costs, and security and functionality features of the proposed protocol with related protocols. Overall, the proposed protocol provides superior security, and it can be efficiently deployed to practical SIoT-based V2G environment.

INDEX TERMS Social Internet of Things (SIoT), vehicle-to-grid (V2G), authentication, AVISPA, formal security, key management.

I. INTRODUCTION

With the advances in Internet of Things (IoT) technologies and widespread use of social networks, users can easily access convenient services using Social Internet of Things (SIoT) technologies. SIoT is the convergence of IoT technologies and social networking [1], [2], and it interconnects social relationships with other IoT devices. IoT devices collect and analyze data for various purposes, and can freely exchange data with users and other devices. Hence, SIoT can be efficiently applied to various fields, including smart healthcare, smart factory, smart grids, etc.

A smart grid is an advanced technology that improves conventional power grid reliability, flexibility and efficiency. Vehicle-to-Grid (V2G) network [3], in particular, is an interesting emerging smart grid technology, providing many advantages to smart grids, including renewable energy generation, solving electrical losses, and providing fast electricity supply. However, despite providing these advantages, concerns remain regarding V2G security and privacy due to their

The associate editor coordinating the review of this manuscript and approving it for publication was Vaibhav Rastogi.

general vulnerabilities whereby an adversary can obtain an electric vehicle (EV) owner's location, sensitive information, and exchanged messages. Thus, V2G privacy, integrity and confidentiality must be guaranteed to provide safe and efficient services.

Kempton and Tomic [4] proposed the V2G network concept with numerous V2G concepts subsequently proposed [5]–[10] and many studies investigating V2G security issues [11]–[15]. In 2011, Stegelmann and Kesdogan [11] proposed an anonymity-preserving method using an adversary algorithm, and a privacy-preserving mechanism for the EV location [12]. In 2012, Liu *et al.* [13] proposed an improved location-preserving mechanism to enhance EV privacy. In 2013, Nicanfar *et al.* [14] proposed robust authentication for communication between EV and power station to ensure customer privacy. In 2014, Rottondi *et al.* [15] proposed privacy-preserving and privacy-friendly V2G infrastructure.

Many previous studies considered for V2G and IoT authentication protocols to ensure user privacy, including location, payment, and sensitive data [16]–[21]. In 2011, Yang *et al.* [16] proposed a secure communication protocol using blind signatures to guarantee secure communication. However, Yang *et al.*'s protocol was vulnerable to key escrow attacks. In 2014 and 2015, Choi *et al.* [17] proposed security enhanced user authentication for Wireless Sensor Networks (WSNs) using elliptic curve cryptography (ECC) and Wang *et al.* [18] proposed a traceable privacy-preserving protocol using bilinear pairing. However, both these protocols have high computational overheads and cannot be applied to practical V2G systems. Abdallah and Shen [19], Liu *et al.* [20], Fouda *et al.* [21], and Shen *et al.* [22] proposed lightweight authentication protocols for V2G and smart grids to reduce computation costs. However, the protocols [19] and [20] only use an informal approach to analyze the security of their protocols, and [21] focused on V2G structures. In 2018, Shen *et al.* [22] proposed a practical lightweight authentication protocol for V2G in SIoT to overcome these issues, and claimed the proposed protocol could prevent impersonation, replay, and man-in-the-middle attacks, while also achieving perfect forward secrecy and secure mutual authentication. However, we demonstrate that Shen *et al.*'s protocol does not prevent impersonation and offline password guessing attacks, and it does not achieve perfect forward secrecy, session key security, and secure mutual authentication. Therefore, we propose a more secure dynamic privacy-preserving and lightweight key agreement protocol for V2G in SIoT that resolves these security issues.

A. RESEARCH CONTRIBUTIONS

The main contributions of the work are as follows.

- We show that Shen *et al.*'s proposed protocol does not guarantee security, being vulnerable to impersonation and offline password guessing attacks and it does not also achieve secure mutual authentication and secure key agreement.
- We propose a dynamic privacy-preserving and lightweight key agreement protocol for V2G in SIoT to overcome problems of Shen *et al.*'s protocol. The proposed protocol prevents impersonation, offline password guessing and trace attacks, and guarantees secure mutual authentication, key agreement, anonymity, untraceability and session key security.
- We show that the proposed protocol achieves secure mutual authentication and session key security using Burrows-Abadi-Needham (BAN) logic [23] and the Real-Or-Random (ROR) model, respectively. In addition, we also perform informal analysis to show its security against other potential attacks.
- We simulated the proposed protocol for formal security verification using the "Automated Validation of Internet Security Protocols and Applications (AVISPA) tool".
- A detailed comparative study reveals that the performances for the proposed scheme is superior than other related existing protocols.

B. ORGANIZATION

The remainder of this paper is organized as follows. Section II introduces the necessary background to discuss the proposed protocol. Section III presents the general V2G network system model. Sections IV and V review and cryptanalyze Shen *et al.*'s protocol, respectively. Sections VI and VII propose a dynamic privacy-preserving and lightweight key agreement protocol for V2G in SIoT and its security analysis, respectively. Sections VII-C and VIII perform simulation analysis to prove the proposed protocol security and performance analysis comparison with related protocols, respectively. Finally, Section IX summarizes and concludes the paper.

II. PRELIMINARIES

In this section, we introduce the treat model, and other relevant mathematical preliminaries including the fuzzy extractor used in this paper.

A. THREAT MODEL

This paper uses the broadly-accepted "Dolev-Yao (DY) threat model" [24] to analyze protocol security. Under the DY model, a malicious attacker can delete, inject, modify or eavesdrop messages transmitted over the Internet. Apart from these capabilities of the attacker, we assume the following:

- A malicious attacker can obtain or steal a mobile device from legitimate users, and can then extract values stored in the smart card or mobile device using the power analysis attacks [25], [26].
- A malicious attacker can be a legal user (privileged-insider user) in the system or an outsider, and can attempt various attacks using obtained data [27], [28].

Apart from the DY threat model, we also consider the CK-adversary model [29], which is a more stronger threat model and it is treated as the current *de facto* standard in modeling key-exchange protocols [30]. Under the CK-adversary model, the attacker can compromise secure information like session state, private and session keys in addition to his/her all capabilities under the DY model. Hence, the key-exchange protocols should assure that in the event of ephemeral (short-term) secret leakage, the effect on the security of session keys established among the communicating entities in an authenticated key-exchange protocol should be minimal [31].

We also follow the following assumptions as stated in Amin *et al.*'s scheme [33]. The registered legal users always use the words as passwords and identities from the dictionary available to the adversary $\mathcal{A}$ in the password based user authentication protocols. The password and identity of a legal user can be individually guessed by $\mathcal{A}$. However, guessing both password and identity of a registered user and then verifying those in polynomial time is a computationally infeasible task for $\mathcal{A}$ when the right procedures are adopted (e.g., by not choosing an easy-to-guess password and identity pair). Furthermore, it is also computationally infeasible for $\mathcal{A}$ to guess the secret keys and random numbers (nonces) in polynomial time as these are high entropy entities.

B. FUZZY EXTRACTOR

The fuzzy extractor [32] is a data extraction technique from user biometric data. Biometric data acquisition commonly suffers from recording different values from reality due to various noises. The fuzzy extractor resolves this problem and can uniformly extract a random bit string without noise. Fuzzy extractor procedures are detailed elsewhere [32], [34], but it is based on generation and reproduction processes (*Gen* and *Rep*), respectively).

- *Gen* is a "probabilistic algorithm" that calculates biometric secret data (key) $b_i \in M$, where $M = \{0, 1\}^l$ is a finite l-dimensional metric. After receiving the input biometrics BIO_i, *Gen* uniformly outputs a random bit string b_i, called the biometric secret key and a public reproduction parameter τ_i.
- *Rep* is a "deterministic algorithm" that recovers biometric secret key $b_i \in M$ from inputted noisy biometrics BIO_i' and reproduction parameter τ_i as $b_i = Rep(BIO_i', \tau_i)$ provided that the Hamming distance between the original biometrics BIO_i and current biometrics BIO_i' does not exceed a pre-defined error tolerance threshold value, say et.

An estimate on error tolerance threshold values is given by Cheon *et al.* [35] as follows. If the Hamming distance between the original biometrics BIO_i and current biometrics BIO_i' is HD and the number of bits in input string is b, then $et = \frac{HD}{b}$.

C. ONE-WAY CRYPTOGRAPHIC HASH FUNCTION

Cryptographic one-way hash functions are designed in such a way that they are highly sensitive to even slight perturbations to the input strings. Formally, a "collision-resistant one-way hash function" can be defined as follows [36].

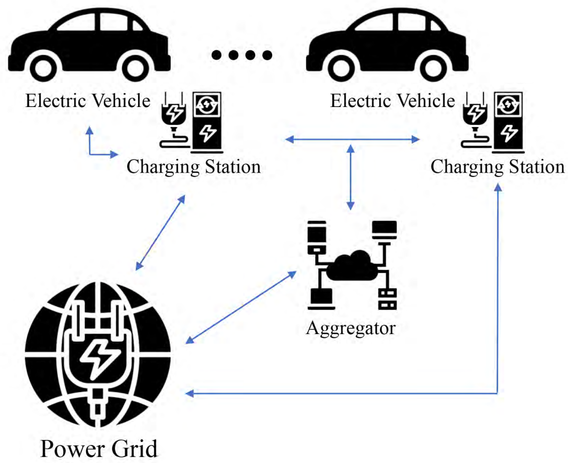

FIGURE 1. Network model for V2G.

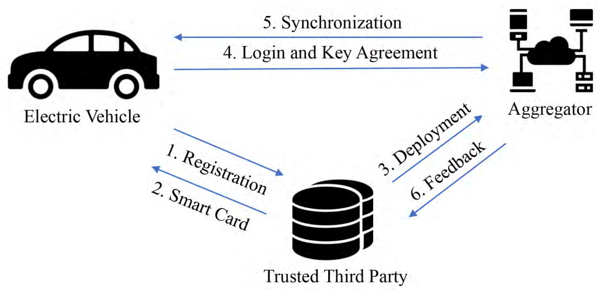

FIGURE 2. V2G system model.

Definition 1: Let h: $\{0, 1\}^* \rightarrow \{0, 1\}^n$ denote a one-way hash function. With a variable length input string, $h(\cdot)$ produces a fixed-size length output string of n bits, called the message digest or hash output. If $Adv_{\mathcal{A}}^{Hash}(t)$ denotes an adversary $\mathcal{A}$'s advantage in finding a hash collision in runtime t, $Adv_{\mathcal{A}}^{Hash}(t) = Pr[(j_1, j_2) \in_R \mathcal{A} : j_1 \neq j_2 \text{ and } h(j_1) = h(j_2)]$, where $Pr[E]$ is the probability of a random event E, and j_1 & j_2 are strings that are randomly selected by $\mathcal{A}$. An (ψ, t)-adversary $\mathcal{A}$ attacking a hash collision of $h(\cdot)$ indicates that $Adv_{\mathcal{A}}^{Hash}(t) \leq \psi$ with maximum permitted execution time t.

III. V2G SYSTEM MODEL

This section introduces the V2G system model and networks. V2G networks incorporate three entities: power grid, EV and charging station, and aggregator (AGT), as shown in Figure 1. A V2G network collects EV battery data and provides efficient power management services. The EV and charging station send monitoring data, such as charging record, payment record, battery status, etc. to the AGT; the AGT collects these data and estimates EV total electricity capacity in the power grid; and the power grid provides electricity to the EV with reasonable price.

Figure 2 shows the authentication process in SIoT based V2G environments to ensure user privacy, such as identity, battery, and payment records. The proposed system incorporates three parties: trusted third party (TTP), EV, and AGT.

Electric vehicle owner (EVO_i)		**Trusted third party** (TTP)
Input ID_i and PW_i Generate a random number r_i Select FPA_i Calculate $SPW_i = h(r_i \vert\vert PW_i)$ $PID_i = h(r_i \vert\vert ID_i)$ $Reg_i = \{SPW_i, PID_i, ID_i, FPA_i\}$		
	$\{Reg_i\}$ $\dashrightarrow$	
		Choose x_i for EVO_i Store PID_i, ID_i, FPA_i, x_i in database Calculate $p_i = h(PID_i \vert\vert x_A)$ $s_i = h(SPW_i \vert\vert x_i)$ $e_i = p_i \oplus s_i$ Issue smart card $V_i = \{PID_i, e_i, p_i, x_i\}$
	$\{V_i\}$ $\dashleftarrow$	
Store r_i in smart card		

FIGURE 3. Registration process of Shen *et al.*'s protocol.

The EV first registers its identity to TTP, and then TTP issues a smart card for the EV and deploys the aggregator. When EV wants to access the V2G system, it performs the login and key agreement procedure to ensure message confidentiality and privacy. After achieving secure mutual authentication between EV and AGT, AGT performs key confirmation and updating to check correct session key distribution and synchronization. Finally, AGT sends feedback to TTP.

