PUF-Enabled Key-Exchange Protocol for Vehicular Ad-Hoc Networks

Khalid Mahmood, *Senior Member, IEEE*, Zahid Ghaffar, Muhammad Farooq, Muhammad Ilyas, *Senior Member, IEEE*, Ashok Kumar Das, *Senior Member, IEEE*, and Shehzad Ashraf Chaudhry, *Senior Member, IEEE*

***Abstract*—The Internet of Vehicles (IoV) enables data exchange among individuals, cloud resources, road infrastructures, and vehicles, interconnected through Vehicular Ad Hoc Networks (VANETs). VANETs comprise vehicles with Onboard Units (OBUs), Roadside Units (RSUs), and a Trusted Party Agent (TPA). The data transmission among these entities supports seamless interaction and collaborative traffic management. However, data transmission on public communication channels in VANETs presents significant challenges, including security, privacy, and authentication of participating entities. Although numerous key exchange and authentication protocols have been introduced to tackle these issues, many protocols remain vulnerable to various attacks, such as a vehicle, RSU, TPA impersonation, denial of service, physical cloning, and desynchronization attacks. Therefore, to address these vulnerabilities, we propose a key exchange protocol that leverages hash functions and Advanced Encryption Standard (AES) encryption. Our protocol also integrates the Physical Unclonable Function (PUF), enhancing its resistance to physical or cloning attacks. Additionally, it effectively counters threats like impersonation, session key leakage, ephemeral secret leakage, and desynchronization attacks. We validate the security and reliability of our protocol through both formal and informal analysis. Informal analysis highlights the protocol's essential security features, while formal analysis provides robust substantiation. Performance evaluation reveals that our protocol achieves an average reduction of 35.53%, and 53.77%, in communication and computation overheads.**

***Index Terms*—Authentication protocol, mutual authentication, key-exchange protocol, key-management, vehicular ad-hoc network.**

Received 10 January 2025; revised 23 April 2025 and 5 August 2025; accepted 2 December 2025. Date of publication 11 December 2025; date of current version 6 April 2026. This work was supported by Abu Dhabi University's Office of Research and Sponsored Programs under Grant 19300938. The Associate Editor for this article was V. Chamola. *(Corresponding author: Shehzad Ashraf Chaudhry.)*

Khalid Mahmood is with the Graduate School of Intelligent Data Science, National Yunlin University of Science and Technology, Douliu, Yunlin 64002, Taiwan (e-mail: khalidm.research@gmail.com).

Zahid Ghaffar is with the Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology, Douliu 64002, Taiwan (e-mail: chzahid337@gmail.com).

Muhammad Farooq is with the Department of Mathematics and Statistics, College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates (e-mail: farooq-muhammad@hotmail.com).

Muhammad Ilyas is with the Department of Cybersecurity, College of Engineering, Al Ain University, Abu Dhabi, United Arab Emirates (e-mail: muhammad.ilyas@aau.ac.ae).

Ashok Kumar Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad 500032, India, and also with the Department of Computer Science and Engineering, College of Informatics, Korea University, Seongbuk-gu, Seoul 02841, South Korea (e-mail: ashok.das@iiit.ac.in).

Shehzad Ashraf Chaudhry is with the Department of Computer Science and Information Technology, College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates, and also with the Department of Software Engineering, Faculty of Engineering and Architecture, Istanbul Nisantasi University, 34398 Istanbul, Türkiye (e-mail: ashraf.shehzad.ch@gmail.com).

Digital Object Identifier 10.1109/TITS.2025.3640830

I. Introduction

THE Internet of Vehicles (IoV) is a network facilitating data exchange among individuals, cloud computing resources, road infrastructures, and vehicles. This network is established through specific communication protocols and standardized data exchange methods [1], [2]. It creates seamless connections between infrastructure, vehicles, human participants, and cloud environments for collaborative traffic management. In the IoV framework, Vehicular Ad Hoc Networks (VANETs) include vehicles equipped with Roadside Units (RSUs), Onboard Units (OBUs), Internet connectivity, and a Trusted Party Agent (TPA). OBUs facilitate data exchange among vehicles, while RSUs act as communication gateways, extending network coverage and enhancing connectivity to the Internet. The TPA ensures network security and integrity by managing authentication, cryptographic key distribution, and authorization, fostering trust among VANET participants. This interconnected system allows for intelligent and collaborative traffic management, promoting efficiency and safety in transportation [3], [4]. VANET enhances communication and coordination among diverse stakeholders by leveraging extensive data resources, ensuring a robust and secure vehicular network [5].

