

PUF-Based Robust and Anonymous Authentication and Key Establishment Scheme for V2G Networks

Sungjin Yu and Kisung Park

Abstract—Vehicle-to-grid (V2G) improves flexibility, reliability, and efficiency and ensures effective charging services by enabling two-way communication along with two-way electricity transmission between the power grid and electric vehicle (EV). However, V2G networks are fragile to lethal security threats because an attacker may try to compromise and control the communication participants at any time. Recently, Sureshkumar et al. presented a robust and lightweight authentication key establishment (AKE) for secure V2G networks to provide essential security properties. However, we prove that their scheme suffered from various security threats and lacked essential security properties. To protect against physical security attacks, a promising solution is the use of physical unclonable function (PUF) technology and many AKE schemes have been designed for V2G networks. However, these schemes are still fragile to machine learning (ML)-based modeling attacks as well as existing security threats. Thus, we design a physical unclonable function (PUF)-based robust and anonymous AKE scheme for V2G networks, called R2AKEV2G to resist ML-based modeling attacks. We prove the security of R2AKEV2G by performing formal security analyses. Moreover, we perform a network simulator (NS) 3 implementation in compliance with IEEE 802.11 to prove its feasibility and verify that R2AKE-V2G is suitable for practical V2G networks. Consequently, R2AKE-V2G supports better security features and functionalities attributes and also guarantees superior costs with regard to communication and computation as compared to existing relevant schemes.

Index Terms—Physical unclonable function (PUF), vehicle-to-grid (V2G) networks, authentication, key establishment

I. INTRODUCTION

WITH the development of “5G, smart grid (SG), and electric vehicle (EV)” technology, the vehicle-to-grid (V2G) is emerging as an attractive new network paradigm and also it has garnered considerable interest from both scientific and industrial communities [1]–[3]. The V2G allows bidirectional energy communication between EV and power grid and mitigates environmental pollution, and also helps overcome the energy crisis. The V2G not only encourages citizens to switch to eco-friendly plug-in hybrid electric vehicles (PHEVs) and EVs but also actively supports load management on the power grid and offers new economic benefits in charging interoperability scenarios [4]. Owing to the V2G, the electrical energy can flow from the smart grid to the EV to charge the

battery and also can flow in the reserve direction to provide surplus and peak power. In addition, an individual owner or single household may engage in trading to purchase and sell energy from their EVs using V2G technology without building formal power generation and distribution systems. However, despite the multiple benefits and advantages of V2G, there are significant difficulties and challenges to be addressed. Since the V2G communication among an electrical vehicle user, utility service provider, and charging station occurs without any encryption or authentication, a malicious attacker can attempt to forge, modify, eavesdrop, and delete the user’s individual data for V2G (i.g. locations, payment records, and battery status) [5]. Moreover, a malicious attacker can steal a smart device of a legitimate user, he/she then extracts the user’s sensitive data stored in the smart device by using differential power analysis [6]. If the sensitive data of the legitimate user is revealed, a malicious attacker may attempt lethal cyber attacks like “forgery, insider, and offline password guessing” attacks. Moreover, physical security is also essential because charging stations and EVs are not normally guarded by humans. Due to these physical and cyber security attacks, a malicious attacker may insert new consumption data and report the wrong energy charging data into the smart devices during charging and discharging processes and then lead to a waste of resources and impose financial charges on the users for electric energy which has not been used [7]. With the escalating need for energy services and applications in V2G networks, another significant challenge is its lightweight feature. Since the smart devices for V2G (e.g., internet of energy things (IoET), smart meters, smart card, etc.) have resource-constrained with respect to computation and communication overheads, memory, and computing power [8], it is not suitable to use public key cryptosystems that require high performance. Hence, “lightweight and robust authentication and key establishment (AKE) schemes” are indispensable for V2G networks [9]–[11].

Sureshkumar et al. [12] recently designed a robust AKE scheme with privacy-preserving for V2G networks to ensure reliable energy services. Sureshkumar et al. claimed that their AKE protocol offers “necessary security requirements” while preventing lethal physical/cyber attacks. However, we indicate that their scheme [12] lacks the ability to withstand severe security attacks like “session key disclosure and impersonation” attacks and lacks “mutual authentication”. To enhance these issues, a promising solution is the use of physical unclonable function (PUF) technology.

A PUF presents to address these issues by allowing smart devices to create secure and unique digital fingerprints with

S. Yu is with the “Electronics and Telecommunications Research Institute, Daejeon, 34129, South Korea” (E-mail: sj.yu@etri.re.kr).

K. Park is with the “Department of Computer Engineering (Smart Security), Gachon University, Seongnam, 13120, South Korea” (E-mail: kisung-ing@gmail.com). (Corresponding Author: Kisung Park)

This work was supported by Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT). (No. 2022-0-01019, Development of eSIM security platform technology for edge devices to expand the eSIM ecosystem)

extremely low computation overheads. Recently, many AKA schemes have been designed for V2G networks using PUF to resist cyber/physical security attacks. Although there are some PUF-based schemes for V2G networks have been presented, these schemes are still susceptible to various security attacks because other security issues have remained unresolved. Moreover, existing PUF utilized in AKE schemes is vulnerable to machine learning (ML)-based modeling attacks since an adversary can clone the PUF model by having access to a subset for the challenge-response pairs (CRP) of the PUF through a public channel. Therefore, we design a PUF-based robust and anonymous AKE scheme, called R2AKE-V2G to enhance machine learning (ML)-based modeling attacks as well as various existing security issues.

A. Motivations

The primary objective of this paper is to show and enhance the security shortcomings of [12]. We demonstrated that scheme [12] is vulnerable to deadly security attacks like “session key disclosure” and “impersonation” attacks and it also lacks “mutual authentication”. Sureshkumar et al.’s scheme [12] put in a tremendous amount of effort to develop a high-level security-supported system for V2G networks. Regrettably, their scheme did not approach AKE protocol from the perspective that we have verified and demonstrated. These discoveries have motivated us to develop a new AKE scheme that is robust and anonymous based on PUF and capable of resisting “potential security attacks” that are present in V2G networks as well as ensuring “necessary security functionalities”.

B. Contributions

This section serves to introduce the primary contribution of R2AKE-V2G.

- We design a “new PUF-based robust and anonymous AKE scheme for V2G networks” to enhance the security drawbacks of [12].
- We present “Automated Verification of Internet Security Protocols and Applications” (AVISPA) [13], [14] which assesses the robustness against potential security attacks like MITM and replay attacks. Moreover, we present “Real-or-Random (ROR) oracle” model [15] which proves the session key security of the proposed scheme.
- We present a performance analysis with regard to computational and communication costs and security functionalities compared to related AKE schemes.
- We present the implementation for performance analysis using the network simulator (NS) 3 [16] on various network scenarios and attributes.

II. RELATED WORKS

Secure and reliable communication is one of the most important necessary requirements for V2G networks to provide secure data exchange and sharing. Thus, a robust and anonymous authentication and key establishment scheme is essential for secure and efficient data exchange between components.

To resolve these problems, many AKE schemes have been presented for V2G networks [17]–[19] to provide secure and efficient data exchange between each entity. Mohammadali et al. [20] designed two protocol scenarios for smart grid networks: elliptic curve cryptosystems (ECC)-based AKE scheme and identity-based AKE scheme. These AKE protocols are resistant to desynchronization and replay attacks and also reduce the computation cost with regard to the smart meter. However, these AKE protocols are fragile to MITM, false data injection, and masquerade attacks. Nicanfar and Leung [21] proposed two protocol scenarios to provide scalability and security for data exchange in smart grid systems: symmetric key-based AKE scheme and ECC-based AKE scheme. Unfortunately, their scheme is insecure to false data injection attacks and also has high computation cost during AKE phase. Wu and Zhou [22] designed a secure and lightweight AKE protocol for smart grid networks by combining public key and symmetric key cryptosystems. However, Xia and Wang [23] pointed out that Wu and Zhou’s scheme [22] cannot prevent MITM attacks and they presented a new secure key distribution scheme for smart grid networks. Unfortunately, Park et al. [24] demonstrated that Xia and Wang’s scheme [23] is still vulnerable to forgery attacks and does not protect the privacy of users. Tsai and Lo [25] designed a secure key distribution scheme for V2G networks by using identity-based encryption and signature. Odelu et al. [26] proved that Tsai and Lo’s scheme [25] does not ensure the session key security and also privacy of the smart meters. However, Gope and Sikdar [27] pointed out that the AKE scheme proposed in [26] is fragile to MITM attacks ultimately leading to DoS attacks.

In the last few years, many AKE research articles have been presented on privacy issues for V2G networks [28], [29] besides Ref. [20]–[27]. However, these AKE schemes have inefficient performance because they use cryptographic primitives such as sign encryption and group signature operations that require a high computation cost, and also the problem of privacy concerns for electrical vehicle users remained unresolved. In this context, Gope and Sikdar designed a cost-effective privacy-preserving AKE scheme for V2G networks [27]. However, Irshad et al. [30] demonstrated that Gope and Sikdar’s scheme [27] has a desynchronization problem during login to the device and also is fragile to key compromise impersonation attacks through the feeble assumptions, in which the private secret key is revealed by mistake to the attacker. Irshad et al. [30] proposed a secure and lightweight AKE scheme for V2G networks to enhance the security drawbacks of Gope and Sikdar’s scheme [27].

Recently, Sureshkumar et al. [12] designed a robust AKE scheme for V2G networks to provide high security and privacy. They claimed that their AKE scheme [12] guarantees necessary security requirements, and also is resistant to lethal security attacks. However, we proved that scheme [12] is fragile to deadly security threats such as “session key disclosure and impersonation” attacks due to wrong protocol design and it lacks “mutual authentication”. Thus, we propose a “PUF-based robust and anonymous AKE scheme for V2G networks” to address the security shortcomings of [12]. The proposed scheme generates a unique temporary key based on the PUF

TABLE I: Existing Authentication and Key Establishment Schemes for V2G Networks: A Comparative Summary

Scheme	Year	Cryptographic Primitives	Advantages/Description	Shortcomings/Limitations
Wu and Zhou [22]	2011	*Elliptic curve cryptography *Symmetric key encryption *One-way hash function	*Fault-tolerant and scalable key management for V2G *Provide a high-level of fault tolerance and scalability	*Not secured against man-in-the-middle (MITM) attack [23] *Does not provide session key security [27]
Xia and Wang [23]	2012	*Symmetric key encryption *One-way hash function	*Secure and efficient key distribution scheme for V2G networks *Provide high-level security as well as effective efficiency *Low computation cost	*Not secured against forgery attack [24] *Does not protect the privacy of user [24]
Tsai and Lo [25]	2016	*Bilinear pairing *Multiplication point *Modular exponential *One-way hash function	*Secure anonymous key distribution scheme for V2G using an identity-based signature/encryption mechanisms *Provide anonymity and data confidentiality	*Does not ensure session key security [26] *Does not ensure privacy of the smart meter [26]
Odelu <i>et al.</i> [26]	2018	*Bilinear Maps *Identity-based encryption *One-way hash function	*Efficient provably robust authenticated key agreement scheme for V2G networks *Provide session key security and strong credentials' privacy *Low computation cost	*Not secured against MITM attack [27] *Not secured against denial of service (DoS) attack [27]
Gope and Sikdar [27]	2019	*One-way hash function	*Efficient privacy-preserving authentication scheme for energy internet-based V2G *Provide lightweight computation and communication costs	*Has a desynchronization problem during login to the device [30] *Not secured against key compromise impersonation attack [30]
Kaveh <i>et al.</i> [31]	2020	*One-way hash function *Physical unclonable function	*Secure and Robust AKE scheme for SG neighborhood area networks *Provide high-level security *Low computation cost	*Not secured against smart meter impersonation attack [32] *Not secured SG server impersonation attack [32]
Bansal <i>et al.</i> [33]	2020	*One-way hash function *Physical unclonable function	*Lightweight AKE protocol for V2G networks using PUF *Provide lightweight computation cost and energy efficient	*Not secured against privilege insider and physical attacks [34] *Does not guarantee user anonymity and untraceability [34]
Sureshkumar <i>et al.</i> [12]	2022	*One-way hash function	*Robust and lightweight authenticated and key agreement scheme for V2G networks *Provide high-level security *Low computation cost	*Not secured against session key disclosure attack *Not secured impersonation attack *Does not ensure mutual authentication

and then utilizes it to use symmetric key encryption to not only ensure high-level security in the current session but also establish secure V2G communication. Although there are some PUF-based schemes for V2G networks have been proposed [31]–[34] to resolve physical capture attacks, these schemes [31], [33] are still susceptible to various security attacks because other security issues have remained unresolved. Due to this fact, it is very difficult to design cryptographic protocols to satisfy all necessary security requirements.

A comparative summary of existing AKE schemes for V2G networks is presented in Table I.

III. PRELIMINARIES

A. Adversary Model

We introduce the “Dolev-Yao (DY) model” [35] and “Canetti and Krawczyk (CK) model” [36]. The adversary capabilities in cryptographic protocol are as below:

- In the DY and CK models, an adversary (\mathcal{A}) can “resend, delete, block, eavesdrop, and so on” the transmitted data under an insecure channel and also can injure the session states with ephemeral secret value.
- \mathcal{A} can steal a smart card (SC) of the user and then extract the information stored in SC by using “differential power analysis” [6].
- \mathcal{A} may attempt lethal security attacks including “stolen verifier, offline guessing, and privileged insider” attacks [37].