IV. REVIEW OF SHEN ET AL.'S PROTOCOL

This section reviews Shen *et al.*'s key agreement protocol for V2G in SIoT. Their proposed protocol comprises three phases: 1) registration, 2) login and key agreement, and 3) key confirmation and pseudonym update. Table 1 shows the notations used in this paper.

A. REGISTRATION PHASE

The EV's owner, EVO_i, registers EV_i to the TTP to enable smart grid services. Figure 3 shows the registration phase for Shen *et al.*'s protocol, with detailed steps as follows.

Step 1: EVO_i chooses its identity ID_i; password PW_i; and a random number r_i. EVO_i then calculates $SPW_i = h(r_i||PW_i)$ and $PID_i = h(r_i||ID_i)$, chooses parking address FPA_i within the service providing region, and sends a registration request $Reg_i = \{SPW_i, PID_i, ID_i, FPA_i\}$ to the TTP through a secure channel.

Step 2: TTP chooses a secret random number x_i for EVO_i after receiving $\{SPW_i, PID_i, ID_i, FPA_i\}$; and stores PID_i, ID_i, FPA_i, and x_i in its database. Then TTP calculates $p_i = h(PID_i||x_A)$, $s_i = h(SPW_i||x_i)$, and $e_i = p_i \oplus s_i$; and issues smart card $V_i = \{PID_i, e_i, p_i, x_i\}$ to EVO_i through a secure channel.

Step 3: EVO_i stores r_i in the smart card upon receiving V_i, and $\{PID_i, e_i, p_i, x_i, r_i\}$ are subsequently included V_i.

TABLE 1. Notations used in this paper.

Notation	Description
EV_i	i^{th} electric vehicle
EVO_i	EV_i's owner
AGT	Aggregator
t_1, t_2, t_3	Current timestamps in messages
t_r	Timestamp when a message is received
PW_i	EVO_i's password
ID_i	EVO_i's real identity
PID_i	EVO_i's pseudo identity
FPA_i	EVO_i's parking address
X_i	EVO_i's secret key selected by TTP
x_A	AGT's long-term secret key
$Skey_i$	Session key of an entity i
$h(\cdot)$	Collision-resistant one-way cryptographic hash function
$\vert\vert$	Concatenation
$\oplus$	Bitwise exclusive-OR (XOR) operation

B. LOGIN AND KEY AGREEMENT PROCESS PHASE

EVO_i can freely access smart grid services after registration. Figure 4 shows the subsequent login and key agreement process for Shen *et al.*'s protocol, with detailed steps as follows.

Step 1: EVO_i inserts the smart card into terminal or onboard unit in EV_i, and inputs its identity ID_i and password PW_i^*.

Electric vehicle & smart card owner (EVO_i/Smart card)		**Aggregator** (AGT)
Input ID_i and PW_i^* Calculate $SPW_i^* = h(r_i\|\|PW_i^*)$ $s_i^* = h(SPW_i^*\|\|x_i)$ $s_i = p_i \oplus e_i$ Check $s_i^* \stackrel{?}{=} s_i$ Generate a random number k_i Choose current timestamp t_1 Calculate $Ver_i = h(s_i\|\|x_i\|\|t_1)$ $MK_i = k_i \oplus p_i$		
	$\{PID_i, e_i, Ver_i, MK_i, t_1\}$ $\dashrightarrow$	
		Check $\|t_r - t_1\| \leq \Delta t$ Retrieve x_i corresponding with PID_i Calculate $p_i^* = h(PID_i\|\|x_A)$ $s_i^* = e_i \oplus p_i^*$ $Ver_i^* = h(s_i^*\|\|x_i\|\|t_1)$ Check $Ver_i^* \stackrel{?}{=} Ver_i$ Generate a random number k_A Calculate $k_i^* = MK_i \oplus p_i^*$ $Skey_a = h(k_A\|\|k_i^*)$ Choose current timestamp t_2 Calculate $MVer_i = h(Ver_i^*\|\|p_i^*\|\|t_1\|\|t_2)$ $KE_{Ai} = h(p_i^*\|\|AID\|\|t_1\|\|t_2) \oplus k_A$
	$\{MVer_i, KE_{Ai}, t_1, t_2\}$ $\dashleftarrow$	
Check $\|t_r - t_2\| \leq \Delta t$ Calculate $MVer_i^* = h(Ver_i\|\|p_i\|\|t_1\|\|t_2)$ Check $MVer_i^* \stackrel{?}{=} MVer_i$ Calculate $k_A^* = KE_{Ai} \oplus h(p_i\|\|AID\|\|t_1\|\|t_2)$ $Skey_i = h(k_A^*\|\|k_i)$		

FIGURE 4. Login and key agreement process of Shen *et al.*'s protocol.

Step 2: The smart card computes $SPW_i^* = h(r_i \,||PW_i^*)$, $s_i^* = h(SPW_i^* \,||x_i)$, and $s_i = e_i \oplus p_i$. Then, it checks $s_i^* \stackrel{?}{=} s_i$. If it is valid, EV_i selects a random number k_i; otherwise, the login process is aborted.

Step 3: EV_i calculates $Ver_i = h(s_i \,||x_i \,||t_1)$ and $MK_i = k_i \oplus p_i$, where t_1 is the current timestamp. Then, EV_i sends login request $\{PID_i, e_i, Ver_i, MK_i, t_1\}$ to AGT.

Step 4: Upon receiving $\{PID_i, e_i, Ver_i, MK_i, t_1\}$, AGT checks $|T_r - T_1| \leq \Delta t$, where t_r is message reception time and Δt is maximum transmission delay bound. If it is valid, AGT retrieves x_i corresponding to PID_i and calculates $p_i^* = h(PID_i \,||x_A)$, $s_i^* = e_i \oplus p_i^*$ and $Ver_i^* = h(s_i^* \,||x_i \,||t_1)$.

Step 5: AGT checks $Ver_i^* \stackrel{?}{=} Ver_i$. If it is valid, AGT generates random number k_A and calculates $k_i^* = MK_i \oplus p_i^*$, session key $Skey_a = h(k_A \,||k_i^*)$, $MVer_i = h(Ver_i^* \,||p_i^* \,||t_1 \,||t_2)$, and $KE_{Ai} = h(p_i^* \,||AID \,||t_1 \,||t_2) \oplus k_A$, where t_2 is the current timestamp and AID is AGT's identity. Finally, AGT sends the message $\{MVer_i, KE_{Ai}, t_1, t_2\}$ to EV_i.

Step 6: Upon the receiving $\{MVer_i, KE_{Ai}, t_1, t_2\}$ from AGT, EV_i checks $|T_r - T_2| \leq \Delta t$. If valid, EV_i calculates $MVer_i^* = h(Ver_i \,||p_i \,||t_1 \,||t_2)$, and then

Electric vehicle & smart card owner (EVO_i/Smart card)		**Aggregator** (AGT)
Choose current timestamp t_3 Calculate $U_i = h(p_i \oplus x_i \oplus Skey_i \oplus t_3)$ Update $PID'_i = h(PID_i \oplus ID_i \oplus x_i)$		
	$\{PID_i, U_i, t_3\}$ $\dashrightarrow$	
		Check $\lvert t_r - t_3 \rvert \leq \Delta t$ Calculate $U_i^* = h(p_i \oplus x_i \oplus Skey_a \oplus t_3)$ Check $U_i^* \stackrel{?}{=} U_i$ Update $PID'_i = h(PID_i \oplus ID_i \oplus x_i)$ Calculate $p'_i = h(PID'_i \| x_A)$ $MP'_i = p'_i \oplus x_A$ $Auth_i = h(PID'_i \oplus p_i)$ Choose current timestamp t_4
	$\{Mp'_i, Auth_i, t_4\}$ $\dashleftarrow$	
Check $\lvert t_r - t_4 \rvert \leq \Delta t$ Calculate $Auth_i^* = h(PID'_i \oplus p_i)$ Checks $Auth_i^* \stackrel{?}{=} Auth_i$ Update $\{PID'_i, e'_i, p'_i\}$		

FIGURE 5. Key confirmation and pseudonym update process of Shen *et al.*'s protocol.

checks $MVer_i^* \stackrel{?}{=} MVer_i$. If they are equal, EV_i computes the session key $Skey_i = h(k_A^* || k_i)$.

After successful completing login and key agreement, EV_i must continuously confirm the key and update the pseudonym to ensure user privacy and check the session key distribution between EV_i and AGT is correct.

C. KEY CONFIRMATION AND PSEUDONYM UPDATE PHASE

EV_i updates its pseudonymous identity PID_i to guarantee user privacy and prevent desynchronization attacks. This process also checks message transmission and successful session key distribution. Figure 5 shows the key confirmation and pseudonym update processes for Shen *et al.*'s protocol, with detailed steps as follows.

Step 1: After login and key agreement, EV_i calculates $U_i = h(p_i \oplus x_i \oplus Skey_i \oplus t_3)$, where t_3 is the current timestamp, and then replaces PID_i with PID'_i for the smart card and sends request messages $\{PID_i, U_i, t_3\}$ to the AGT.

Step 2: Upon receiving request message $\{PID_i, U_i, t_3\}$, AGT checks the condition $|t_r - t_3| \leq \Delta t$; retrieves p_i and e_i corresponding to PID_i; calculates $U_i^* = h(p_i \oplus x_i \oplus Skey_a \oplus t_3)$; checks $U_i^* \stackrel{?}{=} U_i$; replaces PID_i with PID'_i; and calculates $p'_i = h(PID'_i || x_A)$, $MP'_i = p'_i \oplus x_i$, and $Auth_i = h(PID'_i \oplus p_i)$. Finally, AGT sends response messages $\{MP'_i, Auth_i, t_4\}$.

Step 3: Upon receiving the response message from AGT, EV_i checks the condition $|t_r - t_4| \leq \Delta t$; calculates $Auth_i^* = h(PID'_i \oplus p_i)$; and checks $Auth_i^* \stackrel{?}{=} Auth_i$. If they match, EV_i replaces $\{PID_i, e_i, p_i\}$ with $\{PID'_i, e'_i, p'_i\}$.

V. CRYPTANALYSIS OF SHEN ET AL.'S PROTOCOL

This section highlights various security flaws in Shen *et al.*'s protocol. Shen *et al.* claimed their proposed protocol was secure against impersonation and man-in-the-middle attacks, and achieved perfect forward security. However, we prove that Shen *et al.*'s protocol is not secure against the following attacks.

A. IMPERSONATION ATTACK

Section II-A introduces the threat model to analyze the security of the protocol proposed in this paper. Suppose that an attacker U_{at} can obtain the smart card of legal user EVO_i and intercept messages transmitted in previous and current sessions. Further, suppose U_{at} obtains the values $\{PID_i, e_i, p_i, x_i, r_i\}$ stored in the smart card using the power analysis attacks [25], [26]. Finally, U_{at} performs impersonation attack using the following detailed steps.

Step 1: U_{at} generates a random number k_{at}, and computes $s_i = p_i \oplus e_i$, $Ver_a = h(s_i \,||x_i\,||t_1)$, and $MK_{at} = k_{at} \oplus p_i$, where p_i, e_i and x_i are stored in the smart card; and U_{at} sends the message $\{PID_i, e_i, Ver_{at}, MK_{at}, t_1\}$ to AGT, where PID_i is stored in the smart card and t_1 is the current timestamp.

Step 2: Upon receiving $\{PID_i, e_i, Ver_{at}, MK_{at}, t_1\}$, AGT checks $|T_r - T_1| \leq \Delta t$, where t_r is the message reception time and Δt is the maximum transmission delay bound. If it is valid, AGT retrieves x_i corresponding to PID_i, and computes $p_i^* = h(PID_i\,||x_A)$, $s_i^* = e_i \oplus p_i^*$, and $Ver_{at}^* = h(s_i^*\,||x_i\,||t_1)$.

Step 3: AGT checks $Ver_{at}^* \stackrel{?}{=} Ver_{at}$. If is correct, AGT generates a random number k_A and calculates $k_{at}^* = MK_{at} \oplus p_i^*$, session key $Skey_a = h(k_A\,||k_{at}^*)$, $MVer_i = h(Ver_{at}^*\,||p_i^*\,||t_1\,||t_2)$, and $KE_{Ai} = h(p_i^*\,||AID\,||t_1\,||t_2) \oplus k_A$; where t_2 is the current timestamp and AID is AGT's identity. Finally, AGT sends the message $\{MVer_i, KE_{Ai}, t_1, t_2\}$ to U_{at}.

Step 4: Upon receiving $\{MVer_i, KE_{Ai}, t_1, t_2\}$ from AGT, U_{at} checks $|T_r - T_2| \leq \Delta t$. If it is valid, U_{at} computes $MVer_i^* = h(Ver_{at}\,||p_i\,||t_1\,||t_2)$, and then checks $MVer_i^* \stackrel{?}{=} MVer_i$. If it is also valid, U_{at} calculates the session key $Skey_i = h(k_A^*\,||k_{at})$.

As a result, U_{at} generates $Skey_i$ and performs key confirmation and pseudonym update, i.e., U_{at} successfully impersonates legal user EVO_i. Thus, Shen *et al.*'s protocol does not prevent impersonation attack.