Despite the potential of VANET architecture to enhance information transmission and data convergence, the current communication framework faces significant security challenges. Wireless communication in VANETs is particularly vulnerable due to high vehicle mobility, dynamic topology changes, and the broadcast nature of safety messages, which increase the risk of real-time interception, impersonation, and message tampering [6], [7]. These vulnerabilities can lead to unauthorized access and alteration of critical vehicle-related data, such as speed, location, and personal information, posing severe and potentially catastrophic consequences. Consequently, addressing these security issues is essential to maintaining data integrity within VANETs [8], [9]. These unique challenges require authentication protocols that are

TABLE I
SUMMARY OF EXISTING RELATED WORK

Authors	Year	Methodology and Techniques	Advantage(s)	Limitation(s)
Awais et al. [11]	2024	Hash Function, Fuzzy Extractor	Resists impersonation, session key leakage, and replay attacks	Vulnerable to physical tampering or cloning and desynchronization attacks
Tomar et al. [12]	2024	Chaotic Maps, Hash Function, and Fuzzy Extractor	Offers anonymity and resists impersonation, and replay attacks	Vulnerable to ephemeral secret leakage, desynchronization, and physical tampering or cloning
Awais et al. [13]	2023	Hash Function, Physical Unclonable Function (PUF)	Resists denial of service, session key leakage, and replay attacks	Vulnerable to ephemeral secret leakage attacks and does not offer vehicle location privacy
Kumar et al. [14]	2023	Elliptic Curve Cryptography (ECC), Hash Function	Resists impersonation and physical or cloning attacks and offers anonymity and location privacy	Vulnerable to desynchronization and ephemeral secret leakage attacks
Vinod Kumar [15]	2023	ECC, Hash Function	Resists denial of service, session key leakage, and replay attacks	Vulnerable to user impersonation and smart card stolen attacks
Xie et al. [16]	2023	PUF and ECC	Resists session key leakage, guessing, and tampering attacks	Susceptible to impersonation attacks and violates vehicle anonymity
Wang et al. [17]	2022	ECC and Hash Function	Resists password guessing and impersonation attacks	Prone to the session key leakage attacks and does not offer anonymity
Kumar et al. [18]	2022	ECC and Hash Function	Resists impersonation attacks and offers vehicle anonymity	Does not ensure perfect forward secrecy and also vulnerable to physical or cloning attacks
Safkhani et al. [19]	2021	ECC and Hash Function	Resists impersonation attacks and offers vehicle anonymity	Does not offer perfect forward secrecy and prone to physical or cloning attacks
Nandy et al. [20]	2021	Fuzzy Extractor and ECC	Resists impersonation and password guessing attacks	Violates vehicle anonymity and prone to physical security
Wu et al. [21]	2021	Hash Function and ECC	Resists impersonation attacks and offers mutual authentication	Lacks conditional privacy and requires high computation cost

both lightweight and capable of maintaining security under rapidly changing network conditions [10].

Effective authentication systems are crucial for verifying the legitimacy of vehicles and RSUs, securing connections, and protecting against malicious entities. These robust authentication procedures are essential for fostering secure communication among all relevant entities, instilling confidence, and effectively mitigating the risks of unauthorized access and data breaches within the VANET environment. The VANET infrastructure ensures secure and reliable data transmission by implementing a robust key-exchange protocol. This approach protects sensitive information and enhances the overall reliability and trustworthiness of the VANETs. As VANETs continue to evolve, prioritizing security is crucial for realizing their full potential in improving traffic management and transportation safety [22], [23].