B. Physical Unclonable Function

PUF is widely recognized as a practical solution safeguarding the security of smart devices with limited computing capabilities from potential adversarial threats [38], [39]. PUF is a widely utilized technique for producing an output based on a given input such as a fingerprint which is derived from the physical microstructure of smart devices. PUF does not retain a private key and poses a considerable challenge in the successful replication of an identical PUF. This is due to the intricate nanoscale variations during the manufacturing process of the IC chip. The optimal PUF ensures the properties of unpredictability, uniqueness, and reliability, all of which are critical components for protecting the security of smart devices. PUF is particularly effective in protecting the smart devices that are deployed in WMSN-based healthcare systems from attacks such as cloning, side-channel, and tampering attacks. PUF is reliant upon the distinct physical attributes of the integrated circuit, and any alteration to the system shall undoubtedly result in a modification of the PUF output. In addition, PUF allows for the verification of the legitimacy of entities prior to the establishment of a session key, as has been demonstrated in previous research [40]. The functionalities of the PUF are as follows:

- “PUF is quite simple to implement and assess”.
- “PUF depends on physical microstructure of system”.
- “Any attempt to interfere with smart devices that have PUF will update of PUF’s behavior and consequently its destruction [41]”.

As depicted in Fig. 1, a PUF-enabled generator procedure utilizes multiple functions, including “decoding, encoding, and key derivation” to produce powerful extractors for secret key. These functions combine to make an optimal solution for “robust authentication of lightweight devices in V2G network”.

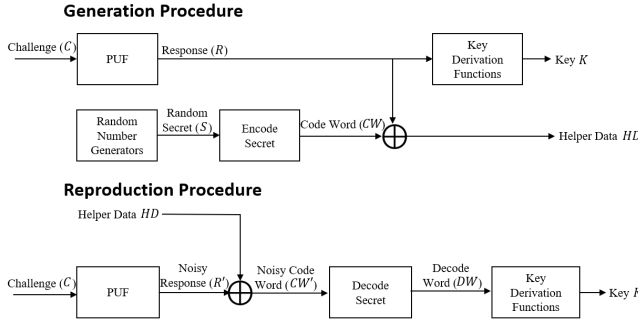


Fig. 1: PUF Key Generator Mechanism

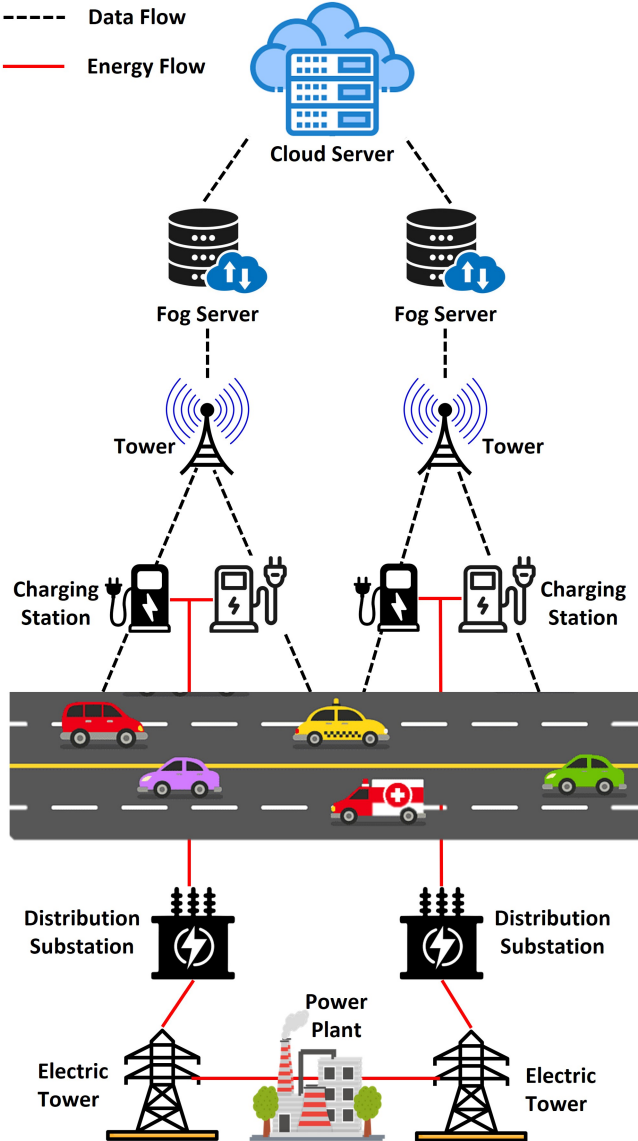


Fig. 2: System Model for V2G Networks

C. System Model

This section presents the system model for V2G network communication in Fig. 2. The system model has consisted of the “utility service provider (USP), smart electric vehicle (SEV), cloud, server (CS), and fog server (FS)”. This model is possible for different levels of communications including “vehicle-to-charging station (V2C), vehicle-to-vehicle (V2V), and charging station-to-utility service provider (C2U)”. In this model, an anonymous, lightweight, and robust AKE scheme is proposed to ensure effective and secure communication for V2G networks. An ordinary server can only process data from one vehicle at a time. Moreover, there is a need for a CS to perform parallel processing. In the system model, the FS controls and monitors the CS and vehicle in real-time. If the vehicles move out of the smart city, the FS transmits a message to the CS to connect to another FS. Therefore, our system model is considered a good solution for a secure, effective, robust, and anonymous AKE scheme in V2G environments.

TABLE II: Symbols

Symbol	Meaning
U_i	i^{th} electrical vehicle user
USP	Utility service provider
CS	Charging station
ID_U, ID_{CS}, ID_{USP}	Identity of U_i , CS , and USP
PW_i	Password of U_i
BIO	Biometric of U_i
C_U^x, R_U^x	Challenge/response of U_i
C_{CS}^x, R_{CS}^x	Challenge/response of CS
ΔT_i	Acceptable time delay
T_i	Timestamp
MK_{CS}, MK_{USP}	Master key of CS and USP
SK	A session key among U_i , CS , and USP
$E_K(\cdot)/D_K(\cdot)$	Symmetric key encryption/decryption
$h(\cdot)$	Hash function
$H(\cdot)$	Bio-hash function
\oplus	XOR function
\parallel	Concatenation

IV. REVIEW OF SURESHKUMAR ET AL.’S SCHEME

We introduce the reviews for Sureshkumar et al.’s scheme [12]. Table II is presented the symbols used in this paper.

A. Initial Setup Phase

USP selects a master private key MK_{USP} and comprises an “one-way hash function $h : (0, 1)^* \rightarrow (0, 1)^n$ and bio-hash function $H(\cdot)$ ”. USP publishes the “bio-hash function $H(\cdot)$ and the one-way hash function $h(\cdot)$ ” as public details.

B. User Registration Phase

U_i duly register with USP and acquires specific confidential credentials from USP .

URP-1: U_i generates a identity ID_U , a biometric BIO , and password PW_i and calculates $b_i = H(BIO)$. Then, U_i computes $A_1 = h(ID_U)$ and $A_2 = h(PW_i || b_i)$ and transmits $\{A_1, A_2\}$ to USP through a secure channel.

URP-2: USP calculates $S_i = h(A_1 || MK_{USP})$, $B_i = A_2 \oplus S_i$, $C_i = h(A_2 || B_i)$, $D_i = B_i \oplus C_i$, and

$E_i = h(S_i|C_i|D_i)$. After that, *USP* builds a $SC = (B_i, E_i, D_i)$ and transmits it to U_i .

URP-3: After receiving the SC , U_i computes $F_i = B_i \oplus A_2 \oplus A_1$ so that $F_i = A_1 \oplus S_i$ and $G_i = A_1 \oplus (PW_i|h_1(b_i))$ where h_1 is an one-way hash function whose output concatenated to PW_i results in the size of the output of $h(\cdot)$. Finally, U_i computes $K_i = h(A_1|b_i)$ and reconstructs $SC = (F_i, G_i, E_i, K_i)$ where G_i and K_i are included only for the contribution of password recovery functionality.

C. Charging Station Registration Phase

CS generates a ID_{CS} and then sends it to the *USP*. After that, *USP* calculates $c_j = h(ID_{CS}|MK_{USP})$ and sends it to the *CS* via a secure channel. Finally, *USP* removes the parameter c_j in the system. *CS* keeps the c_j securely.

D. Authentication and Key Establishment Phase

The registered U_i is required to establish a mutually authenticated session key SK in order to access reliable V2G services.

AKE-1: U_i inputs ID_U , PW_i and imprints BIO in *SC*. After that, *SC* calculates $b_i = H(BIO)$, $A_1 = h(ID_U)$, and $A_2 = h(PW_i|b_i)$. Then, *SC* computes $S_i^* = F_i \oplus A_1$, $B_i^* = A_2 \oplus S_i^*$, $C_i^* = h(A_2|B_i^*)$, $D_i^* = B_i^* \oplus C_i^*$, and $E_i^* = h(S_i^*|C_i^*|D_i^*)$. After that, *SC* verifies whether $E_i^* \stackrel{?}{=} E_i$. If it matches, *SC* accepts U_i , otherwise; terminates and rejects the current session.

AKE-2: *SC* generates a random nonce R_1 and calculates $L_1 = h(A_1|R_1)$, $L_2 = L_1 \oplus S_i$, $Auth_U = h(L_1|L_2|T_1)$, $W_1 = L_1 \oplus C_i^* \oplus L_2$, and $W_2 = L_1 \oplus A_1 \oplus C_i^*$. Then, *SC* transmits $M_1 = \{W_1, W_2, L_2, Auth_U, T_1\}$ to the *CS* via an insecure channel.

AKE-3: *CS* checks the $|T_2 - T_1| \leq \Delta T_i$. If the timestamp is matches, *CS* generates a R_2 and then calculates $L_3 = h(ID_{CS}|R_2)$, $L_4 = L_3 \oplus c_j$, and $Auth_{CS} = h(L_3|L_4|T_2)$. *CS* transmits $M_2 = \{ID_{CS}, L_4, Auth_{CS}, T_2, W_1, W_2, L_2, Auth_u, T_1\}$ to the *USP*.

AKE-4: *USP* check the freshness of $|T_2 - T_3| \leq \Delta T_i$. If it is correct, *USP* computes $c_j = h(ID_{CS}|MK_{USP})$, $L_3^* = L_4 \oplus c_j$, and $Auth_{CS}^* = h(L_3^*|L_4|T_2)$, and verifies $Auth_{CS}^* \stackrel{?}{=} Auth_{CS}$. If it matches, *USP* calculates $A_1^* \oplus L_2^* = W_1 \oplus W_2$, $A_1^* = A_1^* \oplus L_2^* \oplus L_2$, $S_i^* = h(A_1^*|MK_{USP})$, $L_1^* = L_2 \oplus S_i^*$, and $Auth_U^* = h(L_1^*|L_2|T_1)$.

AKE-5: *USP* verifies $Auth_U^* \stackrel{?}{=} Auth_U$. If it matches, *USP* selects a random nonce R_3 and computes $L_5 = h(T_1|T_2|T_3|R_3)$, $N_{u_1} = L_3^* \oplus h(L_1^*|S_i)$, $N_{u_2} = L_5 \oplus h(L_1^*|S_i)$, and $N_{CS_1} = L_1^* \oplus h(L_3^*|c_j)$, $N_{CS_2} = L_5 \oplus h(L_3^*|c_j)$, $NAuth_{CS} = h(N_{CS_1}|N_{CS_2}|c_j|L_3^*)$, and $NAuth_U = h(N_{u_1}|N_{u_2}|S_i|L_1^*)$. Finally, *USP* transmits $M_3 =$

$\{N_{CS_1}, N_{CS_2}, NAuth_{CS}, N_{u_1}, N_{u_2}, NAuth_U\}$ to the *CS*.

AKE-6: *CS* computes $NAuth_{CS}^* = h(N_{CS_1}|N_{CS_2}|c_j|L_3)$ and verifies $NAuth_{CS}^* \stackrel{?}{=} NAuth_{CS}$. If it is equal, *CS* authenticates *USP* and then computes $L_1^* = N_{CS_1} \oplus h(L_3|c_j)$ and $L_5^* = N_{CS_2} \oplus h(L_3|c_j)$. Finally, *CS* transmits $M_4 = \{N_{u_1}, N_{u_2}, NAuth_U\}$ to the U_i .

AKE-7: U_i calculates $NAuth_U^* = h(N_{u_1}|N_{u_2}|S_i|L_1)$ and checks $NAuth_U^* \stackrel{?}{=} NAuth_U$. If it is correct, U_i computes $L_3^* = N_{u_1} \oplus h(L_1|S_i)$ and $L_5^* = N_{u_2} \oplus h(L_1|S_i)$.

Consequently, U_i , *USP*, and *CS* are mutually authenticated and successfully establish a $SK = h(L_1|L_3|L_5)$.

E. Password Update Phase

If U_i wants to change a new PW_i , U_i may update their previous PW_i without requiring interaction with the *USP*.