B. OFFLINE PASSWORD GUESSING ATTACK

In Shen *et al.*'s protocol, an attacker U_{at} can obtain $s_i = h(SPW_i\,||x_i)$ to calculate $s_i = p_i \oplus e_i$. Then, U_{at} can guess the password of a legitimate user EVO_i as follows. U_{at} guesses some password PW_i^* and calculates $SPW_i^* = h(r_i\,||PW_i^*)$, where r_i is stored in smart card, and then $s_i^* = h(SPW_i^*\,||x_i)$, where x_i is the value stored in smart card. Finally, U_{at} checks $s_i^* \stackrel{?}{=} s_i$. If it is valid, U_{at} has correctly guessed EVO_i's password. Thus, Shen *et al.*'s protocol does not resist offline password guessing attack.

C. PRIVILEGED-INSIDER ATTACK

In this attack, we assume that a privileged-insider user of the TTP, being an insider attacker, say $\mathcal{A}$ has the registration information $\{SPW_i, PID_i, ID_i, FPA_i\}$ that were supplied by the legal registered user EVO_i during the registration process of Shen *et al.*'s protocol. It is worth noticing that $SPW_i = h(r_i\,||PW_i)$. Now, we further assume that $\mathcal{A}$ can obtain the lost or stolen smart card of EVO_i after completing the registration process. Hence, $\mathcal{A}$ will have all the extracted credentials including r_i stored in the smart card of EVO_i using the power analysis attacks [25], [26]. Next, $\mathcal{A}$ can guess a password PW_i^*, computes $SPW_i^* = h(r_i\,||PW_i^*)$ and checks $SPW_i^* \stackrel{?}{=} SPW_i$. If it is valid, $\mathcal{A}$ will be successful in guessing correct password PW_i of EVO_i. Thus, Shen *et al.*'s protocol does not resist privileged-insider attack.

D. MUTUAL AUTHENTICATION AND KEY AGREEMENT

In Section V-A, we showed that U_{at} can successfully generate the session key and can also impersonate a legal user EVO_i. Thus, Shen *et al.*'s protocol does not achieve secure mutual authentication and key agreement.

E. PERFECT FORWARD SECURITY

Assume that U_{at} obtains the smart card data and messages transmitted in the public channel. U_{at} can then calculate the session key for a legal user EVO_i. U_{at} computes $s_i = p_i \oplus e_i$ and $Ver_i = h(s_i\,||x_i\,||t_1)$, retrieves $k_i = p_i \oplus MK_i$, calculates $MVer^* = h(Ver_i\,||p_i\,||t_1\,||t_2)$, and also retrieves $k_A^* = KE_{Ai} \oplus h(p_i\,||AID\,||t_1\,||t_2)$. Finally, U_{at} computes the session key $Skey = h(k_i\,||k_A^*)$. Thus, Shen *et al.*'s protocol does not provide perfect forward security without compromising long-term secret parameters.

VI. THE PROPOSED PROTOCOL

This section proposes a more secure dynamic privacy-preserving and lightweight key agreement protocol for V2G in SIoT by resolving various security weaknesses of Shen *et al.*'s protocol (discussed in Section V). The proposed protocol consists of three phases: 1) registration, 2) login and key agreement, and 3) key confirmation and pseudonym update. We also utilize the notations and their significance listed in Table 1 for describing the proposed protocol. To assure resilience against replay attack, current timestamps have been used in the proposed protocol. Thus, the clocks of all involved entities are assumed to be synchronized. This is a typical assumption in the literature, such as the schemes presented in [37]–[43].

A. REGISTRATION PHASE

EVO_i must first register with the TTP to access V2G services. Figure 6 shows the proposed protocol's registration process with detailed procedures as follows.

Step 1: EVO_i chooses identity ID_i, password PW_i, and FPA_i; and then imprints biometrics BIO_i, such as fingerprint, iris, palmprint, etc. EVO_i calculates $Gen(BIO_i) = \langle b_i, \tau_i \rangle$, $SPW_i = h(PW_i\,||b_i)$, $PID_i = h(ID_i\,||b_i)$, and $p_i = h(PID_i\,||b_i)$ and sends the registration request message $\{SPW_i, FPA_i, PID_i\}$ to the TTP through a secure channel.

Step 2: Upon receiving $\{SPW_i, FPA_i, PID_i\}$ from EVO_i, TTP chooses a secret random number x_i for EVO_i, calculates $s_i = h(SPW_i\,||x_i)$, stores the information $\{PID_i, FPA_i, x_i\}$, and issues smart card $V_i = \{PID_i, x_i, s_i\}$ to EVO_i through a secure channel.

Step 3: Upon receiving V_i from TTP, EVO_i computes $Ex_i = x_i \oplus h(PW_i\,||ID_i\,||b_i)$, $a_i = s_i \oplus h(ID_i\,||b_i\,||PW_i)$, $C_i = s_i \oplus h(p_i\,||PW_i\,||ID_i)$, and

Electric vehicle owner (EVO_i)		**Trusted third party** (TTP)
Input ID_i, PW_i Select FPA_i Imprint BIO_i Calculate $Gen(BIO_i) = \langle b_i, \tau_i \rangle$ $SPW_i = h(PW_i \vert\vert b_i)$ $PID_i = h(ID_i \vert\vert b_i)$ $p_i = h(PID_i \vert\vert b_i)$		
	$\{SPW_i, FPA_i, PID_i\}$ $\dashrightarrow$	
		Choose x_i for EVO_i Compute $s_i = h(SPW_i \vert\vert x_i)$ Store PID_i, FPA_i, x_i, s_i in secure database Issue smart card $V_i = \{PID_i, x_i, s_i\}$
	$\{V_i\}$ $\dashleftarrow$	
Calculate $Ex_i = x_i \oplus h(PW_i \vert\vert ID_i \vert\vert b_i)$ $a_i = s_i \oplus h(ID_i \vert\vert b_i \vert\vert PW_i)$ $C_i = s_i \oplus h(p_i \vert\vert PW_i \vert\vert ID_i)$ $PID_i^* = PID_i \oplus h(b_i \vert\vert PW_i \vert\vert ID_i)$ Store Ex_i, a_i, C_i, PID_i^*, τ_i and et in smart card Also, store $h(\cdot)$, $Gen(\cdot)$ and $Rep(\cdot)$ in smart card Delete x_i, s_i from the smart card		

FIGURE 6. Registration process of the proposed scheme.

$PID_i^* = PID_i \oplus h(b_i ||PW_i ||ID_i)$, and then stores $Ex_i, a_i, C_i, PID_i^*, \tau_i$ and et in the smart card. Finally, the smart card V_i contains the information $\{PID_i^*, C_i, Ex_i, a_i, h(\cdot), Gen(\cdot), Rep(\cdot), et\}$. In addition, the smart card also deletes other information x_i and s_i from its memory.

B. LOGIN AND KEY AGREEMENT PHASE

The registered EVO_i can freely access V2G services using the smart card and biometrics as shown in Figure 7, with the detailed procedures as follows.

Step 1: EVO_i inserts the smart card and imprints biometrics BIO_i into a terminal or onboard unit in EV_i, and then inputs its ID_i and PW_i.

Step 2: The smart card calculates $Rep(BIO_i, \tau_i) = b_i$ provided that the Hamming distance between the registered biometrics and current biometrics does not exceed the pre-defined error tolerance threshold value et, $SPW_i = h(PW_i ||b_i)$, $PID_i = PID_i^* \oplus h(b_i ||PW_i ||ID_i)$, $p_i^* = h(PID_i ||b_i)$, $s_i^* = a_i \oplus h(ID_i ||b_i ||PW_i)$, $x_i = Ex_i \oplus h(PW_i ||ID_i ||b_i)$ and $C_i^* = s_i^* \oplus h(p_i^* ||PW_i ||ID_i)$, and then checks $C_i^* \stackrel{?}{=} C_i$. If it is valid, the smart card generates a random number k_i, and computes $Ver_i = h(s_i^* ||x_i ||PID_i ||t_1 ||k_i)$ and $MK_i = k_i \oplus h(s_i^* || PID_i||t_1)$, where t_1 is the current timestamp. The smart card then sends the login request message $\{PID_i, Ver_i, MK_i, t_1\}$ to AGT through open channel.

Step 3: Upon receiving $\{PID_i, Ver_i, MK_i, t_1\}$, AGT checks if $|t_r - t_1| \leq \Delta t$, where t_r is the message reception time and Δt is the maximum transmission delay bound. If it is valid, AGT retrieves $\{s_i, x_i\}$ corresponding to PID_i, and calculates $k_i = MK_i \oplus h(s_i ||PID_i ||t_1)$ and $Ver_i^* = h(s_i ||x_i ||PID_i ||t_1 ||k_i)$. AGT then checks $Ver_i^* \stackrel{?}{=} Ver_i$. If it is valid, AGT continues to generate a random number k_a; and calculates the session key $Skey_a = h(k_a ||k_i ||s_i)$ shared with EV_i, $MVer_i = h(Ver_i ||s_i ||t_1 ||t_2)$, and $KE_{Ai} = h(s_i ||AID ||t_1 ||t_2) \oplus k_a$, where AID is the AGT's identity and t_2 is the current timestamp. Finally, AGT sends the response message $\{MVer_i, KE_{Ai}, t_1, t_2\}$ to EV_i through open channel.

Step 4: Upon receiving the response message $\{MVer_i, KE_{Ai}, t_1, t_2\}$ from AGT, EV_i checks if $|t_r - t_2| \leq \Delta t$, where t_r is the message reception time. If it is

Owner of Electric Vehicle & smart card(EVO_i/Smart card)		**Aggregator**(AGT)
Input ID_i, PW_i Imprint BIO_i Calculate $Rep(BIO_i, \tau_i) = b_i$ $SPW_i = h(PW_i\|\|b_i)$ $PID_i = PID_i^* \oplus h(b_i\|\|PW_i\|\|ID_i)$ $p_i^* = h(PID_i\|\|b_i)$ $s_i^* = a_i \oplus h(ID_i\|\|b_i\|\|PW_i)$ $x_i = Ex_i \oplus h(PW_i\|\|ID_i\|\|b_i)$ $C_i^* = s_i^* \oplus h(p_i^*\|\|PW_i\|\|ID_i)$ Check $C_i^* \stackrel{?}{=} C_i$ Generate a random number k_i Choose current timestamp t_1 Calculate $Ver_i = h(s_i^*\|\|x_i\|\|PID_i\|\|t_1\|\|k_i)$ $MK_i = k_i \oplus h(s_i^*\|\|PID_i\|\|t_1)$		
	$\{PID_i, Ver_i, MK_i, t_1\}$ $\dashrightarrow$	
		Check $\|t_r - t_1\| \leq \Delta t$ Retrieve s_i, x_i corresponding to PID_i Compute $k_i = MK_i \oplus h(s_i\|\|PID_i\|\|t_1)$ $Ver_i^* = h(s_i\|\|x_i\|\|PID_i\|\|t_1\|\|k_i)$ Check $Ver_i^* \stackrel{?}{=} Ver_i$ Generate a random number k_a Compute $Skey_a = h(k_a\|\|k_i\|\|s_i)$ Choose current timestamp t_2 Calculate $MVer_i = h(Ver_i\|\|s_i\|\|t_1\|\|t_2)$ $KE_{Ai} = h(s_i\|\|AID\|\|t_1\|\|t_2) \oplus k_a$
	$\{MVer_i, KE_{Ai}, t_1, t_2\}$ $\leftarrow - -$	
Check $\|t_r - t_2\| \leq \Delta t$ Calculate $MVer_i^* = h(Ver_i\|\|s_i^*\|\|t_1\|\|t_2)$ Check $MVer_i^* \stackrel{?}{=} MVer_i$ Calculate $k_a^* = KE_{Ai} \oplus h(s_i^*\|\|AID\|\|t_1\|\|t_2)$ $Skey_i = h(k_a^*\|\|k_i\|\|s_i^*)$		

FIGURE 7. Login and key agreement process of the proposed scheme.

valid, EV_i computes $MVer_i^* = h(Ver_i\,||s_i^*\,||t_1\,||t_2)$ and checks if $MVer_i^* = MVer_i$. If it is valid, EV_i calculates $k_a^* = KE_{Ai} \oplus h(s_i^*\,||AID\,||t_1\,||t_2)$ and the session key $Skey_i = h(k_a^*\,||k_i\,||s_i^*)$ shared with AGT.

After finishing this phase, EV_i performs key confirmation and pseudonym update to ensure user privacy and session key security. Thus, mutual authentication between EV_i and AGT occurs in the proposed protocol, and both EV_i and AGT share the same session key $Skey_i$ $(= Skey_a)$.

C. KEY CONFIRMATION AND PSEUDONYM UPDATE PHASE

This process updates EV_i's pseudonymous identity PID_i and secret parameter s_i to ensure user privacy and resist various attacks. This process also guarantees session key and transmitted message security. Figure 8 shows the proposed confirmation and pseudonym update process with detailed procedures as follows.