Numerous key exchange protocols have been introduced by researchers to enhance security and privacy, as detailed in Table I. However, many of these protocols still face significant security issues, including vulnerabilities to impersonation, physical or cloning, replay, denial of service, and desynchronization attacks. This underscores the need for further research and advancements in designing key exchange protocols to achieve more secure and reliable communication in VANETs. Although numerous authentication and key agreement protocols have also been proposed for other networks such as IoT, IoV, and IoD, VANETs and their extensions pose novel challenges due to their highly dynamic topology, rapid vehicle mobility, and strict latency requirements for safety-critical messaging. These constraints demand authentication protocols that not only ensure strong security but also operate efficiently under time-sensitive and mobility-intensive conditions.

To address and bridge the existing research gaps, we propose a secure and lightweight authentication and key-exchange protocol for VANETs. The protocol integrates Physical Unclonable Functions (PUFs) and Advanced Encryption Standards (AES) to enhance both security and efficiency. Unlike traditional or existing protocols, our protocol leverages the hardware-level uniqueness of PUFs to resist physical tampering, cloning, and desynchronization attacks. Additionally, it employs lightweight cryptographic operations to reduce computational and communication overhead, making it suitable for resource-constrained vehicular environments. The protocol also establishes dynamic session keys and mutual authentication, ensuring strong protection against replay, impersonation, and man-in-the-middle attacks. These design choices bridge the limitations of existing solutions and offer a robust framework for secure, scalable, and real-time communication among vehicles, RSUs, and the trusted authority in dynamic VANET environments.

While several context-aware authentication schemes have also been introduced, such as CAPS for VANET safety applications [24], SoftAuthZ for behavior-based IoT authorization [25], and an application-aware authentication protocol for the IoD [26]. These works primarily focus on tailoring authentication-based on message types, environmental context, or behavioral profiles. Although effective in their respective domains, they do not directly address the stringent real-time

security demands of high-mobility vehicular environments. In contrast, our protocol is designed to operate independently of the application context, providing lightweight, tamper-resistant, and scalable authentication across varying vehicular scenarios. By integrating PUFs and efficient cryptographic mechanisms, the proposed protocol offers broader applicability and enhanced protection against physical and session-based attacks, making it a robust fit for latency-sensitive and dynamic VANET environments.

A. Motivation and Contributions

VANETs are vulnerable to various security attacks due to public communication channels. These limitations enable adversaries to access and tamper with critical data, leading to operational disruptions and security attacks. Recently, researchers have introduced numerous key exchange protocols to enhance secure communication, reliability, and efficiency for VANETs. However, these existing protocols often fail to meet the specific security requirements of VANETs. Moreover, the high computational demands of these existing protocols make them impractical for real-world VANET applications. Additionally, vehicles and RSUs in VANETs are highly susceptible to physical and cloning attacks. Therefore, this paper proposes a PUF-enabled key exchange protocol for VANETs to meet all the required security features. The main contributions of this article are outlined as follows:

- We introduced a key exchange protocol that leverages AES and hash functions. The proposed protocol strengthens the VANETs against potential security threats, ensuring data integrity and security across VANETs.
- We also integrate a PUF to enhance resistance against physical attacks on vehicles and RSUs, which is a key aspect of the novelty of our approach. Unlike existing schemes that rely solely on stored cryptographic keys, our use of hardware-intrinsic PUFs provides device-specific authentication that is inherently unclonable and tamper-evident. This integration significantly strengthens security by preventing cloning and physical tampering, ensuring that only legitimate, untampered vehicles and RSUs can participate in the network.
- To preserve user anonymity, we utilize AES to compute the masked identity due to its computational efficiency and proven security.
- Employing lightweight cryptographic operations helps us achieve resource efficiency. Consequently, the proposed protocol demonstrates it by reducing computation and communication overhead by 53.77%, and 35.53%,, respectively.

II. Preliminaries

This section discusses the preliminary elements, which include threat model, system model, physically unclonable function and notation guide.