PUP-1: U_i inputs an ID_i , an old PW_i^{old} , and imprints a biometric BIO in the *SC*. The *SC* computes $b_i = H(BIO)$, $A_1 = h(ID_U)$, and $A_2 = h(PW_i|b_i)$. Moreover, *SC* computes $S_i^* = F_i \oplus A_1$, $B_i^* = A_2 \oplus S_i^*$, $C_i^* = h(A_2|B_i^*)$, $D_i^* = B_i^* \oplus C_i^*$, and $E_i^* = h(S_i^*|C_i^*|D_i^*)$. The *SC* checks $E_i^* \stackrel{?}{=} E_i$. If it is equal, the *SC* accepts U_i , otherwise; terminates and rejects the current session.

PUP-2: After the successful validation of the U_i , *SC* enters a new password PW_i^{new} . *SC* computes $A_2^{new} = h(PW_i^{new}|b_i)$, $B_i^{new} = A_2^{new} \oplus S_i$, $C_i^{new} = h(A_2^{new}|B_i^{new})$, $D_i^{new} = B_i^{new} \oplus C_i^{new}$, $E_i^{new} = h(S_i|C_i^{new}|D_i^{new})$, and $G_i^{new} = A_1 \oplus (PW_i^{new}|h_1(b_i))$. Finally, *SC* is updated as $SC = (F_i^{new}, F_i, K_i, G_i^{new})$.

V. SECURITY FLAWS OF SURESHKUMAR ET AL.'S SCHEME

We serve to show the security vulnerabilities inherent in the scheme presented by Sureshkumar et al.'s scheme [12].

A. Session Key Disclosure Attack

Referring to Section III-A, \mathcal{A} can extract the secret credentials $\{F_i, E_i, G_i, K_i\}$ stored in *SC*. In addition, \mathcal{A} can "delete, block, and replay" the transmitted messages through an insecure channel. \mathcal{A} first computes $A_1 = W_1 \oplus W_2 \oplus L_2$, $S_i = A_1 \oplus F_i$, $L_1 = L_2 \oplus S_i$. After that, $L_3 = N_{u_1} \oplus h(L_1^*|S_i)$ and $L_5 = N_{u_2} \oplus h(L_1^*|S_i)$. Finally, \mathcal{A} generates a $SK = h(L_1|L_3|L_5)$, successfully. Consequently, Sureshkumar et al.'s scheme is deemed to be vulnerable to this attack that aims to compromise the confidentiality of the session key.

B. Impersonation Attack

Based on the adversary model III-A, \mathcal{A} may attempt lethal security threats and also can extract the secret parameters $\{F_i, E_i, G_i, K_i\}$ of the *SC*. Thus, \mathcal{A} attempts to impersonate the legitimate U_i in this attack.

- **IA-1:** \mathcal{A} computes $C_i = (L_2 \oplus F_i \oplus W_2) = \{(L_1 \oplus S_i) \oplus (A_1 \oplus S_i) \oplus (L_1 \oplus A_1 \oplus C_i)\}$, $L_1 = W_1 \oplus C_i \oplus L_2$, $A_1 = W_2 \oplus L_1 \oplus C_i$, and $S_i = L_2 \oplus L_1$. And then, \mathcal{A} calculates $L_3 = N_{u_1} \oplus h(L_1 \oplus S_i)$ and $L_5 = N_{u_2} \oplus h(L_1 || S_i)$.
- **IA-2:** \mathcal{A} generates a random number R_A and computes $L_{A1} = h(A_1 || R_A)$, $L_{A2} = L_{A1} \oplus S_i$, $Auth_A = h(L_{A1} || L_{A2} || T_{A1})$, $W_{A1} = L_{A1} \oplus C_i$, and $W_{A2} = L_{A1} \oplus A_1 \oplus C_i$. Then, \mathcal{A} transmits the login message $M_1 = \{W_{A1}, W_{A2}, L_{A2}, Auth_A, T_{A1}\}$ to the CS through an insecure channel.
- **IA-3:** CS checks the $|T_2 - T_{A1}| \leq \Delta T_i$. If the timestamp is matches, CS generates a random number R_2 and then calculates $L_3 = h(ID_{CS} || R_2)$, $L_4 = L_3 \oplus c_j$, and $Auth_{CS} = h(L_3 || L_4 || T_2)$. CS transmits $M_2 = \{ID_{CS}, L_4, Auth_{CS}, T_2, W_{A1}, W_{A2}, L_{A2}, Auth_A, T_{A1}\}$ to the USP .
- **IA-4:** USP verifies the freshness of $|T_2 - T_3| \leq \Delta T_i$. If it is equal, USP computes $c_j = h(ID_{CS} || MK_{USP})$, $L_3^* = L_4 \oplus c_j$, and $Auth_{CS}^* = h(L_3^* || L_4 || T_2)$, and checks $Auth_{CS}^* \stackrel{?}{=} Auth_{CS}$. If it matches, USP computes $A_1^* \oplus L_{A2}^* = W_1 \oplus W_{A2}$, $A_1^* = A_1^* \oplus L_{A2}^* \oplus L_{A2}$, $S_i^* = h(A_1^* || MK_{USP})$, $L_{A1}^* = L_{A2} \oplus S_i^*$, and $Auth_A^* = h(L_{A1}^* || L_{A2} || T_{A1})$.
- **IA-5:** USP checks $Auth_A^* \stackrel{?}{=} Auth_A$. If it is matches, USP selects a random nonce R_3 and computes $L_5 = h(T_{A1} || T_2 || T_3 || R_3)$, $N_{A1} = L_{A3}^* \oplus h(L_{A1}^* || S_i)$, $N_{A2} = L_5 \oplus h(L_{A1}^* || S_i)$, and $N_{CS1} = L_{A1}^* \oplus h(L_3^* || c_j)$, $N_{CS2} = L_5 \oplus h(L_{A3}^* || c_j)$, $NAuth_{CS} = h(N_{CS1} || N_{CS2} || c_j || L_3^*)$, and $NAuth_A = h(N_{A1} || N_{A2} || S_i || L_{A1}^*)$. Finally, USP sends $M_3 = \{N_{CS1}, N_{CS2}, NAuth_{CS}, N_{A1}, N_{A2}, NAuth_A\}$ to the CS .
- **IA-6:** CS calculates $NAuth_{CS}^* = h(N_{CS1} || N_{CS2} || c_j || L_{A3})$ and verifies $NAuth_{CS}^* \stackrel{?}{=} NAuth_{CS}$. If it is equal, CS authenticates USP and then computes $L_{A1}^* = N_{CS1} \oplus h(L_{A3} || c_j)$ and $L_5^* = N_{CS2} \oplus h(L_{A3} || c_j)$. Finally, CS transmits $M_4 = \{N_{A1}, N_{A2}, NAuth_A\}$ to the \mathcal{A} .
- **IA-7:** \mathcal{A} computes $NAuth_A^* = h(N_{A1} || N_{A2} || S_i || L_{A1})$ and checks $NAuth_A^* \stackrel{?}{=} NAuth_A$. If it is correct, \mathcal{A} computes $L_{A3}^* = N_{A1} \oplus h(L_{A1} || S_i)$ and $L_5^* = N_{A2} \oplus h(L_{A1} || S_i)$. Finally, \mathcal{A} establishes a common session key $SK_A = h(L_{A1} || L_{A3} || L_5)$ with the USP and CS .

Consequently, Sureshkumar et al.'s scheme cannot prevent impersonation attacks because \mathcal{A} can impersonate as the legitimate U_i .

C. Mutual Authentication

Sureshkumar et al. claimed scheme [12] guarantees secure "mutual authentication". Unfortunately, according to Section V-A and V-B, \mathcal{A} can successfully create the login message $Auth_U = h(L_1 || L_2 || T_1)$ and authentication message $NAuth_U = h(N_{u_1} || N_{u_2} || S_i || L_1)$ for mutual authentication. As a result, Sureshkumar et al.'s scheme lacks secure "mutual authentication" between U_i , USP , and CS .

VI. PROPOSED SCHEME

We design a "PUF-based robust and anonymous AKE scheme for V2G networks (R2AKE-V2G)" to enhance the security shortcomings of [12].

A. Initial Setup Phase

USP first generates a master private key MK_{USP} and comprises the $h(\cdot)$. And then, USP publishes the $h(\cdot)$ as public details.

B. Registration Phase

The registration phase has consisted of two parts: CS and U_i registration phases. This phase is performed via a secure channel.

1) *Charging Station Registration Phase:* CS selects a identity ID_{CS} and a set of (C_{CS}^x, R_{CS}^x) then transmits $\{ID_{CS}, (C_{CS}^x, R_{CS}^x)\}$ to the USP via a secure channel. After that, USP computes $Z_j = h(ID_{CS} || ID_{USP} || MK_{USP} || R_{CS}^x)$ and $c_j = h(ID_{CS} || MK_{USP})$ and then transmits its to the CS securely. Finally, USP removes Z_j and c_j and stores $\{(C_{CS}^x, R_{CS}^x), ID_{CS}\}$ in the database (DB). CS stores $\{(C_{CS}^x, R_{CS}^x), Z_j, c_j\}$ securely.

2) *User Registration Phase:* Before AKE phase, U_i registers within USP to access the useful V2G services and gets the credential from USP .

URP-1: U_i selects a ID_U and PW_i and imprints BIO . After that, U_i generates a set of (C_U^x, R_U^x) and calculates $RID_i = h(ID_i || BIO)$ and $RPW_i = h(PW_i || BIO)$ and then transmits $\{RID_i, RPW_i, (C_U^x, R_U^x)\}$ to the USP .

URP-2: USP computes $X_i = h(RID_i || MK_{USP} || R_U^x)$, $Q_i = X_i \oplus h(RID_i || R_U^x) \oplus RPW_i$ and $W_i = h(RID_i || R_U^x || X_i || RPW_i)$. After that, USP stores $\{Q_i, W_i\}$ in the SC and transmits SC to the U_i . Then, USP_i computes $E_i = X_i \oplus ID_{USP} \oplus MK_{USP}$ and stores $\{E_i, (C_U^x, R_U^x)\}$ in the DB .

C. Authentication and Key Establishment Phase

If U_i wants to access V2G services, U_i must mutually authenticates USP with the help of CS and establishes a SK among U_i , CS , and USP . This AKE phase is performed over an open channel. This AKE phase is presented as shown in Fig. 3 and presents detailed descriptions of AKE phase.

AKE-1: U_i inputs ID_U , PW_i , and imprints BIO in SC . Then, SC computes $RID_i = h(ID_U || BIO)$, $RPW_i = h(PW_i || BIO)$, $X_i = Q_i \oplus h(RID_i || R_U^x) \oplus RPW_i$, $W_i^* = h(RID_i || R_U^x || X_i || RPW_i)$, and verifies whether $W_i^* \stackrel{?}{=} W_i$. If it matches, SC accepts U_i , otherwise; terminates and rejects the current session. SC generates a random nonce R_1 , a timestamp T_1 , and a pair of (C_U^1, R_U^1) from the premise set (C_U^x, R_U^x) . Then, SC computes $M_1 = (ID_U || R_1) \oplus h(X_i || RID_i || R_U^1 || T_1)$ and

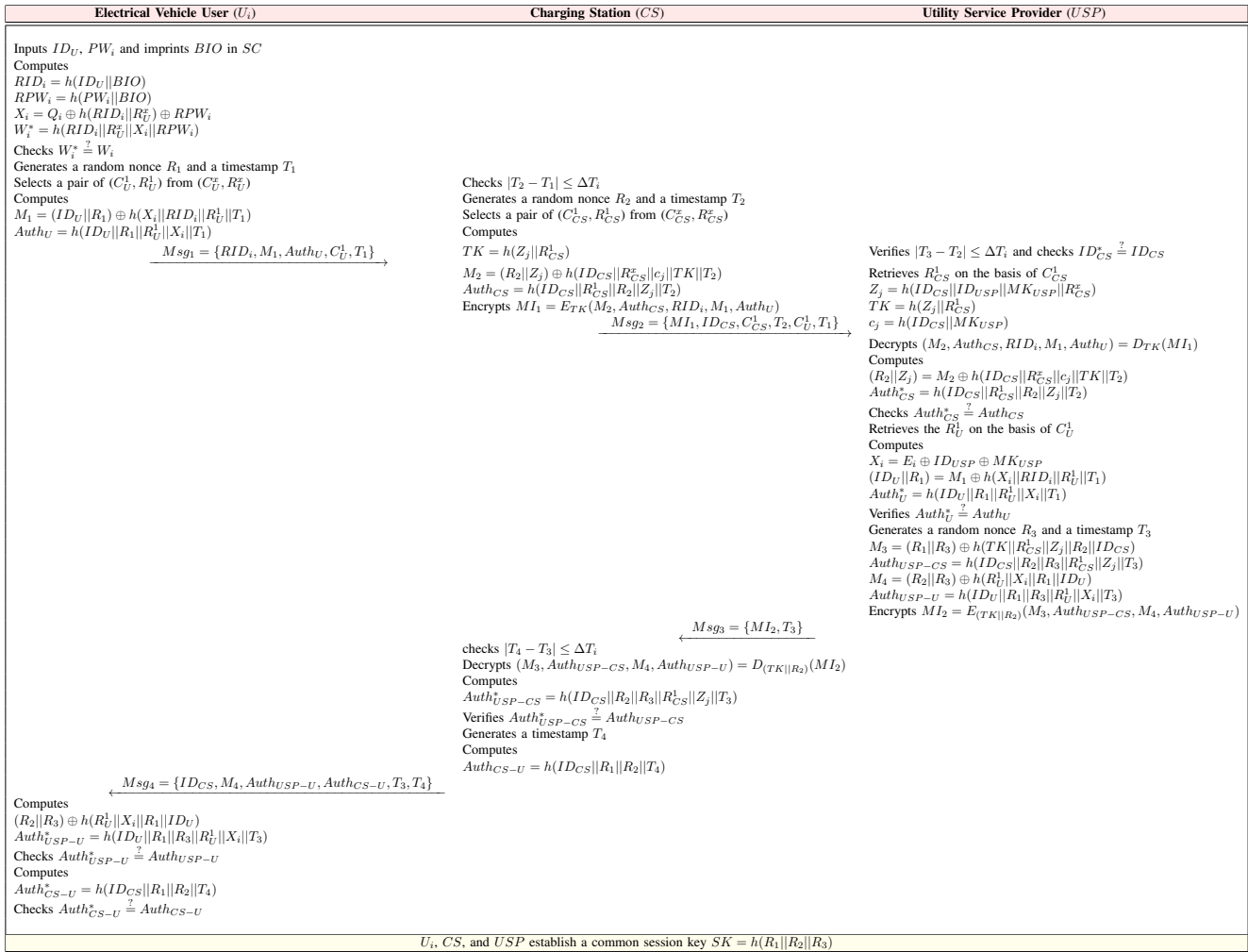


Fig. 3: Summary of Authentication and Key Establishment Phase of R2AKE-V2G

$Auth_U = h(ID_U || R_1 || R_U^1 || X_i || T_1)$ and then sends $Msg_1 = \{RID_i, M_1, Auth_U, C_U^1, T_1\}$ to CS .