Step 1: After successful login and key agreement, EV_i calculates $U_i = h(s_i^*\,||x_i\,||Skey_i\,||t_3)$, where t_3 is the

Electric vehicle & smart card owner(EVO_i/Smart card)	**Aggregator**(AGT)
Choose current timestamp t_3 Calculate $U_i = h(s_i^*\|\|x_i\|\|Skey_i\|\|t_3)$ $D_i = SPW_i \oplus h(Skey_i\|\|t_3)$	
$\{PID_i, U_i, D_i, t_3\}$ $\dashrightarrow$	
	Check $\|t_r - t_3\| \leq \Delta t$ Calculate $U_i^* = h(s_i\|\|x_i\|\|Skey_a\|\|t_3)$ $SPW_i = D_i \oplus h(Skey_a\|\|t_3)$ Check $U_i^* \stackrel{?}{=} U_i$ Calculate $s_i^{new} = h(SPW_i\|\|x_i\|\|Skey_a)$ $PID_i^{new} = h(PID_i\|\|s_i^{new}\|\|Skey_a)$ Choose current timestamp t_4 $S_1 = s_i^{new} \oplus h(PID_i\|\|Skey_a)$ $S_2 = PID_i^{new} \oplus h(s_i^{new}\|\|Skey_a)$ $Auth_i = h(PID_i^{new}\|\|s_i^{new}\|\|t_4)$
$\{S_1, S_2, Auth_i, t_4\}$ $\dashleftarrow$	
Check $\|t_r - t_4\| \leq \Delta t$ Calculate $h(PID_i\|\|Skey_i)$ $s_i^{new} = S_1 \oplus h(PID_i\|\|Skey_i)$ $PID_i^{new} = S_2 \oplus h(s_i^{new}\|\|Skey_i)$ $Auth_i^* = h(PID_i^{new}\|\|s_i^{new}\|\|t_4)$ Check $Auth_i^* \stackrel{?}{=} Auth_i$ Calculate $p_i = h(PID_i\|\|b_i)$ $C_i^{new} = s_i^{new} \oplus h(p_i\|\|PW_i\|\|ID_i)$ $a_i^{new} = s_i^{new} \oplus h(ID_i\|\|b_i\|\|PW_i)$ Choose current timestamp t_5 Calculate $AuthVer_i = h(s_i^{new}\|\| PID_i^{new}\|\|t_5)$	
$\{AuthVer_i, t_5\}$ $\dashrightarrow$	
	Check $\|t_r - t_5\| \leq \Delta t$ Compute $AuthVer_i^* = h(s_i^{new}\|\| PID_i^{new}\|\|t_5)$
Compute $PID_i^{**} = PID_i^{new} \oplus h(b_i \|\|PW_i \|\|ID_i)$ Replace $\{PID_i^*, C_i, a_i\}$ with $\{PID_i^{**}, C_i^{new}, a_i^{new}\}$ in the smart card	Check $AuthVer_i^* \stackrel{?}{=} AuthVer_i$ Replace $\{s_i, PID_i\}$ with $\{s_i^{new}, PID_i^{new}\}$ in a secure database

FIGURE 8. Key confirmation and pseudonym update process of the proposed scheme.

current timestamp and $D_i = SPW_i \oplus h(Skey_i \|\|t_3)$. EV_i then sends a request message $\{PID_i, U_i, D_i, t_3\}$ to AGT through open channel.

Step 2: Upon receiving request message $\{PID_i, U_i, D_i, t_3\}$, AGT checks $|t_r - t_3| \leq \Delta t$. If it is valid, AGT calculates $U_i^* = h(s_i \|\|x_i \|\|Skey_a \|\|t_3)$ and $SPW_i = D_i \oplus h(Skey_a \|\|t_3)$, and then checks $U_i^* \stackrel{?}{=} U_i$. If it is valid, AGT computes $s_i^{new} = h(SPW_i \|\|x_i \|\|Skey_a)$, $PID_i^{new} = h(PID_i \|\|s_i^{new} \|\|Skey_a)$, $S_1 = s_i^{new} \oplus h(s_i^{new} \|\|Skey_a)$, $S_2 = PID_i^{new} \oplus h(s_i^{new} \|\|Skey_a)$, and $Auth_i = h(PID_i^{new} \|\|s_i^{new} \|\|t_4)$, where t_4 is the current timestamp. AGT sends the response message $\{S_1, S_2, Auth_i, t_4\}$ to EV_i through open channel.

Step 3: Upon receiving the response message $\{S_1, S_2, Auth_i, t_4\}$ from AGT, EV_i checks if $|t_r - t_4| \leq \Delta t$. If it is valid, EV_i calculates $h(PID_i \|\|Skey_i)$, $s_i^{new} = S_1 \oplus h(PID_i \|\|Skey_i)$, $PID_i^{new} = S_2 \oplus h(s_i^{new}$

$||Skey_i)$, $Auth_i^* = h(PID_i^{new} ||s_i^{new} ||t_4)$; and then checks if $Auth_i^* = Auth_i$. If it is valid, EV_i calculates $p_i = h(PID_i ||b_i)$, $C_i^{new} = s_i^{new} \oplus h(p_i|| PW_i ||ID_i)$ and $a_i^{new} = s_i^{new} \oplus h(ID_i ||b_i ||PW_i)$. After that EV_i generates current timestamp t_5, calculates $AuthVer_i = h(s_i^{new}|| PID_i^{new} ||t_5)$ and sends the acknowledgment message $\{AuthVer_i, t_5\}$ to AGT through open channel. In addition, EV_i computes $PID_i^{**} = PID_i^{new} \oplus h(b_i ||PW_i ||ID_i)$ and replaces $\{PID_i^*, C_i, a_i\}$ with $\{PID_i^{**}, C_i^{new}, a_i^{new}\}$ in the smart card.

Step 4: Upon receiving the acknowledgment message $\{AuthVer_i, t_5\}$ from EV_i, AGT checks if $|t_r - t_5| \leq \Delta t$. If it is valid, AGT calculates $AuthVer_i^* = h(s_i^{new} || PID_i^{new} ||t_5)$ and checks if $AuthVer_i^* = AuthVer_i$. If it is valid, AGT updates $\{s_i, PID_i\}$ with $\{s_i^{new}, PID_i^{new}\}$ in its secure database.

VII. SECURITY ANALYSIS

This section analyzes the proposed protocol security using "formal security analysis through the widely-accepted Real-or-Random (ROR) model" [46]. Furthermore, "mutual authentication proof is carried out with the help of the broadly-accepted Burrow-Abadi-Needham (BAN) logic" [23] and the "formal security verification using the widely-accepted Automated Validation of Internet Security Protocols and Applications (AVISPA)" tool. In addition, the informal (non-mathematical) security analysis also reveals that the proposed protocol is secure against other various attacks.

A. FORMAL SECURITY ANALYSIS THROUGH REAL-OR-RANDOM MODEL

The ROR model [46] is applied in order to prove the semantic security of the proposed protocol. Using the ROR model, we prove that the proposed protocol satisfies the "session key security (SK-security)". We now discuss shortly the ROR model before proving the SK-security of the proposed protocol in Theorem 1.

Under the ROR model, an adversary, say $\mathcal{A}$ interacts with the t^{th} instance of an executing participant, say $\mathcal{P}^t$. In the proposed protocol, EV_i, TTP or AGT is considered as $\mathcal{P}^t$. Let $\mathcal{P}_{EV_i}^{t_1}$, $\mathcal{P}_{AGT}^{t_2}$ and $\mathcal{P}_{TTP}^{t_3}$ are the t_1^{th}, t_2^{th} & t_3^{th} instances of EV_i, AGT and TTP, respectively. Moreover, the ROR model considers various queries simulating a real attack, such as *Execute*, *CorruptSC*, *Reveal*, *Send* and *Test* queries that are shown in Table 2. In addition, a "collision-resistant cryptographic one-way hash function $h(\cdot)$ is modeled as a random oracle, say *Hash*", which is also available to all the communicating participants including the adversary $\mathcal{A}$.

Wang *et al.* [47] discovered that "the user-chosen passwords follow the Zipf's law that is a vastly different distribution from the uniform distribution". Also, "the size of password dictionary is generally much constrained in the sense that the users will not use the whole space of passwords, but rather a small space of the allowed characters space" [47].

TABLE 2. Various queries and their significance.

Query	Significance
$Execute(\mathcal{P}_{EV_i}^{t_1}, \mathcal{P}_{AGT}^{t_2}, \mathcal{P}_{TTP}^{t_3})$	Using this query, $\mathcal{A}$ can eavesdrop the messages exchanged between the communicating entities EV_i, AGT and TTP through open channels. This is modeled as an "eavesdropping attack".
$CorruptSC(\mathcal{P}_{EV_i}^{t})$	Under this corrupt smart card query, $\mathcal{A}$ can extract all the sensitive credentials stored in the smart card of EVO_i. This is modeled as an "active attack".
$Reveal(\mathcal{P}^t)$	Under this query, the current session key $Skey_i/Skey_a$ between $\mathcal{P}^t$ and its partner to $\mathcal{A}$ is revealed to $\mathcal{A}$.
$Send(\mathcal{P}^t, Msg)$	Using this query, "$\mathcal{A}$ can transmit a message Msg to $\mathcal{P}^t$, and in response, it can also receive the message from $\mathcal{P}^t$". It is modeled as an "active attack".
$Test(\mathcal{P}^t)$	Under this query, "an unbiased coin c is flipped before the game begins. Depending on the outcome, the following decision is taken. $\mathcal{A}$ executes this query and if the session key $Skey_i/Skey_a$ between EV_i and AGT is fresh, $\mathcal{P}^t$ returns $SKey_i/Skey_a$ if $c = 1$ or a pure random number if $c = 0$; otherwise, it will return a null value $(\perp)$".

The Zipf's law is applied in the formal security analysis to prove the SK security of the proposed protocol.

In the following, we prove that the proposed protocol satisfies the SK-security.

Theorem 1: If $Adv_{\mathcal{A}}^{AKM}$ is the advantage function of an adversary $\mathcal{A}$ in breaking the SK-security of the proposed authenticated key-management (AKM) protocol, q_h, q_s and $|Hash|$ are "the number of *Hash* queries, the number of *Send* queries and the range space of the hash function $h(\cdot)$", respectively, l_b is the number of bits present in the EVO_i's biometric secret key b_i, and C' and s' denote the Zipf's parameters [47], then

$$Adv_{\mathcal{A}}^{AKM} \leq \frac{q_h^2}{|Hash|} + 2\max\left\{C' \cdot q_s^{s'}, \frac{q_s}{2^{l_b}}\right\}$$

proof 1: The similar proof as applied in [38], [43], [48], [49] is followed here. We define the four games, namely G_j, $j \in [0, 3]$ in which an event is also defined wherein "$\mathcal{A}$ can guess the random bit c in the G_j correctly" and its success probability is defined by $Succ_{\mathcal{A}}^{G_j}$. In addition, the "advantage of $\mathcal{A}$ in winning the game G_j" is denoted and defined by $Adv_{\mathcal{A},G_j}^{AKE} = Pr[Succ_{\mathcal{A}}^{G_j}]$.

Next, we provide the details of the above defined games $G_j, j \in [0, 3]$ below.

- **Game** G_0: This game corresponds to the "actual attack executed by $\mathcal{A}$ against our proposed protocol in the ROR model" with respect to the game G_0. As the bit c is selected randomly at the beginning of G_0, we get,

$$Adv_{\mathcal{A}}^{AKM} = |2.Adv_{\mathcal{A},G_0}^{AKM} - 1| \quad (1)$$

- **Game** G_1: This game is modeled as an "eavesdropping attack" in which the adversary $\mathcal{A}$ can eavesdrop all the communicated messages, say $Msg_1 = \{PID_i, Ver_i,$

MK_i, $t_1\}$ and $Msg_2 = \{MVer_i, KE_{Ai}, t_1, t_2\}$ during the login and key agreement process of the proposed scheme (Section VI-B) using the *Execute* query defined in Table 2. Once the game ends, $\mathcal{A}$ can execute the *Reveal* and *Test* queries to verify the following: "if the derived session key $Skey_i/Skey_a$ between EV_i and AGT is actual or a random key". It is worth noticing that the session key is constructed as $Skey_a = h(k_a\ ||k_i\ ||s_i) = h(k_a^*\ ||k_i\ ||s_i) = Skey_i$. To derive $Skey_i/Skey_a$, $\mathcal{A}$ needs the temporal (short-term) secrets (k_i and k_a) and also long term secret (s_i) which are unknown to $\mathcal{A}$. This shows that only eavesdropping of the messages Msg_1 and Msg_2 does not at all help in increasing the game G_1's winning probability of $\mathcal{A}$. As both the games G_0 and G_1 are indistinguishable, we then have

$$Adv^{AKM}_{\mathcal{A},G_1} = Adv^{AKM}_{\mathcal{A},G_0} \tag{2}$$

- **Game** G_2: This game includes the simulation of the *Hash* query to model it as an "active attack". In the message Msg_1, the terms PID_i, Ver_i and MK_i are protected by the "collision-resistant cryptographic one-way hash function $h(\cdot)$ (see Definition 1)". Also, in the message Msg_2, the terms $MVer_i$ and KE_{Ai} are protected by $h(\cdot)$. Furthermore, deriving s_i and k_i from the intercepted Ver_i, and MK_i, and also s_i and k_a from the intercepted $MVer_i$ and KE_{Ai} are "computationally infeasible task" due to "collision-resistant property of $h(\cdot)$". In addition, all the random numbers, current timestamps and secret credentials are used in the messages Msg_1 and Msg_2. Therefore, no collision happens if the *Hash* query is executed by the adversary $\mathcal{A}$. Since both the games G_1 and G_2 are "indistinguishable" except the inclusion of the simulation of the *Hash* query in the game G_2, the birthday paradox results lead to the following result:

$$|Adv^{AKM}_{\mathcal{A},G_1} - Adv^{AKM}_{\mathcal{A},G_2}| \leq \frac{q_h^2}{2|Hash|} \tag{3}$$

- **Game** G_3: It is the final game where the adversary $\mathcal{A}$ makes execution of the *CorruptSC* query. Thus, $\mathcal{A}$ will have the credentials $\{PID_i^*, C_i, Ex_i, a_i, h(\cdot), Gen(\cdot), Rep(\cdot), \tau_i, et\}$. Here, $SPW_i = h(PW_i\ ||b_i)$, $PID_i = h(ID_i\ ||b_i) = PID_i^* \oplus h(b_i\ ||PW_i\ ||ID_i)$, $p_i = h(PID_i\ ||b_i)$, $s_i = h(SPW_i\ ||x_i)$, $Ex_i = x_i \oplus h(PW_i\ ||ID_i\ ||b_i)$, $a_i = s_i \oplus h(ID_i\ ||b_i\ ||PW_i)$ and $C_i = s_i \oplus h(p_i\ ||PW_i\ ||ID_i)$. Now, to derive the secrets x_i and s_i from Ex_i, a_i and C_i, $\mathcal{A}$ needs the unknowns ID_i, PW_i and b_i. Hence, without the secret credentials s_i, ID_i and b_i of EVO_i, it becomes a "computationally difficult problem for $\mathcal{A}$ to guess password PW_i of EVO_i correctly with the help of the *Send* query defined in Table 2". Also, the probability of guessing the biometric key b_i of l_b bits by the adversary $\mathcal{A}$ is approximately $\frac{1}{2^{l_b}}$ [50]. It is worth noting that the games G_2 and G_3 are identical when the password/biometrics guessing attacks are not present. Hence, using the Zipf's law on passwords [47], we have the following result:

$$|Adv^{AKM}_{\mathcal{A},G_2} - Adv^{AKM}_{\mathcal{A},G_3}| \leq \max\left\{C' \cdot q_s^{s'}, \frac{q_s}{2^{l_b}}\right\} \tag{4}$$

where C' and s' are the Zipf's parameters [47]

As all the games are executed, $\mathcal{A}$ needs to guess the correct bit c. It follows that

$$Adv^{AKM}_{\mathcal{A},G_3} = \frac{1}{2} \tag{5}$$

Eqs. (1), (2) and (5), we have the following result:

$$\begin{aligned} \frac{1}{2}.Adv^{AKM}_{\mathcal{A}} &= |Adv^{AKM}_{\mathcal{A},G_0} - \frac{1}{2}| \\ &= |Adv^{AKM}_{\mathcal{A},G_1} - \frac{1}{2}| \\ &= |Adv^{AKM}_{\mathcal{A},G_1} - Adv^{AKM}_{\mathcal{A},G_3}| \end{aligned} \tag{6}$$

The triangular inequality and Eqs. (4), (5) and (6) lead to the following result:

$$\begin{aligned} \frac{1}{2}.Adv^{AKM}_{\mathcal{A}} &= |Adv^{AKM}_{\mathcal{A},G_1} - Adv^{AKM}_{\mathcal{A},G_3}| \\ &\leq |Adv^{AKM}_{\mathcal{A},G_1} - Adv^{AKM}_{\mathcal{A},G_2}| \\ &\quad + |Adv^{AKM}_{\mathcal{A},G_2} - Adv^{AKM}_{\mathcal{A},G_3}| \\ &\leq \frac{q_h^2}{2|Hash|} + \max\left\{C' \cdot q_s^{s'}, \frac{q_s}{2^{l_b}}\right\} \end{aligned} \tag{7}$$

Finally, multiplying both sides of Eq. (7) by a factor of 2, we have required result:

$$Adv^{AKM}_{\mathcal{A}} \leq \frac{q_h^2}{|Hash|} + 2\max\left\{C' \cdot q_s^{s'}, \frac{q_s}{2^{l_b}}\right\}.$$

TABLE 3. BAN logic notation.

Notation	Description
$P\mid\equiv X$	P **believes** formula X
#X	formula X is **fresh**
$P \lhd X$	P **see** formula X
$P\mid\sim X$	P once **said** X
$P \Rightarrow X$	P **controls** formula X
$< X >_Y$	formula X is **combined** with secret formula Y
$\{X\}_K$	formula X is **masked** by secret key K
$P \overset{K}{\leftrightarrow} Q$	P and Q use secret key K to communicate with each other
SK	Session key is used in the current session

B. MUTUAL AUTHENTICATION PROOF USING BAN LOGIC

We perform the BAN logic analysis to verify secure mutual authentication for the proposed protocol. Table 3 defines BAN logic postulates and notations, and we detail the goals, assumptions, and idealized forms before performing the BAN logic analysis confirming secure mutual authentication for the proposed protocol.

1) BAN LOGIC POSTULATES

The BAN logic postulates are given below.

- Message meaning rule:

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

- Nonce verification rule:

$$\frac{P \mid\equiv \#(X), \quad P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$$

- Jurisdiction rule:

$$\frac{P \mid\equiv Q \mid\Longrightarrow X, \quad P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$$

- Freshness rule:

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)}$$

- Belief rule:

$$\frac{P \mid\equiv (X, Y)}{P \mid\equiv X.}$$

2) GOALS AND ASSUMPTIONS

We make the following goals (G_1–G_4) and assumptions (A_1–A_8) to analyze the proposed protocol security.

G_1: $EV_i \mid\equiv AGT \mid\equiv (EV_i \overset{Skey}{\longleftrightarrow} AGT)$
G_2: $EV_i \mid\equiv (EV_i \overset{Skey}{\longleftrightarrow} AGT)$
G_3: $AGT \mid\equiv EV_i \mid\equiv (EV_i \overset{Skey}{\longleftrightarrow} AGT)$
G_4: $AGT \mid\equiv (EV_i \overset{Skey}{\longleftrightarrow} AGT)$
A_1: $AGT \mid\equiv (EV_i \overset{s_i}{\longleftrightarrow} AGT)$
A_2: $AGT \mid\equiv \#(k_i)$
A_3: $EV_i \mid\equiv (EV_i \overset{s_i}{\longleftrightarrow} AGT)$
A_4: $EV_i \mid\equiv \#(k_a)$
A_5: $AGT \mid\equiv \#(t_3)$
A_6: $AGT \mid\equiv EV_i \mid\Rightarrow (Skey)$
A_7: $EV_i \mid\equiv (EV_i \overset{s_i^{new}}{\longleftrightarrow} AGT)$
A_8: $EV_i \mid\equiv AGT \mid\Rightarrow (Skey)$

3) IDEALIZED FORMS

The idealized forms are as follows.

M_1: $EV_i \rightarrow AGT$: $(PID_i, x_i, k_i, t_1)_{s_i}$
M_2: $AGT \rightarrow EV_i$: $(AID, x_i, k_i, k_a, t_1, t_2)_{s_i}$
M_3: $EV_i \rightarrow AGT$:
$(PID_i, x_i, EV_i \overset{Skey}{\longleftrightarrow} AGT, SPW_i, t_3)_{s_i}$
M_4: $AGT \rightarrow EV_i$:
$(PID_i, EV_i \overset{Skey}{\longleftrightarrow} AGT, PID_i^{new}, t_4)_{s_i^{new}}$

4) BAN LOGIC PROOF

We employed BAN logic analysis to check that the proposed protocol achieves secure mutual authentication.

Step 1: We can obtain S_1 from M_1:

$$S_1 : AGT \lhd (PID_i, x_i, k_i, t_1)_{s_i}.$$

Step 2: We can obtain S_2 from the message meaning rule with S_1 and A_1:

$$S_2 : AGT \mid\equiv EV_i \mid\sim (PID_i, x_i, k_i, t_1)_{s_i}.$$

Step 3: We can obtain S_3 from the freshness rule with A_2:

$$S_3 : AGT \mid\equiv \#(PID_i, x_i, k_i, t_1)_{s_i}.$$

Step 4: We can obtain S_4 from the nonce verification rule with S_2 and S_3:

$$S_4 : AGT \mid\equiv EV_i \mid\equiv (PID_i, x_i, k_i, t_1)_{s_i}.$$

Step 5: We can obtain S_5 from M_2:

$$S_5 : EV_i \lhd (AID, x_i, k_i, k_a, t_1, t_2)_{s_i}.$$

Step 6: We can obtain S_6 from the message meaning rule with S_5 and A_3:

$$S_6 : EV_i \mid\equiv AGT \mid\sim (AID, x_i, k_i, k_a, t_1, t_2)_{s_i}.$$

Step 7: We can obtain S_7 from the freshness rule with A_4:

$$S_7 : EV_i \mid\equiv \#(AID, x_i, k_i, k_a, t_1, t_2)_{s_i}.$$

Step 8: We can obtain S_8 from the nonce verification rule with S_6 and S_7:

$$S_8 : EV_i \mid\equiv AGT \mid\equiv (AID, x_i, k_i, k_a, t_1, t_2)_{s_i}.$$

Step 9: We can obtain S_9 from M_3:

$$S_9 : AGT \lhd (PID_i, x_i, EV_i \overset{Skey}{\longleftrightarrow} AGT, SPW_i, t_3)_{s_i}.$$

Step 10: We can obtain S_{10} from the message meaning rule with S_9 and A_1:

$$S_{10} : AGT \mid\equiv EV_i \mid\sim (PID_i, x_i, EV_i \overset{Skey}{\longleftrightarrow} AGT, SPW_i, t_3)_{s_i}.$$

Step 11: We can obtain S_{11} from the freshness rule with A_5:

$$S_{11} : AGT \mid\equiv \#(PID_i, x_i, EV_i \overset{Skey}{\longleftrightarrow} AGT, SPW_i, t_3)_{s_i}.$$

Step 12: We can obtain S_{12} from the nonce verification rule with S_{10} and S_{11}:

$$S_{12} : AGT \mid\equiv EV_i \mid\equiv (PID_i, x_i, EV_i \overset{Skey}{\longleftrightarrow} AGT, SPW_i, t_3)_{s_i}.$$

Step 13: We can obtain S_{13} from the belief rule with S_{12}:

$$S_{13} : AGT \mid\equiv EV_i \mid\equiv (EV_i \overset{Skey}{\longleftrightarrow} AGT) \qquad (Goal \quad G_3)$$

```
role vehicle(EV, TTP, AGT : agent, SKevtt : symmetric_key, H: hash_func, SND, RCV : channel(dy))

played_by EV
def=
local State: nat,
   IDi, PWi, FPAi, Bi, Pi, SPWi, PIDi, EXi, Ai, Xi, Ci, Si: text,
   T1,T2,T3,T4, S1, S2, VERi, MKi, Ki, Skeya, Skeyi, Ui, Di : text,
   Ka, MVERi, KEAi, Sinew, PIDinew, AUTHi, Pinew, Cinew, Ainew,AID : text
const sp1, sp2, ev_agt_ki, ev_agt_ui, agt_ev_t2, agt_ev_authi : protocol_id
init State := 0
transition

%%%%%%%%%%%%Registration phase
1. State = 0 /\ RCV(start) =|>
State' := 1 /\ Bi' := new()
      /\ FPAi' := new()
      /\ SPWi' := H(PWi.Bi')
      /\ PIDi' := H(IDi.Bi')
      /\ Pi' := H(PIDi'.Bi')
      /\ SND({SPWi'.FPAi'.PIDi'}_SKevtt)
        /\ secret({IDi, Pi, PWi, Bi }, sp1, {EV})
      /\ secret({PIDi, SPWi, FPAi}, sp2, {EV,TTP})


%%%%%%%%%%%%Login & Authentication phase
2. State = 1 /\ RCV({H(IDi.Bi').Xi'.H(H(PWi.Bi').Xi')}_SKevtt) =|>
State' := 2 /\ Ki' := new()
      /\ T1' := new()
      /\ VERi' := H(H(H(PWi.Bi').Xi').Xi'.T1'.Ki')
      /\ MKi' := xor(Ki',H(H(H(PWi.Bi').Xi').T1'))
      /\ SND(H(IDi.Bi').VERi'.MKi'.T1')
      /\ witness(EV,AGT,ev_agt_ki, Ki')

3. State = 2
/\ RCV(H(H(H(H(PWi.Bi').Xi').Xi'.T1'.Ki').H(H(PWi.Bi').Xi').T1'.T2').xor(H(H(H(PWi.Bi').Xi').AID.T1'.T2'),Ka').T1'.T
2') =|>
State' := 3 /\ Skeyi' := H(Ka'.Ki'.H(H(PWi.Bi').Xi'))
      /\ T3' := new()
      /\ Ui' := H(H(H(PWi.Bi').Xi').Xi'.Skeyi'.T3')
      /\ Di' := xor(H(PWi.Bi'),H(H(PWi.Bi').T3'))
      /\ SND(H(IDi.Bi').Ui'.Di'.T3')
      /\ witness(EV, AGT, ev_agt_ui, T3')
      /\ request(AGT, EV, agt_ev_t2, T2')
4. State = 3
/\ RCV(xor(H(H(PWi.Bi').Xi'.H(Ka'.Ki'.H(H(PWi.Bi').Xi'))),H(H(H(IDi.Bi').H(H(PWi.Bi').Xi'.H(Ka'.Ki'.H(H(PWi.Bi').X
i'))).H(Ka'.Ki'.H(H(PWi.Bi').Xi'))).H(Ka'.Ki'.H(H(PWi.Bi').Xi')))).xor(H(H(IDi.Bi').H(H(PWi.Bi').Xi'.H(Ka'.Ki'.H(H(PW
i.Bi').Xi'))).H(Ka'.Ki'.H(H(PWi.Bi').Xi'))),H(H(H(PWi.Bi').Xi'.H(Ka'.Ki'.H(H(PWi.Bi').Xi'))).H(Ka'.Ki'.H(H(PWi.Bi').Xi'
)))).H(H(H(IDi.Bi').H(H(PWi.Bi').Xi'.H(Ka'.Ki'.H(H(PWi.Bi').Xi'))).H(Ka'.Ki'.H(H(PWi.Bi').Xi'))).H(H(PWi.Bi').Xi'.H(
Ka'.Ki'.H(H(PWi.Bi').Xi'))).T4').T4') =|>
State' := 4 /\ Pinew' := H(H(H(IDi.Bi').H(H(PWi.Bi').Xi'.H(Ka'.Ki'.H(H(PWi.Bi').Xi'))).H(Ka'.Ki'.H(H(PWi.Bi').Xi'))).Bi')
      /\ Cinew' := xor(H(H(PWi.Bi').Xi'.H(Ka'.Ki'.H(H(PWi.Bi').Xi'))), H(Pinew'.IDi))
      /\ Ainew' := xor(H(H(PWi.Bi').Xi'.H(Ka'.Ki'.H(H(PWi.Bi').Xi'))),H(H(PWi.Bi').IDi))
      /\ request(AGT, EV, agt_ev_authi, Xi')
end role
```

FIGURE 9. Role specification for ***EV***.

Step 14: We can obtain S_{14} from the jurisdiction rule with S_{13} and A_6:

$$S_{14} : AGT \mid\equiv (EV_i \overset{Skey}{\longleftrightarrow} AGT) \qquad (Goal \quad G_4)$$

Step 15: We can obtain S_{15} from M_4:

$$S_{15} : EV_i \lhd (PID_i, EV_i \overset{Skey}{\longleftrightarrow} AGT, PID_i^{new}, t_4)_{s_i^{new}}.$$

Step 16: We can obtain S_{16} from the message meaning rule with S_{15} and A_7:

$$S_{16} : EV_i \mid\equiv AGT \mid\sim (PID_i, EV_i \overset{Skey}{\longleftrightarrow} AGT, PID_i^{new}, t_4)_{s_i^{new}}.$$

Step 17: We can obtain S_{17} from the freshness rule with A_7:

$$S_{17} : EV_i \mid\equiv \#(PID_i, EV_i \overset{Skey}{\longleftrightarrow} AGT, PID_i^{new}, t_4)_{s_i^{new}}.$$

Step 18: We can obtain S_{18} from the nonce verification rule with S_{16} and S_{17}:

$$S_{18} : EV_i \mid\equiv AGT \mid\equiv (PID_i, EV_i \overset{Skey}{\longleftrightarrow} AGT, PID_i^{new}, t_4)_{s_i^{new}}.$$

Step 19: We can obtain S_{19} from the belief rule with S_{18}:

$$S_{19} : EV_i \mid\equiv AGT \mid\equiv (EV_i \overset{Skey}{\longleftrightarrow} AGT) \qquad (Goal \quad G_1)$$

```
role agg(EV, TTP, AGT : agent, SKevtt : symmetric_key, H: hash_func, SND, RCV : channel(dy))

played_by AGT
def=
local State: nat,
   IDi, PWi, FPAi, Bi, Pi, SPWi, PIDi, EXi, Ai, Xi, Ci, Si: text,
   T1,T2,T3,T4, S1, S2, VERi, MKi, Ki, Skeya, Skeyi, Ui, Di, AID : text,
   Ka, MVERi, KEAi, Sinew, PIDinew, AUTHi, Pinew, Cinew, Ainew : text
const sp1, sp2, ev_agt_ki, ev_agt_ui, agt_ev_t2, agt_ev_authi  : protocol_id
init State := 0
transition


1. State = 0 /\ RCV(H(IDi.Bi').H(H(H(PWi.Bi').Xi').Xi'.T1'.Ki').xor(Ki',H(H(H(PWi.Bi').Xi').T1'))) =|>
State' := 1 /\ Ka' := new()
      /\ Skeya' := H(Ka'.Ki'.H(H(PWi.Bi').Xi'))
      /\ T2' := new()
      /\ MVERi' := H(H(H(H(PWi.Bi').Xi').Xi'.T1'.Ki').H(H(PWi.Bi').Xi').T1'.T2')
      /\ KEAi' := xor(H(H(H(PWi.Bi').Xi').AID.T1'.T2'),Ka')
      /\ SND(MVERi'.KEAi'.T1'.T2')
      /\ witness(AGT, EV, agt_ev_t2, T2')
      /\ request(EV,AGT, ev_agt_ki, Ki')
2. State = 1
/\ RCV(H(IDi.Bi').H(H(H(PWi.Bi').Xi').Xi'.H(Ka'.Ki'.H(H(PWi.Bi').Xi')).T3').xor(H(PWi.Bi'),H(H(PWi.Bi').
T3')).T3') =|>
State' := 2 /\ Sinew' := H(H(PWi.Bi').Xi'.H(Ka'.Ki'.H(H(PWi.Bi').Xi')))
      /\ PIDinew' := H(H(IDi.Bi').Sinew'.H(Ka'.Ki'.H(H(PWi.Bi').Xi')))
      /\ T4' := new()
      /\ S1' := xor(Sinew',H(PIDinew'.H(Ka'.Ki'.H(H(PWi.Bi').Xi'))))
      /\ S2' := xor(PIDinew',H(Sinew'.H(Ka'.Ki'.H(H(PWi.Bi').Xi'))))
      /\ AUTHi':= H(PIDinew'.Sinew'.T4')
      /\ SND(S1'.S2'.AUTHi'.T4')
      /\ witness(AGT, EV, agt_ev_authi, Xi')
      /\ request(EV, AGT, ev_agt_ui, T3')

end role
```

FIGURE 10. Role specification for *AGT*.

Step 20: Finally, we can obtain S_{20} from the jurisdiction rule with S_{19} and A_8:

$$S_{20} : EV_i \mid\equiv (EV_i \overset{Skey}{\longleftrightarrow} AGT) \qquad (Goal \quad G_2)$$

Thus, the goals (G_1–G_4) prove that the proposed protocol ensures secure mutual authentication between EV_i and AGT.

C. FORMAL SECURITY VERIFICATION USING AVISPA TOOL: SIMULATION STUDY

This section implements simulations to evaluate the proposed protocol security using AVISPA [52], a widely adopted security analysis model, and prove that the protocol prevents replay and man-in-the-middle attacks [53]–[57].

The AVISPA tool checks if protocols are safe using High-Level Protocol Specification Language (HLPSL) [58], which has four backends: "On-the-fly ModelChecker (OMFC), Constraint Logic-based Attack Searcher (CL-AtSE), SAT-based Model Checker (SATMC), and Tree automata based on Automatic Approximations for Analysis of Security Protocol (TA4SP)". First, the HLPSL code is changed from the "Intermediate Format (IF)" and input to one of backends, and then IF is changed from the "Output format (OF)", which precisely presents security analysis results. Detailed information regarding HLPLS and AVISPA structure can be found elsewhere [52].

We included three basic roles in the AVISPA implementation for the proposed protocol: EV, TTP, and AGT; with two composition roles: goal & environment and session, representing participants and environment conditions, respectively, with detailed roles as shown in Figs. 9, 10, 11 and 12, respectively.

Figure 13 shows AVISPA analysis results under OFMC and CL-AtSE backends. OMFC and CL-AtSE prove that a legal entity can successfully perform the protocol by checking for a passive attacker. They also show that the protocol can prevent man-in-the-middle and replay attacks under the DY model. The OFMC backend took 1.29s search time with 130 visited nodes. The CL-AtSE backend analyzed three states in 0.09s translation time. Thus, the OFMC and CL-AtSE checks ensure the proposed protocol is secure against man-in-the-middle and replay attacks.

D. INFORMAL SECURITY ANALYSIS

We demonstrate that the proposed protocol is secure against various attacks, including impersonation, offline password guessing, man-in-the-middle, and trace attacks. We also show that the proposed protocol achieves anonymity, perfect forward secrecy, and secure mutual authentication and key agreement, based on the threat model defined in Section II-A.

```
role party(EV, TTP, AGT : agent, SKevtt : symmetric_key, H: hash_func, SND, RCV : channel(dy))

played_by TTP
def=
local State: nat,
   IDi, PWi, FPAi, Bi, Pi, SPWi, PIDi, EXi, Ai, Xi, Ci, Si: text,
   T1,T2,T3,T4, S1, S2, VERi, MKi, Ki, Skeya, Skeyi, Ui, Di : text,
   Ka, MVERi, KEAi, Sinew, PIDinew, AUTHi, Pinew, Cinew, Ainew : text
const sp1, sp2, ev_agt_ki, ev_agt_ui, agt_ev_t2, agt_ev_authi  : protocol_id
init State := 0
transition

1. State = 0 /\ RCV({H(PWi.Bi').FPAi'.H(IDi.Bi')}_SKevtt) =|>
State' := 1 /\ Xi' := new()
      /\ Si' := H(H(PWi.Bi').Xi')
      /\ SND({H(IDi.Bi').Xi'.Si'}_SKevtt)
      /\ secret({PWi, Bi}, sp1, {EV})
      /\ secret({IDi, Pi, SPWi, FPAi}, sp2, {EV,TTP})

end role
```

FIGURE 11. Role specification for ***TTP***.

```
%%%%Role for the session

role session(EV, TTP, AGT : agent, SKevtt : symmetric_key, H: hash_func)

def=
local SN1, SN2, SN3, RV1, RV2, RV3: channel(dy)
composition
vehicle(EV, TTP, AGT, SKevtt, H, SN1, RV1)
/\ party(EV, TTP, AGT, SKevtt, H, SN2, RV2)
/\ agg(EV, TTP, AGT, SKevtt, H, SN3, RV3)
end role

role environment()
def=
const ev, ttp, agt : agent,
skevtt: symmetric_key,
h : hash_func,
idi : text,
sp1, sp2, ev_agt_ki, ev_agt_ui, agt_ev_t2, agt_ev_authi: protocol_id

intruder_knowledge = {ev,ttp, agt,idi, h}
composition
session(ev, ttp, agt, skevtt, h)
/\session(i, ttp, agt, skevtt, h)
/\session(ev,i,agt, skevtt, h)
/\session(ev,ttp,i, skevtt, h)

end role

goal
secrecy_of sp1, sp2
authentication_on ev_agt_ki, ev_agt_ui
authentication_on agt_ev_t2, agt_ev_authi
end goal

environment()
```

FIGURE 12. Session and goal & environment.

1) IMPERSONATION ATTACK

We assume an attacker U_{at} can obtain the smart card of a legitimate user EVO_i and intercept messages transmitted in a session, and then try to impersonate EVO_i. However, U_{at} cannot generate the login request $\{PID_i, Ver_i, MK_i, t'_1\}$ and key confirmation request $\{PID_i, U_i, t'_3\}$ messages by generating current timestamps t'_1 and t'_3, because U_{at} would need to know secret parameters s_i, x_i and k_i, and session key $Skey_i$. Therefore, the proposed protocol prevents impersonation attack as U_{at} cannot correctly generate request messages.