AKE-2: CS checks the freshness of $|T_2 - T_1| \leq \Delta T_i$.

If T_1 matches, CS generates a R_2 , a T_2 , and a pair of (C_{CS}^1, R_{CS}^1) from the premise set (C_{CS}^*, R_{CS}^*) . Then, CS computes $TK = h(Z_j || R_{CS}^1)$, $M_2 = (R_2 || Z_j) \oplus h(ID_{CS} || R_{CS}^1 || c_j || TK || T_2)$ and $Auth_{CS} = h(ID_{CS} || R_{CS}^1 || R_2 || Z_j || T_2)$. CS encrypts $MI_1 = E_{TK}(M_2, Auth_{CS}, RID_i, M_1, Auth_U)$ and sends $Msg_2 = \{MI_1, ID_{CS}, C_{CS}^1, T_2, C_U^1, T_1\}$ to USP .

AKE-3: USP verifies $|T_3 - T_2| \leq \Delta T_i$ and checks $ID_{CS}^* \stackrel{?}{=} ID_{CS}$. If T_1 and ID_{CS} matches, USP retrieves the R_{CS}^1 on the basis of C_{CS}^1 and computes $Z_j = h(ID_{CS} || ID_{USP} || MK_{USP} || R_{CS}^1)$, $TK = h(Z_j || R_{CS}^1)$, $c_j = h(ID_{CS} || MK_{USP})$, and decrypts $(M_2, Auth_{CS}, RID_i, M_1, Auth_U) = D_{TK}(MI_1)$. After that, USP computes $(R_2 || Z_j) = M_2 \oplus h(ID_{CS} || R_{CS}^1 || c_j || TK || T_2)$, and $Auth_{CS}^* = h(ID_{CS} || R_{CS}^1 || R_2 || Z_j || T_2)$, and verifies whether $Auth_{CS}^* \stackrel{?}{=} Auth_{CS}$. If

it matches, USP authenticates CS and then retrieves the R_U^1 on the basis of C_U^1 and computes $X_i = E_i \oplus ID_{USP} \oplus MK_{USP}$, $(ID_U || R_1) = M_1 \oplus h(X_i || RID_i || R_U^1 || T_1)$, and $Auth_U^* = h(ID_U || R_1 || R_U^1 || X_i || T_1)$ and verifies whether $Auth_U^* \stackrel{?}{=} Auth_U$. If it matches, USP authenticates U_i successfully. After that, USP generates a R_3, T_3 and computes $M_3 = (R_1 || R_3) \oplus h(TK || R_{CS}^1 || Z_j || R_2 || ID_{CS})$, $Auth_{USP-CS} = h(ID_{CS} || R_2 || R_3 || R_{CS}^1 || Z_j || T_3)$, $M_4 = (R_2 || R_3) \oplus h(R_U^1 || X_i || R_1 || ID_U)$, and $Auth_{USP-U} = h(ID_U || R_1 || R_3 || R_U^1 || X_i || T_3)$, and encrypts $MI_2 = E_{(TK || R_2)}(M_3, Auth_{USP-CS}, M_4, Auth_{USP-U})$. Finally, USP transmits $Msg_3 = \{MI_2, T_3\}$ to CS .

AKE-4: CS checks freshness of $|T_4 - T_3| \leq \Delta T_i$. If T_3 matches, CS decrypts $(M_3, Auth_{USP-CS}, M_4, Auth_{USP-U}) = D_{(TK || R_2)}(MI_2)$ and computes $Auth_{USP-CS}^* = h(ID_{CS} || R_2 || R_3 || R_{CS}^1 || Z_j || T_3)$ and verifies $Auth_{USP-CS}^* \stackrel{?}{=} Auth_{USP-CS}$. If it equals, CS authenticates USP . CS selects a T_4 computes $Auth_{CS-U} = h(ID_{CS} || R_1 || R_2 || T_4)$ and sends

$M_{sg4} = \{ID_{CS}, M_4, Auth_{USP}, Auth_{CS}, T_3, T_4\}$.

AKE-5: U_i computes $(R_2||R_3) \oplus h(R_U^1||X_i||R_1||ID_U)$, $Auth_{USP-U}^* = h(ID_U||R_1||R_3||R_U^1||X_i||T_3)$, and verifies $Auth_{USP-U}^* \stackrel{?}{=} Auth_{USP-U}$. If it matches, U_i authenticates USP . After that, U_i computes $Auth_{CS-U}^* = h(ID_{CS}||R_1||R_2||T_4)$ and checks $Auth_{CS-U}^* \stackrel{?}{=} Auth_{CS-U}$. If it is equal, U_i authenticates CS successfully.

Consequently, U_i , CS , and USP are mutually authenticated and establish a common $SK = h(R_1||R_2||R_3)$.

We present the authentication and key establishment by executing the following sequence of procedures whose details are as shown in Algorithm 1, 2, 3, and 4.

D. Password Update Phase

If U_i wants to change a new PW_i , U_i may update previous PW_i without requiring interaction with the USP .

PUP-1: U_i inputs an ID_U and an old PW_i^{old} , and imprints BIO in SC . SC computes $RID_i = h(ID_U||BIO)$, $RPW_i = h(PW_i||BIO)$, $X_i = Q_i \oplus h(RID_i||R_U^x) \oplus RPW_i$, and $W_i^* = h(RID_i||R_U^x||X_i||RPW_i)$. SC verifies $W_i^* \stackrel{?}{=} W_i$. If it matches, SC accepts U_i , otherwise; terminates and rejects the current session.

PUP-2: After that, SC enters a new PW_i^{new} and computes $RPW_i^{new} = PW_i^{new}||BIO$, $Q_i^{new} = X_i \oplus h(RID_i||R_U^x) \oplus RPW_i^{new}$, and $W_i = RID_i||R_U^x||X_i||RPW_i^{new}$. Finally, SC is updated as $SC = (Q_i^{new}, W_i^{new})$.

Algorithm 1 User Authentication Request

```

1: Input: The identity,  $ID_i$ , password  $PW_i$ , and biometric  $BIO$ 
2: Procedure Electrical vehicle user ( $U_i$ )
3:    $RID_i \leftarrow h(ID_U||BIO)$ 
4:    $RPW_i \leftarrow h(PW_i||BIO)$ 
5:    $X_i \leftarrow Q_i \oplus h(RID_i||R_U^x) \oplus RPW_i$ 
6:    $W_i^* \leftarrow h(RID_i||R_U^x||X_i||RPW_i)$ 
7:    $(C_U^1, R_U^1)$  from  $(C_U^x, R_U^x)$ 
8:    $M_1 \leftarrow (ID_U||R_1) \oplus h(X_i||RID_i||R_U^1||T_1)$ 
9:    $Auth_U \leftarrow h(ID_U||R_1||R_U^1||X_i||T_1)$ 
10: if then( $W_i^* \stackrel{?}{=} W_i$ ) then
11:    $R_1 \leftarrow RandomNonce()$ ; and  $T_1 \leftarrow Timestamp()$ ;
12:   Select a pair of  $(C_U^1, R_U^1)$  from the premise set  $(C_U^x, R_U^x)$ 
13:    $M_1 \leftarrow (ID_U||R_1) \oplus h(X_i||RID_i||R_U^1||T_1)$ 
14:    $Auth_U \leftarrow h(ID_U||R_1||R_U^1||X_i||T_1)$ 
15:   Transmits  $M_{sg1} \leftarrow \{RID_i, M_1, Auth_U, C_U^1, T_1\}$  to  $USP$  with help from  $CS$ ;
16:   else
17:     return Reject Authentication Request;
18: end if

```

VII. SECURITY ANALYSIS

This section introduces the formal and informal security analyses.

A. Formal Security Analysis Using ROR Oracle Model

We evaluate a SK of R2AKE-V2G over the ‘‘ROR oracle model [15]’’. We introduce the necessary queries for ‘‘ROR oracle model [15]’’.

Algorithm 2 User Authentication Confirmation and Response

```

1: Procedure  $USP (MI_1, ID_{CS}, M_1, Auth_U, C_U^1, C_{CS}^1, T_1, T_2)$ 
2: if then( $|T_3 - T_2| \leq \Delta T_i$ ) and  $(ID_{CS}^1 \stackrel{?}{=} ID_{CS})$  then
3:    $R_{CS}^1$  on the basis of  $C_{CS}^1$ 
4:    $Z_j \leftarrow h(ID_{CS}||ID_{USP}||MK_{USP}||R_{CS}^x)$ 
5:    $TK \leftarrow h(Z_j||R_{CS}^1)$ 
6:    $(M_2, Auth_{CS}, RID_i, M_1, Auth_U) = D_{TK}(MI_1)$ 
7:   else
8:     return Reject Authentication Request and Report Replay Attack;
9: end if
10:   $R_3 \leftarrow RandomNonce()$ ; and  $T_3 \leftarrow Timestamp()$ ;
11:   $R_U^1$  on the basis of  $C_U^1$ 
12:   $X_i \leftarrow E_i \oplus ID_{USP} \oplus MK_{USP}$ 
13:   $(ID_U||R_1) \leftarrow M_1 \oplus h(X_i||RID_i||R_U^1||T_1)$ 
14:   $Auth_U^* \leftarrow h(ID_U||R_1||R_U^1||X_i||T_1)$ 
15: if then( $Auth_U^* \stackrel{?}{=} Auth_U$ ) then
16:    $USP$  authenticates  $U_i$ 
17:    $M_4 \leftarrow (R_2||R_3) \oplus h(R_U^1||X_i||R_1||ID_U)$ 
18:    $Auth_{USP-U} \leftarrow h(ID_U||R_1||R_3||R_U^1||X_i||T_3)$ 
19:   Transmits  $M_{sg4} \leftarrow \{ID_{CS}, M_4, Auth_{USP-U}, T_3\}$  to  $U_i$  with help from  $CS$ ;
20:   else
21:     return Reject Authentication Confirmation;
22: end if
23: Procedure  $U_i (ID_{CS}, M_4, Auth_{USP-U}, T_3)$ 
24:   $(R_2||R_3) \oplus h(R_U^1||X_i||R_1||ID_U)$ 
25:   $Auth_{USP-U}^* \leftarrow h(ID_U||R_1||R_3||R_U^1||X_i||T_3)$ 
26: if then( $Auth_{USP-U}^* \stackrel{?}{=} Auth_{USP-U}$ ) then
27:   $U_i$  authenticates  $USP$ 
28:   $Auth_{CS-U}^* \leftarrow h(ID_{CS}||R_1||R_2||T_4)$ 
29:  else
30:    return Reject Authentication Response;
31: end if
32: if then( $Auth_{CS-U}^* \stackrel{?}{=} Auth_{CS-U}$ ) then
33:   $U_i$  authenticates  $CS$ 
34:  else
35:    return Reject Authentication;
36: end if

```

Algorithm 3 Charging Station Authentication Request

```

1: Procedure Charging Station  $CS (RID_i, M_1, Auth_U, C_U^1, T_1)$ 
2: if then( $|T_2 - T_1| \leq \Delta T_i$ ) then
3:    $R_2 \leftarrow RandomNonce()$ ; and  $T_2 \leftarrow Timestamp()$ ;
4:    $(C_{CS}^1, R_{CS}^1)$  from the premise set  $(C_{CS}^x, R_{CS}^x)$ 
5:    $TK \leftarrow h(Z_j||R_{CS}^1)$ 
6:    $M_2 \leftarrow (R_2||Z_j) \oplus h(ID_{CS}||R_{CS}^x||c_j||TK||T_2)$ 
7:    $Auth_{CS} \leftarrow h(ID_{CS}||R_{CS}^1||R_2||Z_j||T_2)$ 
8:    $MI_1 \leftarrow E_{TK}(M_2, Auth_{CS}, RID_i, M_1, Auth_U)$ 
9:   Transmits  $M_{sg2} \leftarrow \{MI_1, ID_{CS}, C_{CS}^1, T_2, C_U^1, T_1\}$  to  $USP$ ;
10:  else
11:    return Reject Authentication Request and Report Replay Attack;
12: end if

```

In R2AKE-V2G, there are three parties: ‘‘the electrical vehicle user $\Gamma_U^{t_1}$, the charging station $\Gamma_{CS}^{t_2}$, and the utility service provider $\Gamma_{USP}^{t_3}$, where $\Gamma_U^{t_1}$, $\Gamma_{CS}^{t_2}$, and $\Gamma_{USP}^{t_3}$ are instances t_1^{th} of U , t_2^{th} of CS , and t_3^{th} of USP ’’, respectively. Table III indicates the essential queries like ‘‘Send(\cdot), Execute(\cdot), Corrupt $CS(\cdot)$, Corrupt $SC(\cdot)$, Reveal(\cdot), and Test(\cdot)’’. We use a ‘‘hash function Hash(\cdot)’’ and a ‘‘PUF function PUF(\cdot)’’ as a random oracle. We use ‘‘Zipf’s law [42]’’ to prove SK security of R2AKE-V2G.