2) OFFLINE PASSWORD GUESSING ATTACK

The proposed protocol prevents U_{at} from obtaining private parameters, including the password, because $a_i = s_i \oplus h(ID_i||b_i||PW_i)$, $Ex_i = x_i \oplus h(PW_i||ID_i||b_i)$, and $C_i = s_i \oplus h(p_i\ ||PW_i\ ||ID_i)$ are masked with a random secret number

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/EV.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 1.29s
 visitedNodes: 130 nodes
 depth: 6 plies
```

(a)

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/EV.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 3 states
 Reachable  : 0 states
 Translation: 0.09 seconds
 Computation: 0.00 seconds
```

(b)

FIGURE 13. Analysis results under OFMC and CL-AtSE backends. (a) OFMC. (b) CL-AtSe.

and secret parameters ID_i, PW_i and b_i. Therefore, U_{at} cannot guess EV_i's password correctly as he/she needs to guess ID_i and b_i simultaneously, which is computationally expensive task for the adversary U_{at} as explained in the threat model in Section II-A.

It is worth noting that three factors used in the proposed protocol are the smart card V_i, password PW_i and biometric BIO_i of a legal registered user U_i. For achieving the three-factor security, we assume that if two factors are compromised, U_{at} can not compromise (guess) third factor in the proposed protocol. For this purpose, assume that V_i and PW_i are compromised by U_{at}. Using the power analysis attacks (explained in the threat model in Section II-A) U_{at} will have the extracted information $\{PID_i^*, C_i, Ex_i, a_i, h(\cdot), Gen(\cdot), Rep(\cdot), et\}$ from the memory of V_i, where $Ex_i = x_i \oplus h(PW_i || ID_i || b_i)$, $a_i = s_i \oplus h(ID_i || b_i || PW_i)$, $C_i = s_i \oplus h(p_i || PW_i || ID_i)$, and $PID_i^* = PID_i \oplus h(b_i || PW_i || ID_i)$. To guess and validate correctly the biometric secret key b_i from Ex_i, a_i, C_i and PID_i^*, U_{at} needs guessing of both ID_i and b_i simultaneously, which is computationally expensive task for the adversary U_{at} as explained in the threat model in Section II-A. Similarly, if V_i and b_i are also compromised by U_{at}, to guess and validate correctly the password PW_i from Ex_i, a_i, C_i and PID_i^*, U_{at} also needs guessing of both ID_i and PW_i simultaneously, which is computationally expensive task for the adversary U_{at}. Hence, the offline guessing attacks are prevented in the proposed protocol.

3) MAN-IN-THE-MIDDLE ATTACK

Section VII-D.1 shows that U_{at} cannot generate request messages $\{PID_i, U_i, D_i, t_3\}$ and $\{PID_i, Ver_i, MK_i, t_1\}$, and also cannot generate valid $\{MVer_i, KE_{Ai}, t_1, t_2\}$, $\{S_1, S_2, Auth_i, t_4\}$ and $\{AuthVer_i, t_5\}$ without knowing secret parameters s_i, x_i and session key $Skey_a$ $(= Skey_i)$. Thus, the proposed protocol prevents man-in-the-middle attack.

4) REPLAY ATTACK

The proposed protocol prevents replay attack because all transmitted parameters are changed in every session. EV_i and AGT also check for valid timestamps using the conditions $Ver_i^* \stackrel{?}{=} Ver_i$, $MVer_i^* \stackrel{?}{=} Mver_i$, $U_i^* \stackrel{?}{=} U_i$, $Auth_i^* \stackrel{?}{=} Auth_i$ and $AuthVer_i^* \stackrel{?}{=} AuthVer_i$. Thus, the proposed protocol identifies and discards previous messages, forbidding replay attacks.

5) PRIVILEGED-INSIDER ATTACK

Suppose a privileged-insider user of the TTP, being an insider adversary $\mathcal{A}$, knows the registration information $\{SPW_i, FPA_i, PID_i\}$ of a legal user EVO_i during the registration process of the proposed protocol. Later, assume that $\mathcal{A}$ has lost or stolen smart card V_i of the same EVO_i after the registration process is done. Hence, using the power analysis attacks, $\mathcal{A}$ can extract all the stored information $\{PID_i, C_i, Ex_i, a_i, h(\cdot), Gen(\cdot), Rep(\cdot), et\}$ from the lost or stolen smart card V_i. However, without having the biometric secret key b_i of EVO_i, it is "computationally expensive" to guess correctly the password PW_i of EVO_i and then to validate it using SPW_i. Also, deriving secret credentials x_i and s_i is "computationally infeasible" as $\mathcal{A}$ requires to guess correctly ID_i, PW_i and b_i. As a result, the proposed protocol prevents privileged-insider attack.

6) DESYNCHRONIZATION ATTACK

In the key confirmation and pseudonym update phase of our protocol, we assume that the smart card V_i does not receive the response message $\{S_1, S_2, Auth_i, t_4\}$ from AGT because of unexpected termination or malicious attacks. However, an adversary cannot perform the desychronization attack because the protocol checks whether $Auth_i^* \stackrel{?}{=} Auth_i$. If it is not correct, the session is terminated. Furthermore, on successful validation EV_i sends the acknowledgment message $\{AuthVer_i, t_5\}$ to AGT. Only after successful validation of the received message, AGT will replace $\{s_i, PID_i\}$ with $\{s_i^{new}, PID_i^{new}\}$ in a secure database. In a similar way, EV_i will

TABLE 4. Security and functionality features comparison.

Security property	Wang *et al.* [18]	Abdallah and Shen [19]	Shen *et al.* [22]	Our
Impersonation attack	○	○	×	○
Perfect forward secrecy	○	×	×	○
Replay attack	○	○	○	○
Man-in-the-middle attack	○	○	×	○
Secure mutual authentication	○	○	×	○
Offline password guessing attack	–	–	×	○
De-synchronization attack	×	×	○	○
Privileged-insider attack	○	○	×	○
ESL attack under CK-adversary model	○	×	×	○
Formal security analysis under ROR model	×	×	×	○
Formal security verification under AVISPA tool	×	×	×	○

○: "a scheme is secure or it supports a functionality feature"; ×: "a scheme is insecure or it does not support a functionality feature".

also replace $\{PID_i^*, C_i, a_i\}$ with $\{PID_i^{**}, C_i^{new}, a_i^{new}\}$ in the smart card V_i. Therefore, the proposed protocol prevents desynchronization attack.

7) EPHEMERAL SECRET LEAKAGE (ESL) ATTACK

In the proposed protocol, EV_i and AGT establish the common session key as $Skey_a = h(k_a||k_i||s_i) = h(k_a^*||k_i||s_i^*) = Skey_i$. The session key is now dependent on both the "session-temporary (ephemeral or short term) secrets" k_i and k_a, and the long-term secret s_i. We consider the following two cases here:

- ***Case 1.*** Even if the "short term secrets k_i and k_a" are compromised through compromise of session states according to the CK-adversary model discussed in the threat model (Section II-A) to an adversary $\mathcal{A}$, it is "computationally difficult problem to derive the session key without the long-term secret s_i".
- ***Case 2.*** Even if the "long term secret s_i" is somehow compromised to $\mathcal{A}$, it is also "computationally difficult problem to derive the session key without the short-term secrets k_i and k_a".

Therefore, from the above two cases it is clear that the session key is only calculated if $\mathcal{A}$ can compromise both short & long term secret credentials. Since the session keys between any EV_i and AGT are distinct and unique, "a secret key leakage to $\mathcal{A}$ in a session does not lead to calculate other session keys in other sessions and it is also computationally infeasible problem due to application of both short & long term secrets in the session keys". Hence, the "session-temporary information attack is protected in the proposed protocol". Thus, the proposed protocol prevents "ESL attack".

8) TRACE ATTACK AND ANONYMITY

Sections VI-B and VI-C show that all transmitted messages ($\{PID_i, Ver_i, MK_i, t_1\}$, $\{MVer_i, KE_{Ai}, t_1, t_2\}$, $\{PID_i, U_i, D_i, t_3\}$, $\{S_1, S_2, Auth_i, t_4\}$, $\{AuthVer_i, t_5\}$) are changed in each session because they are masked with the timestamps t_1, t_2, t_3, t_4, and random numbers k_i, k_a. Pseudo-identity PID_i is also updated by AGT. Thus, the proposed protocol prevents trace attack and ensures anonymity.

9) PERFECT FORWARD SECRECY

Suppose the secret parameter s_i is compromised, and U_{at} wants to obtain the session key. However, since the proposed protocol updates s_i and PID_i in every session, U_{at} cannot obtain x_i and x_a. Thus, the proposed protocol guarantees perfect forward secrecy.

10) MUTUAL AUTHENTICATION AND KEY AGREEMENT

Upon receiving the login message $\{PID_i, Ver_i, MK_i, t_1\}$ from EV_i, AGT checks $Ver_i^* \stackrel{?}{=} Ver_i$. If it is valid, AGT authenticates EV_i. After receiving the response message $\{MVer_i, KE_{Ai}, t_1, t_2\}$ from AGT, EV_i also checks $MVer_i^* \stackrel{?}{=} MVer_i$ to authenticate AGT. Once both have confirmed each other, EV_i and AGT securely compute the session key. Thus, EV_i and AGT successfully authenticate each other and ensure key agreement.

VIII. PERFORMANCE ANALYSIS

This section evaluates the performance of the proposed protocol with regard to security & functionality features and computational cost, and then compares the outcomes with related protocols [18], [19], [22].

A. SECURITY AND FUNCTIONALITY FEATURES COMPARISON

We compare the security & functionality features of the proposed protocol with related protocols [18], [19], [22] as shown in Table 4. All previously proposed protocols cannot prevent various attacks, and also cannot guarantee perfect forward secrecy and secure mutual authentication. Thus, the proposed protocol provides superior security security & functionality features as compared with previous protocols.

B. COMPUTATIONAL COSTS COMPARISON

We compare computational overheads with related protocols [18], [19], [22] as shown in Table 5 during the

TABLE 5. Computation costs comparison.

Protocol	User	Server	Total overhead (in seconds)
Wang *et al.* [18]	$4T_{ecc} + 3T_m$	$2T_{ecc} + 1T_m$	$6T_{ecc} + 4T_m \approx 0.686166$ s
Abdallah and Shen [19]	algebraic operations	algebraic operations	≈ 0.01518 s
Shen *et al.* [22]	$4T_h$	$5T_h$	$9T_h \approx 0.00045$ s
Our	$T_{fe} + 11T_h$	$5T_h$	$T_{fe} + 16T_h \approx 0.06355$ s

T_h: one-way hash operation; T_{ecc}: elliptic curve point multiplication operation;
T_m: modular exponentiation operation; T_{fe}: fuzzy extractor function $Gen(\cdot)/Rep(\cdot)$ execution $\approx T_{ecm}$

login and authentication phase. Following the experimental results reported in [44] and [45], we define times T_m, T_{ecc}, T_{fe} and T_h as the number of modular exponentiation ($\approx$ 0.07231 s), elliptic curve point multiplication ($\approx$ 0.06305 s), fuzzy extractor function $Gen(\cdot)/Rep(\cdot)$ execution ($\approx$ T_{emc} $\approx$ 0.06305 s) [49] and one-way hash ($\approx$ 0.0005 s) operations, respectively.

The bitwise XOR operation is not included in this analysis because it is negligible as compared to other operations (T_m, T_{ecc}, T_{fe}, and T_h). Table 5 shows that the proposed protocol requires T_{fe} + $11T_h$ for each user (e.g., EV_i) and $5T_h$ for the server (e.g., AGT). This is a higher computational cost of the proposed protocol than Shen *et al.*'s protocol, but the proposed protocol guarantees significantly better improved security and functionality features. Thus, the proposed protocol is more secure than comparable previous protocols and can be applied to practical V2G environments.

IX. CONCLUDING REMARKS

This paper showed that Shen *et al.*'s protocol does not prevent impersonation, offline password guessing and privileged-insider attacks, and it does not ensure secure mutual authentication and perfect forward security. Consequently, we proposed a more secure dynamic privacy-preserving and lightweight key agreement protocol for V2G in SIoT to overcome the identified security flaws in Shen *et al.*'s protocol. The proposed protocol prevents impersonation, offline guessing, man-in-the-middle, replay, and trace attacks while also achieving perfect forward secrecy, anonymity, and secure mutual authentication because all transmitted parameters are dynamic in each session. We employed the BAN logic to prove that the proposed protocol provides secure mutual authentication between EV_i and AGT, the formal security analysis using the ROR model to prove that the proposed protocol provides the SK-security, and also implemented formal security verification simulation study using the AVISPA tool to demonstrate it was secure against replay and man-in-the-middle attacks. In addition, through the informal security analysis, we also showed that the proposed protocol can prevent other potential attacks. Furthermore, we performed the performance analysis of our protocol with related protocols. The proposed protocol was shown to be secure and more suitable for application to practical V2G systems.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers and the associate editor for their valuable feedback on the paper which helped us to improve its quality and presentation.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "SIoT: Giving a social structure to the Internet of Things," *IEEE Commun. Lett.*, vol. 15, no. 11, pp. 1193–1195, Nov. 2011.

[2] L. Atzor, A. Iera, G. Morabito, and M. Nitti, "The social Internet of Things (SIoT)—When social networks meet the Internet of Things: Concept, architecture and network characterization," *Comput. Netw.*, vol. 56, no. 16, pp. 3594–3608, 2012.

[3] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid," *IEEE Trans. Cloud Comput.*, vol. 3, no. 2, pp. 233–244, Apr./Jun. 2015.

[4] W. Kempton and J. Tomić, "Vehicle-to-grid power fundamentals: Calculating capacity and net revenue," *J. Power Sour.*, vol. 144, no. 1, pp. 268–279, Jun. 2005.

[5] C. Guille and G. Gross, "A conceptual framework for the vehicle-to-grid (V2G) implementation," *Energy Policy*, vol. 37, no. 11, pp. 4379–4390, 2009.

[6] S. Han, S. Han, and K. Sezaki, "Development of an optimal vehicle-to-grid aggregator for frequency regulation," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 65–72, Jun. 2010.

[7] C. Quinn, D. Zimmerle, and T. H. Bradley, "The effect of communication architecture on the availability, reliability, and economics of plug-in hybrid electric vehicle-to-grid ancillary services," *J. Power Sources*, vol. 195, no. 5, pp. 1500–1509, 2010.

[8] F. Kennel, D. Gorges, and S. Liu, "Energy management for smart grids with electric vehicles based on hierarchical MPC," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1528–1537, Aug. 2013.

[9] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, Nov. 2011.

[10] *Smart Grid Cyber Security Strategy and Requirements*, Standards NISTIR 7628, NIST, U.S. Dept. Commerce, Sep. 2014. Accessed: Apr. 2019. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf

[11] M. Stegelmann and D. Kesdogan, "Design and evaluation of a privacy-preserving architecture for vehicle-to-grid interaction," in *Public Key Infrastructures, Services and Applications* (Lecture Notes in Computer Science), vol. 7163. Leuven, Belgium: Springer, 2011, pp. 75–90.

[12] M. Stegelmann and D. Kesdogan, "Location privacy for vehicle-to-grid interaction through battery management," in *Proc. 9th Int. Conf. Inf. Technol. (ITNG)*, Las Vegas, NV, USA, Apr. 2012, pp. 373–378.

[13] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Enhancing location privacy for electric vehicles (at the right time)," in *Proc. Eur. Symp. Res. Comput. Secur.*, Pisa, Italy, 2012, pp. 397–414.

[14] H. Nicanfar, S. Hosseininezhad, P. TalebiFard, and V. C. M. Leung, "Robust privacy-preserving authentication scheme for communication between electric vehicle as power energy storage and power stations," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Turin, Italy, Apr. 2013, pp. 3429–3434.

[15] C. Rottondi, S. Fontana, and G. Verticale, ''Enabling privacy in vehicle-to-grid interactions for battery recharging,'' *Energies*, vol. 7, no. 5, pp. 2780–2798, 2014.

[16] Z. Yang, S. Yu, W. Lou, and C. Liu, ''P^2: Privacy-preserving communication and precise reward architecture for V2G networks in smart grid,'' *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 697–706, Dec. 2011.

[17] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, ''Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography,'' *Sensors*, vol. 14, no. 6, pp. 10081–10106, 2014.

[18] H. Wang, B. Qin, Q. Wu, L. Xu, and J. Domingo-Ferrer, ''TPP: Traceable privacy-preserving communication and precise reward for vehicle-to-grid networks in smart grids,'' *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2340–2351, Nov. 2015.

[19] A. Abdallah and X. S. Shen, ''Lightweight authentication and privacy-preserving scheme for V2G connections,'' *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2615–2629, Mar. 2017.

[20] H. Liu, H. Ning, Y. Zhang, and L. T. Yang, ''Aggregated-proofs based privacy-preserving authentication for V2G networks in the smart grid,'' *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1722–1733, Dec. 2012.

[21] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. S. Shen, ''A lightweight message authentication scheme for smart grid communications,'' *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 675–685, Dec. 2011.

[22] J. Shen, T. Zhou, F. Wei, X. Sun, and Y. Xiang, ''Privacy-preserving and lightweight key agreement protocol for V2G in the social Internet of Things,'' *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2526–2536, Aug. 2018.

[23] M. Burrows, M. Abadi, and R. Needham, ''A logic of authentication,'' *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.

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

[25] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, ''Examining smart-card security under the threat of power analysis attacks,'' *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[26] P. Kocher, J. Jaffe, and B. Jun, ''Differential power analysis,'' in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 1666. Santa Barbara, CA, USA: Springer, 1999, pp. 388–397.

[27] Y. Park, K. Park, and Y. Park, ''Secure user authentication scheme with novel server mutual verification for multiserver environments,'' *Int. J. Commun. Syst.*, vol. 32, no. 7, 2019, Art. no. e3929. doi: 10.1002/dac.3929.

[28] Y. Park, S. Lee, C. Kim, and Y. Park, ''Secure biometric-based authentication scheme with smart card revocation/reissue for wireless sensor networks,'' *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 7, pp. 1–11, 2016.

[29] R. Canetti and H. Krawczyk, ''Universally composable notions of key exchange and secure channels,'' in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Amsterdam, The Netherlands, 2002, pp. 337–351.

[30] A. Dua, N. Kumar, A. K. Das, and W. Susilo, ''Secure message communication protocol among vehicles in smart city,'' *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4359–4373, May 2018.

[31] V. Odelu, A. K. Das, M. Wazid, and M. Conti, ''Provably secure authenticated key agreement scheme for smart grid,'' *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1900–1910, May 2018.

[32] Y. Dodis, L. Reyzin, and A. Smith, ''Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,'' in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Interlaken, Switzerland, 2004, pp. 523–540.

[33] R. Amin, S. K. H. Islam, N. Kumar, and K.-K. R. Choo, ''An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks,'' *J. Netw. Comput. Appl.*, vol. 104, pp. 133–144, Feb. 2018.

[34] A. Burnett, F. Byrne, T. Dowling, and A. Duffy, ''A biometric identity based signature scheme,'' *Int. J. Netw. Secur.*, vol. 5, no. 3, pp. 317–326, 2007.

[35] J. H. Cheon, J. Jeong, D. Kim, and J. Lee, ''A reusable fuzzy extractor with practical storage size: Modifying Canetti *et al.*'s construction,'' in *Information Security and Privacy* (Lecture Notes in Computer Science), vol. 10946. Wollongong, NSW, Australia: Springer, 2018, pp. 28–44.

[36] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, ''Secure remote user authenticated key establishment protocol for smart home environment,'' *IEEE Trans. Dependable Secure Comput.*, to be published. doi: 10.1109/TDSC.2017.2764083.

[37] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, ''Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment,'' *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.

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

[39] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, ''Design of secure user authenticated key management protocol for generic IoT networks,'' *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, Feb. 2018.

[40] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, ''Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of drones deployment,'' *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3572–3584, Apr. 2019.

[41] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, ''Cloud centric authentication for wearable healthcare monitoring system,'' *IEEE Trans. Dependable Secure Comput.*, to be published. doi: 10.1109/TDSC.2018.2828306.

[42] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, ''Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things,'' *IEEE Trans. Dependable Secure Comput.*, to be published. doi: 10.1109/TDSC.2018.2857811.

[43] S. Jangirala, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, ''TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment,'' *IEEE Trans. Veh. Technol.*, to be published. doi: 10.1109/TVT.2019.2911672.

[44] Q. Xie, D. Hong, M. Bao, N. Dong, and D. S. Wong, ''Privacy-preserving mobile roaming authentication with security proof in global mobility networks,'' *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 5, 2014, Art. no. 325734.

[45] J.-S. Lee and C.-C. Chang, ''Secure communications for cluster-based ad hoc networks using node identities,'' *J. Netw. Comput. Appl.*, vol. 30, no. 4, pp. 1377–1396, 2007.

[46] M. Abdalla, P.-A. Fouque, and D. Pointcheval, ''Password-based authenticated key exchange in the three-party setting,'' in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 3386. Les Diablerets, Switzerland: Springer, 2005, pp. 65–84.

[47] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, ''Zipf's law in passwords,'' *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.

[48] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, ''Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment,'' *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 824–839, Sep./Oct. 2018.

[49] A. K. Das, M. Wazid, N. Kumar, M. K. Khan, K.-K. R. Choo, and Y. Park, ''Design of secure and lightweight authentication protocol for wearable devices environment,'' *IEEE J. Biomed. Health Inform.*, vol. 22, no. 4, pp. 1310–1322, Jul. 2018.

[50] V. Odelu, A. K. Das, and A. Goswami, ''A secure biometrics-based multi-server authentication protocol using smart cards,'' *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[51] AVISPA. (2019). *Automated Validation of Internet Security Protocols and Applications*. Accessed: Mar. 2019. [Online]. Available: http://www.avispa-project.org/

[52] AVISPA. *SPAN, the Security Protocol ANimator for AVISPA*. Accessed: Apr. 2019. [Online]. Available: http://www.avispa-project.org/

[53] K. Park, Y. Park, Y. Park, A. G. Reddy, and A. K. Das, ''Provably secure and efficient authentication protocol for roaming service in global mobility networks,'' *IEEE Access*, vol. 5, pp. 25110–25125, 2017.

[54] V. Odelu, A. K. Das, K.-K. R. Choo, N. Kumar, and Y. Park, ''Efficient and secure time-key based single sign-on authentication for mobile devices,'' *IEEE Access*, vol. 5, pp. 27707–27721, 2017.

[55] S. Yu, J. Lee, K. Lee, K. Park, and Y. Park, ''Secure authentication protocol for wireless sensor networks in vehicular communications,'' *Sensors*, vol. 18, no. 10, pp. 3191–3213, 2018.

[56] K. Park, Y. Park, Y. Park, and A. K. Das, ''2PAKEP: Provably secure and efficient two-party authenticated key exchange protocol for mobile environment,'' *IEEE Access*, vol. 6, pp. 30225–30241, 2018.

[57] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, N. Kumar, Y. Park, and S. Tanwar, ''Design of an anonymity-preserving group formation based authentication protocol in global mobility networks,'' *IEEE Access*, vol. 6, pp. 20673–20693, 2018.

[58] D. von Oheimb, ''The high-level protocol specification language HLPSL developed in the EU project AVISPA,'' in *Proc. 3rd APPSEM II Workshop Appl. Semantics (APPSEM)*, Frauenchiemsee, Germany, 2005, pp. 1–17.

KISUNG PARK received the B.S. and M.S. degrees in electronics engineering Kyungpook National University, Daegu, South Korea, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronics Engineering. His research interests include authentication, computer networks, the Internet of Things, post-quantum cryptography, VANET, and information security.

YOUNGHO PARK (M'17) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyungpook National University, Daegu, South Korea, in 1989, 1991, and 1995, respectively, where he is currently a Professor with the School of Electronics Engineering. From 1996 to 2008, he was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, South Korea. From 2003 to 2004, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, USA. His research interests include computer networks, multimedia, and information security.

ASHOK KUMAR DAS (M'17–SM'18) received the Ph.D. degree in computer science and engineering, the M.Tech. degree in computer science and data processing, and the M.Sc. degree in mathematics from IIT Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory, and Algorithmic Research, the International Institute of Information Technology, Hyderabad, India. He has authored over 185 papers in international journals and in his research areas, including 165 reputed journal papers. Some of his research findings are published in top cited journals such as the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SMART GRID, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, the IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS (formerly the IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE), the *IEEE Consumer Electronics Magazine*, IEEE ACCESS, the *IEEE Communications Magazine*, *Future Generation Computer Systems*, *Computers and Electrical Engineering*, *Computer Methods and Programs in Biomedicine*, *Computer Standards & Interfaces*, *Computer Networks*, *Expert Systems with Applications*, and the *Journal of Network and Computer Applications*. His current research interests include cryptography, wireless sensor network security, hierarchical access control, security in vehicular ad hoc networks, smart grids, the Internet of Things (IoT), cyber-physical systems (CPS) and cloud computing, and remote user authentication. He has served as a Program Committee Member of many international conferences. He has also served as one of the Technical Program Committee Chair of the International Congress on Blockchain and Applications (BLOCKCHAIN 2019), Avila, Spain, in 2019. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the Editorial Board of *KSII Transactions on Internet and Information Systems*, the *International Journal of Internet Technology and Secured Transactions* (Inderscience), and *IET Communications*. He is a Guest Editor of *Computers and Electrical Engineering* (Elsevier) for the special issue on Big data and IoT in e-healthcare and *ICT Express* (Elsevier) for the special issue on Blockchain Technologies and Applications for 5G Enabled IoT.

SUNGJIN YU received the B.S. degree in electronics engineering, Daegu University, Gyeongsangbuk-do, South Korea, in 2017, and the M.S. degree in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2019. His research interests include authentication, the Internet of Things, post-quantum cryptography, blockchain, and information security.

JOONYOUNG LEE received the B.S. degree in electronics engineering, Kyungpook National University, Daegu, South Korea, in 2017, where he is currently pursuing the M.S. degree with the School of Electronics Engineering. His research interests include authentication, the Internet of Things, blockchain, and information security.

YOHAN PARK received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyungpook National University, Daegu, South Korea, in 2006, 2008, and 2013, respectively. He is currently an Assistant Professor with the Information and Communication Department, Division of IT Convergence, Korea Nazarene University. His research interests include computer networks, mobile security, and information security.