Theorem. $Adv_A^{R2AKE-V2G}$ means the advantages of \mathcal{A} in flouting SK security for R2AKE-V2G. Thus, we derive the following:

$$Adv_A^{R2AKE-V2G} \leq \frac{q_h^2}{|Hash|} + \frac{q_P^2}{|PUF|} + 2\{C \cdot q_{send}^s, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\}$$

Algorithm 4 Charging Station Confirmation and Response

```

1: Procedure  $USP(MI_1, ID_{CS}, C_{CS}^1, T_2)$ 
2: if then  $(|T_3 - T_2| \leq \Delta T_i)$  and  $(ID_{CS}^* \stackrel{?}{=} ID_{CS})$  then
3:    $R_{CS}^1$  on the basis of  $C_{CS}^1$ 
4:    $Z_j \leftarrow h(ID_{CS} || ID_{USP} || MK_{USP} || R_{CS}^x)$ 
5:    $TK \leftarrow h(Z_j || R_{CS}^1)$ 
6:    $c_j \leftarrow h(ID_{CS} || MK_{USP})$ 
7:    $(M_2, Auth_{CS}, RID_i, M_1, Auth_U) \leftarrow D_{TK}(MI_1)$ 
8:    $(R_2 || Z_j) \leftarrow M_2 \oplus h(ID_{CS} || R_{CS}^x || c_j || TK || T_2)$ 
9:    $Auth_{CS}^* \leftarrow h(ID_{CS} || R_{CS}^1 || R_2 || Z_j || T_2)$ 
10:  else
11:    return Reject Authentication Request and Report Replay Attack;
12: end if
13: if then  $(Auth_{CS}^* \stackrel{?}{=} Auth_{CS})$  then
14:    $USP$  authenticates  $CS$ 
15:    $R_3 \leftarrow RandomNonce();$  and  $T_3 \leftarrow Timestamp();$ 
16:    $M_3 \leftarrow (R_1 || R_3) \oplus h(TK || R_{CS}^1 || Z_j || R_2 || ID_{CS})$ 
17:    $Auth_{USP-CS} \leftarrow h(ID_{CS} || R_2 || R_3 || R_{CS}^1 || Z_j || T_3)$ 
18:    $MI_2 \leftarrow E_{(TK || R_2)}(M_3, Auth_{USP-CS}, M_4, Auth_{USP-U})$ 
19:   Transmits  $M_{sg3} \leftarrow \{MI_2, T_3\}$  to  $CS$ ;
20:  else
21:    return Reject Authentication Confirmation;
22: end if
23: Procedure Charging Station  $CS(MI_2, T_3)$ 
24: if then  $(|T_4 - T_3| \leq \Delta T_i)$  then
25:    $(M_3, Auth_{USP-CS}, M_4, Auth_{USP-U}) \leftarrow D_{(TK || R_2)}(MI_2)$ 
26:    $Auth_{USP-CS}^* \leftarrow h(ID_{CS} || R_2 || R_3 || R_{CS}^1 || Z_j || T_3)$ 
27:  else
28:    return Report Replay Attack;
29: end if
30: if then  $(Auth_{USP-CS}^* \stackrel{?}{=} Auth_{USP-CS})$  then
31:    $CS$  authenticates  $USP$ 
32:  else
33:    return Reject Authentication Response;
34: end if

```

TABLE III: Queries and Purposes

Query	Purpose
$Send(\Gamma^t, Msg)$	Under this query, \mathcal{A} can send the message Msg to the Γ^t , and get the response message accordingly.
$CorruptSC(\Gamma_U^{t_1})$	This query means as the smart card stolen attacks where \mathcal{A} can extract the secret parameters stored in SC .
$CorruptCS(\Gamma_{CS}^{t_2})$	This query means as the physical capture attacks where \mathcal{A} can obtain the secret parameters stored in CS .
$Test(\Gamma^t)$	An unbiased coin c is tossed prior to game start. If \mathcal{A} gets $c = 1$ under the $Test(\cdot)$, it means a SK among $\Gamma_U^{t_1}$, $\Gamma_{CS}^{t_2}$, and $\Gamma_{USP}^{t_3}$ are fresh. If \mathcal{A} gets the $c = 0$, it means SK is not fresh; otherwise, \mathcal{A} obtains a null value (\perp).
$Execute(\Gamma_U^{t_1}, \Gamma_{CS}^{t_2}, \Gamma_{USP}^{t_3})$	Under this query, \mathcal{A} tries the passive/active attacks by eavesdropping the transmitted messages among $\Gamma_U^{t_1}$, $\Gamma_{CS}^{t_2}$, and $\Gamma_{USP}^{t_3}$ over a public channel.
$Reveal(\Gamma^t)$	Under this query, \mathcal{A} compromises a SK generated among $\Gamma_U^{t_1}$, $\Gamma_{CS}^{t_2}$, and $\Gamma_{USP}^{t_3}$.

$Hash$, q_P , q_h , and q_{send} are the “number of $Hash$ query”, “range space of $PUF(\cdot)$ ”, “range space of $h(\cdot)$ ”, and “ $Send(\cdot)$ query”. And also, l_n , s , l_m , and C are the Zipf’s credentials [42].

Proof. We indicate the games GM_i ($i \in [0, 4]$). We introduce that $Adv_{\mathcal{A}, GM_i}^{R2AKE-V2G}$ is the probability of \mathcal{A} for winning the GM_i .

Game GM_0 : GM_0 is considered as “an actual attacks executed by \mathcal{A} ” in R2AKE-V2G. The GM_0 ’s result is as follows:

$$Adv_{\mathcal{A}}^{R2AKE-V2G} = |2 \cdot Adv_{\mathcal{A}, GM_0}^{R2AKE-V2G} - 1| \quad (1)$$

Game GM_1 : GM_1 means that \mathcal{A} performs an “eavesdropping attack in which the transmitted messages are intercepted among U , CS , and USP performing $Execute(\cdot)$ query”. In this game, \mathcal{A} carry out the “ $Test(\cdot)$ and $Reveal(\cdot)$ ” queries to reveal SK . The output of $Test(\cdot)$ and $Reveal(\cdot)$ queries decide if \mathcal{A} gets SK . To reveal SK , \mathcal{A} needs the $\{R_1, R_2, R_3\}$. Hence, \mathcal{A} ’s probability of winning GM_1 by eavesdropping on the messages does not increase. This game’s result is as below:

$$Adv_{\mathcal{A}, GM_1}^{R2AKE-V2G} = Adv_{\mathcal{A}, GM_0}^{R2AKE-V2G} \quad (2)$$

Game GM_2 : This game is considered as the “active/passive attacks by performing $Hash$ and $Send(\cdot)$ queries”. \mathcal{A} can intercept the $\{Msg_1, Msg_2, Msg_3, Msg_4\}$ during AKE phase. All message are not revealed by \mathcal{A} since it is protected by using $h(\cdot)$ with the $\{R_1, R_2, R_3\}$ and PUF values $\{R_U^1, R_{CS}^1\}$. GM_2 ’s result is as follows:

$$|Adv_{\mathcal{A}, GM_2}^{R2AKE-V2G} - Adv_{\mathcal{A}, GM_1}^{R2AKE-V2G}| \leq \frac{q_h^2}{2|Hash|} \quad (3)$$

Game GM_3 : This particular game is an extension of GM_2 , wherein the simulation of PUF query has been incorporated. Based on analogous argument introduced in GM_2 , this game’s results is below:

$$|Adv_{\mathcal{A}, GM_3}^{R2AKE-V2G} - Adv_{\mathcal{A}, GM_2}^{R2AKE-V2G}| \leq \frac{q_P^2}{2|PUF|} \quad (4)$$

Game GM_4 : This game is considered the simulation of the $CorruptSC(\cdot)$ and $CorruptCS(\cdot)$ queries. \mathcal{A} extract $\{Q_i, W_i\}$ in SC ’s memory by performing the “differential power analysis”. Note that, $Q_i = X_i \oplus h(RID_i || R_U^x) \oplus RPW_i$ and $W_i = h(RID_i || R_U^x || X_i || RPW_i)$. However, this game is computationally infeasible for \mathcal{A} to reveal PW_i of the U_i via $Send(\cdot)$ query without the BIO , And also, \mathcal{A} cannot distinguish the “PUF secret” and “biometric” since the probability of guessing the PUF secret of l_2 and biometric credential of l_1 bits by \mathcal{A} is $\frac{1}{2^{l_2}}$ and $\frac{1}{2^{l_1}}$. Consequently, GM_3 and GM_4 are “indistinguishable if the off-line password and biometric guessing attacks are not implemented”. GM_4 ’s result is as follows:

$$|Adv_{\mathcal{A}, GM_4}^{R2AKE-V2G} - Adv_{\mathcal{A}, GM_3}^{R2AKE-V2G}| \leq \{C \cdot q_{send}^s, \frac{q_s}{2^{l_b}}\} \quad (5)$$

After $GM_0 - GM_4$ are successfully performed, \mathcal{A} attempts to guess the “ c for winning all game by utilizing $Test(\cdot)$ query”. Thus, we get the following:

$$Adv_{\mathcal{A}, GM_4}^{R2AKE-V2G} = \frac{1}{2} \quad (6)$$

Combining the “formulas (1), (2), and (6)”, we get the following:

$$\begin{aligned} \frac{1}{2} Adv_{\mathcal{A}}^{R2AKE-V2G} &= |Adv_{\mathcal{A}, GM_0}^{R2AKE-V2G} - \frac{1}{2}| \\ &= |Adv_{\mathcal{A}, GM_1}^{R2AKE-V2G} - \frac{1}{2}| \\ &= |Adv_{\mathcal{A}, GM_1}^{R2AKE-V2G} - Adv_{\mathcal{A}, GM_4}^{R2AKE-V2G}| \end{aligned} \quad (7)$$

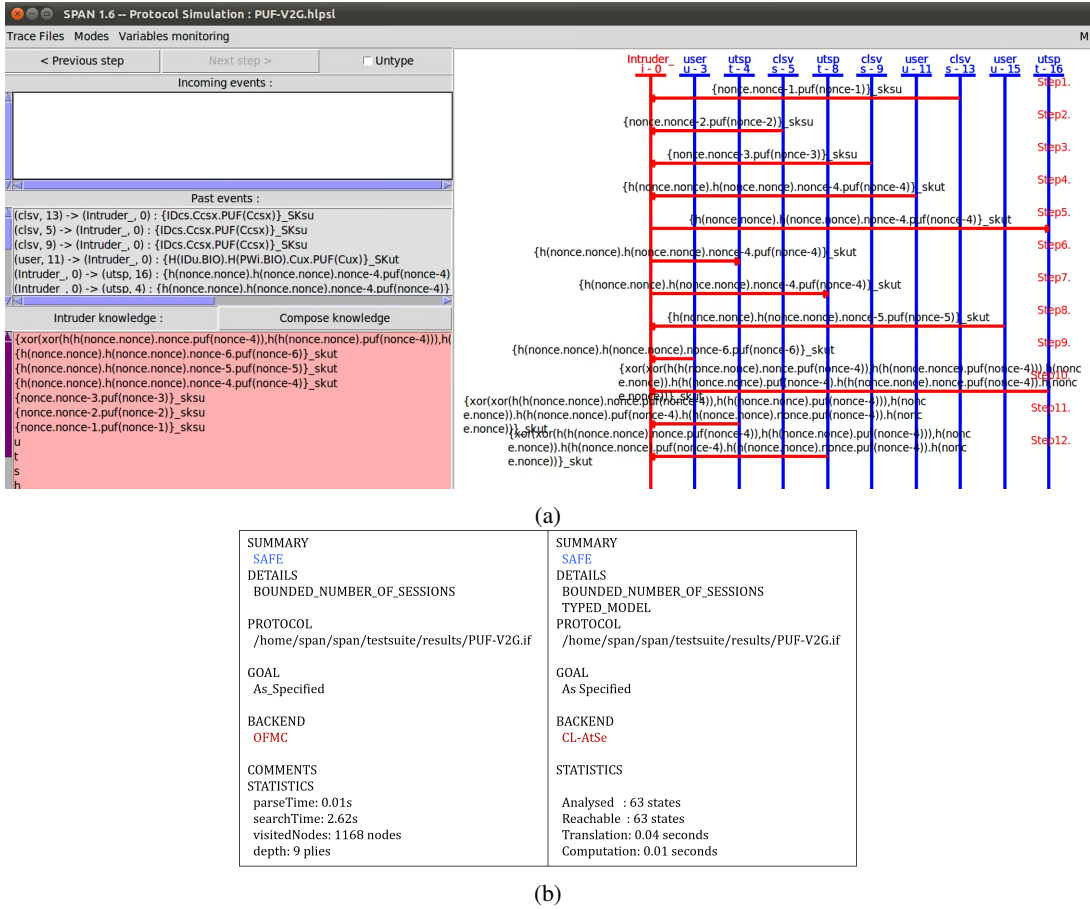


Fig. 4: AVISPA Implementation Results of (a) SPAN, (b) OFMC and CL-AtSe

Based on the application of the “triangular inequality” using the formulas (3, 4, 5, and 7), we can get the subsequent result:

$$\begin{aligned}
 \frac{1}{2} Adv_A^{R2AKE-V2G} &= |Adv_{A,GM_1}^{R2AKE-V2G} - Adv_{A,GM_4}^{R2AKE-V2G}| \\
 &\leq |Adv_{A,GM_1}^{R2AKE-V2G} - Adv_{A,GM_3}^{R2AKE-V2G}| \\
 &\quad + |Adv_{A,GM_3}^{R2AKE-V2G} - Adv_{A,GM_4}^{R2AKE-V2G}| \\
 &\leq |Adv_{A,GM_1}^{R2AKE-V2G} - Adv_{A,GM_2}^{R2AKE-V2G}| \\
 &\quad + |Adv_{A,GM_2}^{R2AKE-V2G} - Adv_{A,GM_3}^{R2AKE-V2G}| \\
 &\quad + |Adv_{A,GM_3}^{R2AKE-V2G} - Adv_{A,GM_4}^{R2AKE-V2G}| \\
 &\leq \frac{q_h^2}{2|Hash|} + \frac{q_P^2}{2|PUF|} + \{C \cdot q_{send}^s \cdot \frac{q_s}{2^{l_1}} \cdot \frac{q_s}{2^{l_2}}\}
 \end{aligned} \tag{8}$$

Finally, by applying a scalar operation of multiplication to both sides of equation (8) with a factor of 2, we get the following: $Adv_A^{R2AKE-V2G} \leq \frac{q_h^2}{|Hash|} + \frac{q_P^2}{|PUF|} + 2\{C \cdot q_{send}^s \cdot \frac{q_s}{2^{l_1}} \cdot \frac{q_s}{2^{l_2}}\}$

B. Formal Security Analysis Using AVISPA Simulation

AVISPA simulation provides evidence of the robustness of the security protocol against lethal security threats. We first implement the R2AKE-V2G as a programming language “High-Level Protocol Specification Language (HLPSSL)” [43]. Subsequently, the simulation commences the analysis of the

Intermediate Format (IF) over the two back-ends: “OFMC” and “CL-AtSe”.

We utilized the “Security Protocol ANimator (SPAN) [14]” based on HLPSSL implementation to simulate R2AKE-V2G. AVISPA supports the DY model and involves a malicious adversary in the security protocol execution with a current session. In Fig. 4, we present the AVISPA implementation results of SPAN, OFMC, and CL-AtSe. SPAN demonstrates the security attacks and the malicious intruder simulated through a GUI” and also OFMC and CL-AtSe show that R2AKE-V2G is secured against lethal security attacks. Consequently, we verified the SAFE output through a formal security analysis and demonstrated that R2AKE-V2G is resistant to various security attacks from a malicious intruder based on the DY threat model.

C. Informal Security Analysis

We demonstrate that R2AKE-V2G exhibits resistance to security attacks and further assures the fulfillment of essential security requirements.

1) *Session Key Disclosure Attack*: \mathcal{A} can steal SC of the legitimate U_i and extract the credentials $\{Q_i, W_i\}$. In R2AKE-V2G, \mathcal{A} must obtain the random nonces $\{R_1, R_2, R_3\}$ to compromise a $SK = h(R_1||R_2||R_3)$. However, \mathcal{A} is difficult to calculate a SK because the random nonces are protected with the PUF secret values $\{R_U^1, R_{CS}^1\}$ and secret

credentials $\{X_i, Z_j\}$ by using “XOR and hash” operations. Thus, R2AKE-V2G is secure to this attack.

2) *Impersonation Attack*: This attack indicates that \mathcal{A} tries to masquerade the U_i by intercepting the transmitted messages under a public channel. \mathcal{A} must create the request messages $\{Msg_1, Msg_2\}$ and response messages $\{Msg_3, Msg_4\}$ related to mutual authentication between other entities. However, it is deemed unfeasible to produce the request and response messages associated with mutual authentication since \mathcal{A} cannot obtain the PUF secret values $\{R_U^1, R_{CS}^1\}$, credentials $\{X_i, Z_j\}$, and temporary secret key $\{TK\}$. Thus, R2AKE-V2G is secure from this attack since \mathcal{A} cannot correctly generate the request and response messages related to mutual authentication.

3) *MITM Attack*: Referring to the information given in Section III-A, \mathcal{A} inject, resend, delete, eavesdrop, intercept, and block the transmitted messages $\{Msg_1, Msg_2, Msg_3, Msg_4\}$ during the bidirectional communication among U_i , CS , and USP . After that, \mathcal{A} tries to get sensitive information for legitimate parties. However, \mathcal{A} is difficult to generate the messages related to authentication because all messages are protected with the PUF secret values $\{R_U^1, R_{CS}^1\}$ and random nonces $\{R_1, R_2, R_3\}$ by using “XOR and hash” operations. Thus, R2AKE-V2G is resistant to this attack because \mathcal{A} cannot obtain the legitimate entity's important information.

4) *ML-based Modeling Attack*: If machine learning methods are used, the existing PUF may be vulnerable to modeling threats. In order to achieve these types of this attack, \mathcal{A} requires accumulating a large subset of possible CRPs like $(C_U^1, R_U^1), (C_{CS}^1, R_{CS}^1) \dots (C_i, R_i)$. Thus, \mathcal{A} make up a mathematical model M^* for PUF behavior from this collection data in order to predict the PUF response R^i to a new challenge C^i . In R2AKE-V2G, even if \mathcal{A} calculates the valid CRP set using a machine learning model method, \mathcal{A} cannot obtain the sensitive information for U_i , CS , and USP because without the knowledge of the shared secret keys $\{X_i, Z_j\}$. Moreover, \mathcal{A} cannot obtain the important information of the legitimate entities even if they obtain CRP by performing the ML-based modeling attack because R2AKE-V2G encrypts and transmits the messages required for authentication using a symmetric key encryption such as advanced encryption standard (AES) algorithm. Consequently, R2AKE-V2G is resistant to this attack because even if \mathcal{A} attempts an attack using the ML-based model method, \mathcal{A} cannot successfully obtain the sensitive information of legitimate entities.

5) *Replay Attack*: \mathcal{A} eavesdrops the U_i 's messages during previous sessions and in another session the \mathcal{A} replays the intercepted messages to involve in the current sessions. Based on the information given in Section III-A, \mathcal{A} eavesdrops the exchanged messages $\{Msg_1, Msg_2, Msg_3, Msg_4\}$ related to mutual authentication during AKE phase. Then, \mathcal{A} tries to authenticate with other entities through the exchange of intercepted messages from the previous session. A solution to resist this attack encompasses the addition of “timestamps” and “random nonces” to the shared information, which renders the data distinctive for each authentication phase. Thus, R2AKE-V2G resists replay attacks since our AKE scheme utilizes timestamps and verifies the freshness of the current times-

tamps T_i . Furthermore, the exchanged messages are protected with the “PUF responses” $\{R_U^1, R_{CS}^1\}$ and the “credentials” $\{X_i, Z_j\}$ in R2AKE-V2G. Thus, our AKE scheme is resistant to this attack.

6) *Physical Capture Attack*: Suppose that CS are physically captured by \mathcal{A} and extracts the credentials $\{c_j\}$ stored in DB , where $c_j = h(ID_{CS} || MK_{USP})$. However, \mathcal{A} does not correctly calculate a common $SK = h(R_1 || R_2 || R_3)$ among U_i , CS , and USP without the knowledge of the secret parameter $\{Z_j\}$, the temporary secret key TK , and random nonces $\{R_2, R_3\}$. Furthermore, there are independent, distinct, and robust for CS 's memory because PUF pair $\{(C_U^x, R_U^x)\}$ and $\{(C_U^1, R_U^1)\}$. Thus, R2AKE-V2G is considerably impervious against this attack given that the output of PUF relies on the inherent physical fluctuations of the IC chip.

7) *Off-line Password Guessing Attack*: According to Section III-A, \mathcal{A} inject, resend, delete, eavesdrop, intercept, and block the transmitted messages and extract the parameters stored in SC . \mathcal{A} tries to this attack to guess the real PW_i for U_i . However, PW_i is comprised of $RPW_i = h(PW_i || BIO)$. Thus, \mathcal{A} is difficult to guess PW_i without knowledge of the biometric BIO . Consequently, this attack is unfeasible in R2AKE-V2G.

8) *Anonymity*: Suppose that \mathcal{A} eavesdrops the exchanged messages during AKE phase. However, \mathcal{A} is unfeasible to obtain the real ID_U for legitimate U_i without knowing such as the “biometric BIO , secret credentials X_i , and PUF secret value R_U^1 ”. R2AKE-V2G provides secure anonymity for U_i .

9) *Perfect Forward Secrecy*: The security protocol for providing “perfect forward secrecy” guarantees that a SK cannot be compromised by any \mathcal{A} even in the event of a long-term key compromise. In the proposed AKE scheme, if USP 's private key MK_{USP} is revealed, \mathcal{A} cannot compute a $SK = h(R_1 || R_2 || R_3)$ because \mathcal{A} cannot get the knowledge of the “PUF responses $\{R_U^1, R_{CS}^1\}$ and secret credentials $\{X_i, Z_j\}$, and random nonces PUF secret values $\{R_1, R_2, R_3\}$ ”. Thus, R2AKE-V2G guarantees perfect forward secrecy.

10) *Mutual Authentication*: In R2AKE-V2G, all entities perform successfully mutual authentication. After receiving the authentication request message $\{Msg_1, Msg_2\}$, USP verifies $Auth_{CS}^* \stackrel{?}{=} Auth_{CS}$. If it is equal, USP authenticates CS . Then, USP verifies whether $Auth_U^* \stackrel{?}{=} Auth_U$. If it matches, USP authenticates U_i . Upon getting the authentication confirmation message $\{Msg_3\}$, CS verifies $Auth_{USP-CS}^* \stackrel{?}{=} Auth_{USP-CS}$. If it matches, CS authenticates USP . After getting the message $\{Msg_4\}$, U_i checks whether $Auth_{USP-U}^* \stackrel{?}{=} Auth_{USP-U}$ and $Auth_{CS-U}^* \stackrel{?}{=} Auth_{CS-U}$. If it is valued, U_i authenticates CS and USP . Hence, R2AKE-V2G allows secure “mutual authentication”.

VIII. PERFORMANCE COMPARATIVE ANALYSIS

We offer a comprehensive analysis of the performance comparison of R2AKE-V2G and existing schemes [12], [21]–[23], [25]–[27] by calculating communication and computation costs during AKE phase. In addition, we compare the security and functionality features.

A. Security Requirement and Functionality

We compare the “security functionalities and requirements” of R2AKE-V2G with the existing schemes for V2G networks [12], [21]–[23], [25]–[27]. Referring to the information given in Table IV, we proved that some related schemes for V2G networks are not fully protected and may be vulnerable to lethal security attacks. Hence, the cryptographic protocol should be designed in such a method that it should be robust and secure to security attacks. In contrast, R2AKE-V2G is secure from “potential security attacks” and also allows the “essential security functionalities and requirements”.

Thus, R2AKE-V2G guarantees more security functionalities and requirements as compared with the existing schemes for V2G networks.

TABLE IV: Comparative Study on Security Features

Security features	[21]	[22]	[23]	[25]	[26]	[27]	[12]	Our
SFR_1	×	×	✓	✓	✓	✓	✓	✓
SFR_2	×	×	×	×	✓	✓	✓	✓
SFR_3	×	×	×	×	✓	×	×	✓
SFR_4	×	×	×	×	✓	✓	×	✓
SFR_5	×	×	×	✓	✓	✓	×	✓
SFR_6	×	×	×	×	✓	✓	✓	✓
SFR_7	×	×	×	×	×	✓	✓	✓
SFR_8	×	×	✓	×	✓	✓	×	✓
SFR_9	×	×	×	✓	✓	✓	✓	✓
SFR_{10}	✓	✓	✓	✓	✓	✓	✓	✓
SFR_{11}	×	×	×	✓	✓	✓	✓	✓

✓: “Protection”; ×: “Non-protection”; SFP_1 : “MITM attack”; SFP_2 : “Offline password guessing attack”; SFP_3 : “Impersonation attack”; SFP_4 : “Smart device stolen attack”; SFP_5 : “Session key disclosure attack”; SFP_6 : “Replay attack”; SFP_7 : “Denial of service attack”; SFP_8 : “Mutual authentication”; SFP_9 : “User anonymity”; SFP_{10} : “Failures of login attempts”; SFP_{11} : “Perfect forward secrecy”.

B. Communication Costs

We present the “communication cost comparison analysis” of the R2AKE-V2G and previous schemes [12], [21]–[23], [25]–[27]. In Table V, we assume the “bit lengths for the timestamp, identity, PUF, random nonce, hash function, symmetric key encryption, elliptic curve point, bilinear pairing, and digital signature”.

TABLE V: Communication Cost (in bits) of Cryptographic Primitives.

Primitives	Communication Cost (bits)
Timestamp	32 bits
Identity	60 bits
PUF	60 bits
Random Nonce	160 bits
Hash Function	160 bits
Symmetric Key Encryption	256 bits
Elliptic Curve Point	320 bits
Bilinear Pairing	320 bits
Digital Signature	1024 bits

In AKE phase of R2AKE-V2G, the transmitted messages require $\{Msg_1 = RID_i, M_1, Auth_U, C_U^1, T_1\}$, $\{Msg_2 = MI_1, ID_{CS}, C_{CS}^1, T_2, C_U^1, T_1\}$, $\{Msg_3 = MI_2, T_3\}$, and $\{Msg_4 = ID_{CS}, M_4, Auth_{USP-U}, Auth_{CS-U}, T_3, T_4\}$ require $(160 + 160 + 160 + 60 + 32 = 572 \text{ bits})$, $(256 +$

$60 + 60 + 32 + 60 + 32 = 500 \text{ bits})$, $(256 + 32 = 288 \text{ bits})$, and $(60 + 160 + 160 + 160 + 32 + 32 = 604 \text{ bits})$. Therefore, R2AKE-V2G an aggregated communication cost of 1964 bits. In Table VII and Fig. 5, we show comparative results for communication costs of R2AKE-V2G and existing related schemes. Consequently, R2AKE-V2G guarantees “lightweight and efficient communication costs” compared with existing schemes [12], [21]–[23], [25]–[27].

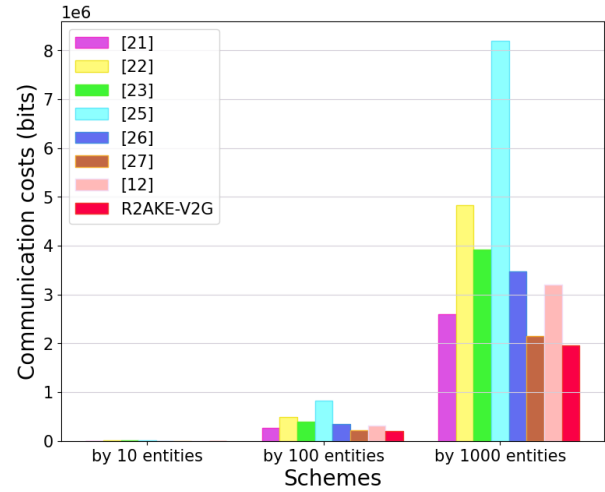


Fig. 5: Communication Cost Comparison

C. Computation Costs

We present the “computation cost comparison analysis” of the R2AKE-V2G and existing schemes [12], [21]–[23], [25]–[27]. We utilize the well-known PBC [44] and JCE [45] libraries in order to deduce the execution times needed for cryptographic primitives. In Table VI, we denote “ T_h , T_s , T_{mp} , T_e , T_b , T_m , $T_{cert_{gen}}$, and $T_{cert_{ver}}$ ” to evaluate the execution times required for “a hash function, a symmetric key encryption/decryption, an elliptic curve multiplication point, a modular exponential, a bilinear pairing, an elliptic curve multiplication, and a certificate generation and verification”. We take the platform for U_i as “Smartphone Lenovo Zuk Z1 with Quad-core 2.5 GHz processor having 4GB RAM and Android Operating System V5.1.2”. And also, we take the platform for CS/USP server as a virtual machine with HP E8300 Core i5 and 2.93 GHz processor with 4GB RAM using Ubuntu 16.11 OS”.

TABLE VI: Execution Time (in milliseconds) of Cryptographic Primitives.

Scheme	User’s Device (ms)	USP/CS Server (ms)
T_h	0.019 ms	0.012 ms
T_s	0.063 ms	0.048 ms
T_{mp}	10.235 ms	5.387 ms
T_e	8.341 ms	3.362 ms
T_b	13.662 ms	7.318 ms
T_m	5.012 ms	2.002 ms
$T_{cert_{gen}}$	69.326 ms	-
$T_{cert_{ver}}$	-	21.257 ms

TABLE VII: Comparative Performance Analysis for Computation and Communication Costs.

Scheme	User's Device	USP/CS Server	Communication Cost
[21]	$3T_{mp} + T_m + T_{cert_{gen}} + T_h \approx 105.062$ ms	$4T_{mp} + T_m + T_{cert_{ver}} + 4T_h + T_s \approx 44.903$ ms	2590 bits
[22]	$2T_{mp} + T_m + T_{cert_{gen}} + T_h + T_s \approx 94.89$ ms	$3T_{mp} + T_m + T_{cert_{ver}} + 3T_h + T_s \approx 39.504$ ms	4836 bits
[23]	$T_s + 4T_h \approx 0.139$ ms	$T_s + 4T_h \approx 0.096$ ms	3922 bits
[25]	$4T_{mp} + T_e + 5T_h \approx 49.376$ ms	$3T_{mp} + T_e + 2T_b + 5T_h \approx 34.219$ ms	8190 bits
[26]	$3T_{mp} + T_e + 6T_h \approx 39.16$ ms	$2T_{mp} + T_e + 2T_b + 6T_h \approx 28.844$ ms	3466 bits
[27]	$6T_h \approx 0.114$ ms	$8T_h \approx 0.096$ ms	2144 bits
[12]	$11T_h \approx 0.209$ ms	$18T_h \approx 0.216$ ms	3196 bits
R2AKE-V2G	$9T_h \approx 0.171$ ms	$15T_h + 4T_s \approx 0.372$ ms	1964 bits

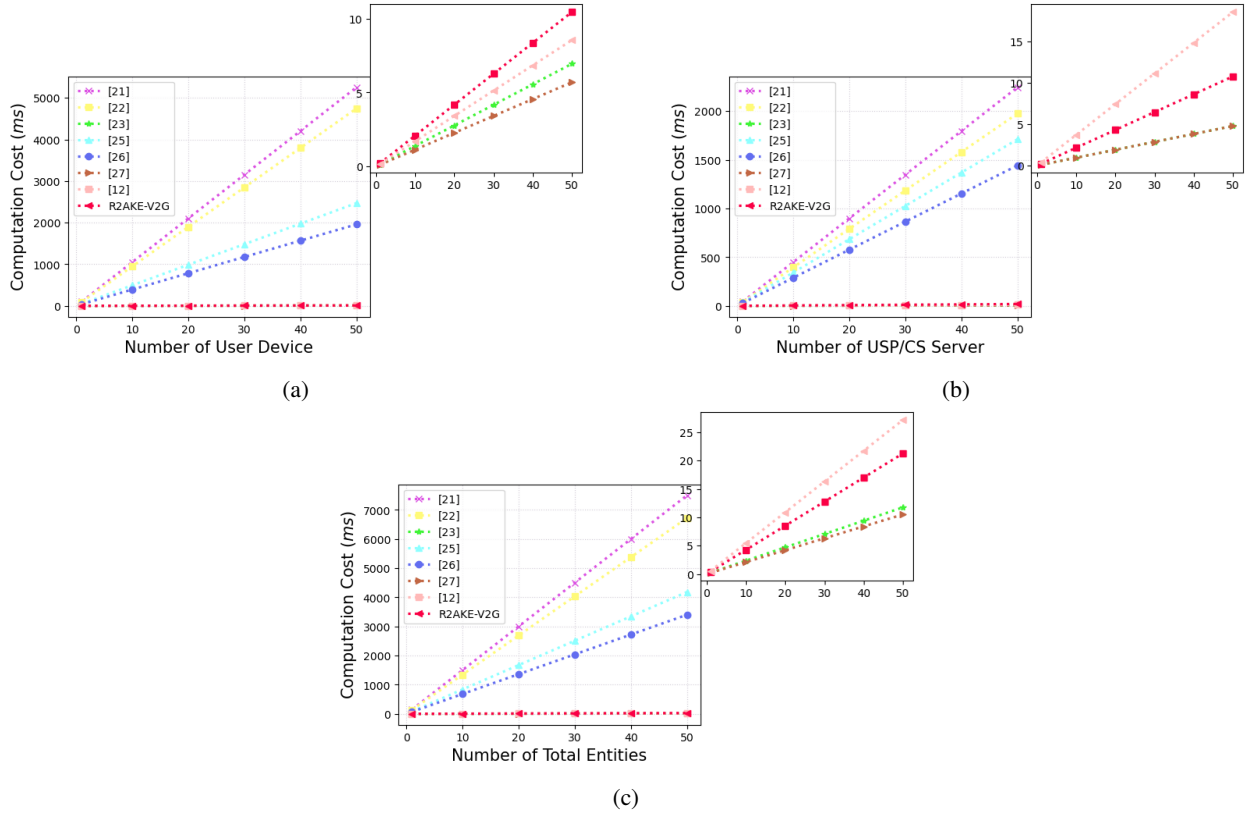


Fig. 6: Computation Cost Comparison of (a) U_i (b) USP/CS (c) total entities

In Table VII and Fig. 6, we demonstrate comparative results for computation costs of R2AKE-V2G and existing related schemes. Although R2AKE-V2G has a slightly higher computation cost compared with existing related scheme [12], [23], [27], the proposed AKE scheme has superior lightweight computation cost compared with another related scheme [21], [22], [25], [26] and also are better the necessary security functionalities and requirements better than existing related scheme [12], [21]–[23], [25]–[27]. Thus, R2AKE-V2G is suitable for practical V2G environments.

IX. NS-3 IMPLEMENTATION

To assess the utility and availability of the R2AKE-V2G, we utilize NS-3 [16] on a system running Ubuntu 20.04.6 LTS, equipped with an Intel Core i5-10400 CPU operating at 2.90 GHz. In the simulation configuration, USP is fixed at the central location, while CS is randomly positioned within a range of 20 to 350m from USP . We also configured the

mobility of U_i to a maximum speed of 15 m/s within a range of 0 to 300m around USP . We perform simulation under three scenarios in compliance with the IEEE 802.11 standard network and the detailed parameters of experimental environments are shown in Table VIII.

In R2AKE-V2G, the entities U_i , CS , and USP engage in the exchange of authentication messages. These messages include $Msg_1 = \{RID_i, M_1, Auth_U, C_U^1, T_1\}$, with a size of 71.5 bytes, $Msg_2 = \{MI_1, ID_{CS}, C_{CS}^1, T_2, C_U^1, T_1\}$ sized at 62.5 bytes, $Msg_3 = \{MI_2, T_3\}$ with a size of 36 bytes, and $Msg_4 = \{ID_{CS}, M_4, Auth_{USP-U}, Auth_{CS-U}, T_3, T_4\}$ with 75.5 bytes. We evaluate the impact of the R2AKE-V2G on “end-to-end delay” and “throughput” of exchanged messages over a duration of 1200 seconds under distinct scenarios.

A. End-to-end delay

We assess the end-to-end delays, representing the average time it takes for data packets to traverse from the source

TABLE VIII: Simulation Parameters

Parameters	Values
Simulation tool	NS-3 (3.29)
Operating system	Ubuntu 20.04.6 LTS
Routing protocol	Optimized link state routing
Simulation time	1800 s
Network	IEEE 802.11
Network area	400 m
Mobility of U_i	Random (0-15 m/s)
Number of U_i	5 (for scenarios 1, 2, 3)
	7 (for scenarios 1, 2, 3)
	10 (for scenarios 1, 2, 3)
Number of CS	3 (for scenario 1)
	6 (for scenario 2)
	9 (for scenario 3)
Number of USP	1 for all scenarios

entity to the destination entity. The computation involves $\sum_{y=0}^{P_t} (T_{R_y} - T_{S_y}) / P_t$, where P_t denotes the total packet count, T_{R_y} signifies the received time of the i th packet, and T_{S_y} denotes the transmission time of the i th packet. Fig. 7 illustrates the observed end-to-end delay in simulation outcomes under three scenarios. As the number of entities increases, the number of packet forwarding increases and the end-to-end delay is likely to be amplified due to resource processing and traffic load. Therefore, we can observe that as the number of U_i and CS increases, the end-to-end delay increases since the distance between entities decreases.

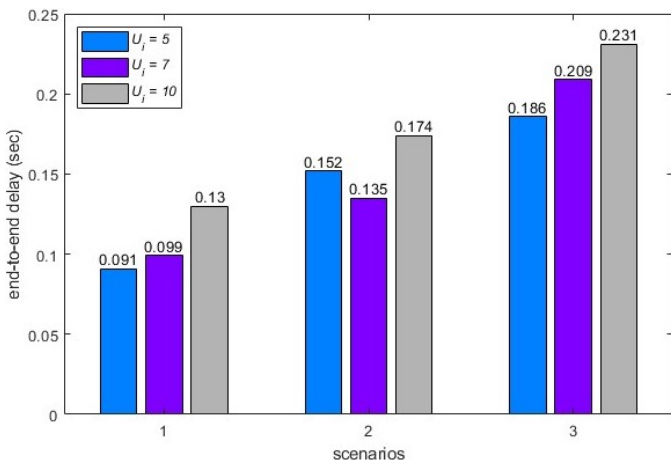


Fig. 7: End-to-End Delay

B. Throughput

Throughput is a metric that quantifies the amount of transmitted data bits per unit of time within a communication network. The calculation for network throughput is expressed by the formula $(R \times |\eta|) / T_R$, where R , $|\eta|$, and T_R represent the number of received packets, the size of an individual packet, and the total time in seconds, respectively. Fig. 8 is the simulation outcomes illustrating the throughput within the proposed scheme, considering an overall simulation duration of 1800 seconds. The analysis reveals that network throughput increases as the number of exchanged messages increases.

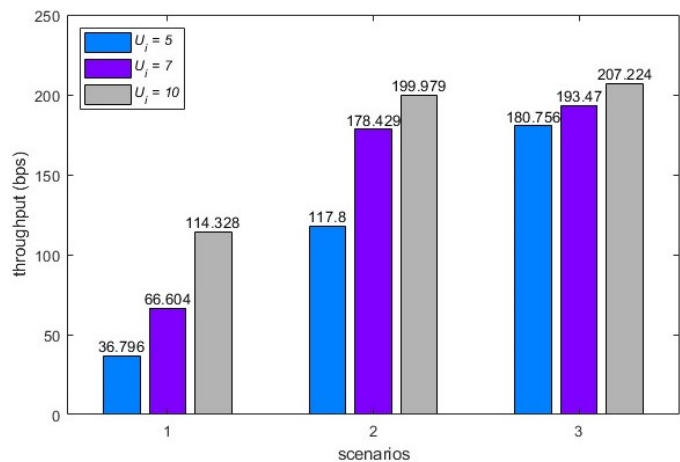


Fig. 8: Throughput.

X. CONCLUSIONS AND FUTURE WORKS

We demonstrate that Sureshkumar et al.'s scheme is not resistant to impersonation and session key disclosure attacks and also lacks secure mutual authentication. Thus, we design a new PUF-based robust and anonymous AKE scheme for V2G networks. We show that R2AKE-V2G is resilient to MITM and replay attacks by using AVISPA implementation analysis. Moreover, we prove the session key security of R2AKE-V2G by using the ROR oracle model. We demonstrate that the implementation of the R2AKE-V2G using NS-3 simulation shows the impact on various network performance parameters. We prove the performance comparison analysis of R2AKE-V2G and the existing schemes for V2G networks with regard to security functionality, communication cost, and computation cost. Hence, R2AKE-V2G guarantees a higher security level than related schemes and also provides better efficient and lightweight computation and communication costs than existing schemes for V2G networks. Therefore, R2AKE-V2G is suitable for actual V2G networks because it is more superior security and lightweight efficiency compared with related schemes for V2G networks.

In future works, we have planned to develop a new architecture and protocol using blockchain technology to integrate R2AKE-V2G into a more complete V2G network.

REFERENCES

- [1] B. Li, M. C. Kisacikoglu, C. Liu, N. Singh, and M. E. Kantarci, "Big Data Analytics for Electric Vehicle Integration in Green Smart Cities," *IEEE Communications Magazine*, vol. 55, pp. 19-25, 2017.
- [2] M. Tao, K. Ota, and M. Dong, "Foud: Integrating Fog and Cloud for 5G-Enabled V2G Networks," *IEEE Networks*, vol. 31, pp. 8-13, 2017.
- [3] S. Yu, K. Park, J. Lee, Y. Park, Y. Park, S. Lee, and B. Chung, "Privacy-Preserving Lightweight Authentication Protocol for Demand Response Management in Smart Grid Environment," *Applied Sciences*, vol. 10, pp. 1-26, 2020.
- [4] I. A. Kamil and S. O. Ogundoyin, "Lightweight Privacy-Preserving Power Injection and Communication Over Vehicular Networks and 5G Smart Grid Slice With Provable Security," *Internet of Things*, vol. 8, pp. 1-24, 2019.
- [5] N. Saxena, S. Grijalva, V. Chukwuka, and A. V. Vasilakos, "Network Security and Privacy Challenges in Smart Vehicle-to-Grid," *IEEE Wireless Communications*, vol. 24, pp. 88-98, 2017.
- [6] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology (CRYPTO)*, Berlin, Germany, 1999, pp. 388-397.

- [7] W. Han and Y. Xiao, "Privacy Preservation for V2G Networks in Smart Grid: A Survey," *Computer Communications*, vol. 91, pp. 17-28, 2016.
- [8] 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 Access*, vol. 5, pp. 2526-2536, 2018.
- [9] N. Saxena and B. J. Choi, "Authentication Scheme for Flexible Charging and Discharging of Mobile Vehicles in the V2G Networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 1438-1452, 2016.
- [10] A. Abdallah and X. S. Shen, "Lightweight Authentication and Privacy-Preserving Scheme for V2G Connections," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 2615-2629, 2017.
- [11] Y. Su, G. Shen, and M. Zhang, "A Novel Privacy-Preserving Authentication Scheme for V2G Networks," *IEEE Systems Journal*, vol. 14, pp. 1963-1971, 2020.
- [12] V. Sureshkumar, P. Chinnaraj, P. Saravanan, R. Amin, and J. J. P. C. Rodrigues, "Authenticated Key Agreement Protocol for Secure Communication Establishment in Vehicle-to-Grid Environment With FPGA Implementation," *IEEE Transactions on Vehicular Technology*, vol. 71, pp. 3470-3479, 2022.
- [13] AVISPA, "Automated Validation of Internet Security Protocols and Applications," 2001, <http://www.avispa-project.org/> (Accessed on 16 March 2022).
- [14] SPAN, "A Security protocol animator for AVISPA," 2001, <http://www.avispa-project.org/> (Accessed on 16 March 2022).
- [15] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authentication key exchange in the three-party setting," in *Public Key Cryptography*, Les Diablerets, Switzerland, 2005, pp. 65-84.
- [16] NS03.29. Accessed: . [Online]. Available: <https://www.nsnam.org/releases/ns-3-29/>
- [17] J. Chen, Y. Zhang, and W. Su, "An Anonymous Authentication Scheme for Plug-in Electric Vehicles Joining to Charging/Discharging Station in Vehicle-to-Grid (V2G) Networks," *China Communications*, vol. 12, pp. 9-19, 2015.
- [18] D. A. Mood, A. O. Sharif, S. M. Mazinani, and M. Nikooghdam, "Provably Secure Escrow-Less Chebyshev Chaotic Map-Based Key Agreement Protocol for Vehicle to Grid Connections With Privacy Protection," *IEEE Transactions on Industrial Informatics*, vol. 16, pp. 7287-7294, 2020.
- [19] Y. Zhang, J. Zou, and R. Guo, "Efficient Privacy-Preserving Authentication for V2G Networks," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 1366-1378, 2021.
- [20] A. Mohammadali, M. S. Haghghi, M. H. Tadayon, and A. M. Nodoshan, "A Novel Identity-Based Key Establishment Method for Advanced Metering Infrastructure in Smart Grid," *IEEE Transactions on Smart Grid*, vol. 9, pp. 2834-2842, 2018.
- [21] H. Nicanfar and V. C. Leung, "Multilayer Consensus ECC-Based Password Authenticated Key-Exchange (MCEPAK) Protocol for Smart Grid System," *IEEE Transactions on Smart Grid*, vol. 4, pp. 253-264, 2013.
- [22] D. Wu and C. Zhou, "Fault-Tolerant and Scalable Key Management for Smart Grid," *IEEE Transactions on Smart Grid*, vol. 2, pp. 375-378, 2011.
- [23] J. Xia and Y. Wang, "Secure Key Distribution for the Smart Grid," *IEEE Transactions on Smart Grid*, vol. 3, pp. 1437-1443, 2012.
- [24] J. H. Park, M. Kim, and D. Kwon, "Security Weakness in the Smart Grid Key Distribution Scheme Proposed By Xia and Wang," *IEEE Transactions on Smart Grid*, vol. 4, pp. 1613-1614, 2013.
- [25] J. L. Tsai and N. W. Lo, "Secure Anonymous Key Distribution Scheme for Smart Grid," *IEEE Transactions on Smart Grid*, vol. 7, pp. 906-914, 2016.
- [26] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably Secure Authenticated Key Agreement Scheme for Smart Grid," *IEEE Transactions on Smart Grid*, vol. 9, pp. 1900-1910, 2018.
- [27] P. Gope and B. Sikdar, "An Efficient Privacy-Preserving Authentication Scheme for Energy Internet-Based Vehicle-to-Grid Communication," *IEEE Transactions on Smart Grid*, vol. 10, pp. 6607-6618, 2019.
- [28] K. Park, Y. Park, A. K. Das, S. Yu, J. Lee, and Y. Park, "A Dynamic Privacy-Preserving Key Management Protocol for V2G in Social Internet of Things," *IEEE Access*, vol. 7, pp. 76812-76832, 2019.
- [29] L. F. A. Roman, P. R. L. Gondim, J. Lloret, "Pairing-Based Authentication Protocol for V2G Networks in Smart Grid," *Ad Hoc Networks*, vol. 90, pp. 1-16, 2019.
- [30] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi, and M. Shafiq, "A Provably Secure and Efficient Authenticated Key Agreement Scheme for Energy Internet-Based Vehicle-to-Grid Technology Framework," *IEEE Transactions on Industry Applications*, vol. 56, pp. 4425-4435, 2020.
- [31] M. Kaveh and M. R. Mosavi, "A Lightweight Mutual Authentication for Smart Grid Neighborhood Area Network Communications Based on Physically Unclonable Function," *IEEE Systems Journal*, vol. 14, pp. 4535-4544, 2020.
- [32] M. Saffkhani, N. Bagheri, S. Ali, M. H. Malik, O. H. Ahmed, M. Hosseinzadeh, and AM. H. Mosavi, "Improvement and Cryptanalysis of a Physically Unclonable Functions Based Authentication Scheme for Smart Grids," *mathematics*, vol. 56, p. 48, 2023.
- [33] G. Bansal, N. Naren, V. Chamola, B. Sikdar, N. Kumar, and M. Guizani, "Lightweight Mutual Authentication Protocol for V2G using Physical Unclonable Function," *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 7234-7246, 2020.
- [34] A. G. Reddy, P. R. Babu, V. Odelu, L. Wang, and S. A. Kumar, "V2G-Auth: Lightweight Authentication and Key Agreement Protocol for V2G Environment Leveraging Physically Unclonable Functions," *IEEE Transactions on Industrial Cyber-Physical Systems*, vol. 1, pp. 66-78, 2023.
- [35] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198-208, 1983.
- [36] R. Canetti and H. Krawczyk, "Universally Composable Notions of Key Exchange and Secure Channels," in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'02)*, Amsterdam, Netherlands, 2002, pp. 337-351.
- [37] S. Yu, A. K. Das, Y. Park, and P. Lorenz, "SLAP-IoD: Secure and Lightweight Authentication Protocol Using Physical Unclonable Functions for Internet of Drones in Smart City Environments," *IEEE Transactions on Vehicular Technology*, vol. 71, pp. 10374-10388, 2022.
- [38] M. N. Aman, K. C. Chua, and B. Sikdar, "Mutual Authentication in IoT Systems Using Physical Unclonable Functions," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1327-1340, 2017.
- [39] S. Yu and Y. Park, "A Robust Authentication Protocol for Wireless Medical Sensor Networks Using Blockchain and Physically Unclonable Functions," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20214-20228, 2022.
- [40] Y. Gao, S. F. A. Sarawi, and D. Abbott, "Physical Unclonable Functions," *Nature Electronics*, vol. 3, pp. 81-91, 2020.
- [41] K. B. Frikken, M. Blanton, and M. J. Atallah, "Robust Authentication Using Physically Unclonable Functions," in *International Conference on Information Security*, Pisa, Italy, 2009, pp. 262-277.
- [42] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's Law in Passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776-2791, 2017.
- [43] D. V. Oheimb, "The High-Level Protocol Specification Language HLPSSL Developed in the EU Project AVISPA," in *Proc. of the APPSEM 2005 Workshop*, Tallinn, Finland, 2005, pp. 1-17.
- [44] PBC Library, "Pairing-Based Cryptography Library," 2006, <https://crypto.stanford.edu/pbc/> (Accessed on 16 March 2022).
- [45] JCE Library, "Java Cryptography Extension Library," 2000, <https://oracle.com/java/technologies/javase-jce-all-downloads.html> (Accessed on 16 March 2022).



Sungjin Yu "received the M.S. degree in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2017, and 2019, respectively, where he is currently pursuing the Ph.D. degree with electronics and electrical engineering. He is currently a researcher with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His research interests include blockchain, authentication, VANET, FANET, Internet of Vehicles, Internet of Drones, information security, AI security, and Metaverse security".



Kisung Park “received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2015 and 2017, respectively. He also received a Ph.D. degree in electronic and electrical engineering from Kyungpook National University, in 2021. He was a researcher with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. He is currently an assistant professor with Gachon University, Seongnam, South Korea. His research interests include authentication, blockchain,

anonymous credentials, decentralized identifier, Internet of Things, post-quantum cryptography, VANET, information security, AI security, and Metaverse security”.