A. Threat Model

The Dolev-Yao (DY) [27] and Canetti-Krawczyk (CK) [28] threat models are realized in this article to evaluate and ensure the robustness of devised protocols against various types of potential attacks. The DY model [27] defines the adversary as having full control over the communication network and being able to intercept, modify, delete, and inject messages. The CK model [28] builds on additional capabilities over the DY model, focusing on key exchange protocols and considering additional threats such as MITM, replay and session hijacking attacks. Together, these models comprehensively evaluate cryptographic protocols against a wide range of network-based attacks.

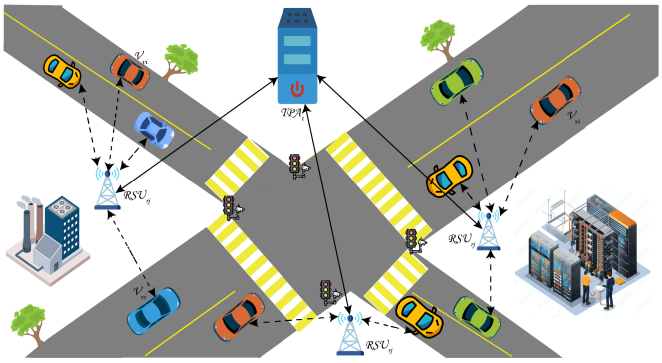

Fig. 1. Architecture diagram for PUF enabled key exchange protocol.

B. System Model

Our Vehicular Ad Hoc Network (VANET) system model, depicted in Figure 1, consists of three well-defined entities working collaboratively to address security challenges: vehicles ($\mathcal{V}_{vi}$), Roadside Units ($\mathcal{RSU}_{rj}$) and a Trusted Party Agent ($\mathcal{TPA}_t$).

Each vehicle, equipped with a Physical Unclonable Function (PUF), generates unique cryptographic keys for secure communication. During the initialization phase, vehicles register with the $\mathcal{TPA}_t$ through nearby $\mathcal{RSU}_{rj}$ to obtain credentials, ensuring only authenticated devices join the network. The $\mathcal{RSU}_{rj}$ serve as critical intermediaries, relaying time-bound session keys from the $\mathcal{TPA}_t$ to vehicles while aggregating and validating traffic data (e.g., location, speed) before forwarding it to the $\mathcal{TPA}_t$. These $\mathcal{RSU}_{rj}$'s also broadcast $\mathcal{TPA}_t$ verified safety alerts to vehicles, creating a secure channel for emergency notifications. The $\mathcal{TPA}_t$ acts as the central authority, managing entity enrollment, key distribution, and real-time threat detection by analyzing data from $\mathcal{RSU}_{rj}$ to identify anomalies like spoofing attacks. This structured interaction model ensures secure communication at every stage, from initial vehicle authentication to dynamic key updates and emergency broadcasts, while explicitly addressing spoofing, man-in-the-middle attacks, and unauthorized network access.

C. Computational Model for RSU Localization

This section introduces the positional transformation formula and essential assumptions used for $\mathcal{RSU}_{rj}$ location calculation. During the authentication, $\mathcal{RSU}_{rj}$ shares its $\mathcal{X}_{NP}$ and $\mathcal{Y}_{NP}$ with $\mathcal{TPA}_t$. To determine the precise value, $\mathcal{TPA}_t$ computes: $\mathcal{X}_{NP} = \mathcal{V}_p * Cos\theta + \mathcal{X}_{rj}$, $\mathcal{Y}_{NP} = \mathcal{V}_p * Sin\theta + \mathcal{Y}_{rj}$

and checks for $\mathcal{X}_{NP}$ and $\mathcal{Y}_{NP}$ against LI_{rj}. Here, $\mathcal{X}_{NP}$ and $\mathcal{Y}_{NP}$ represents the next approximate X ad Y coordinates of $\mathcal{RSU}_{rj}$. In contrast, $\mathcal{V}_p$ and θ symbolize the velocity of $\mathcal{V}_{vi}$ and $\mathcal{RSU}_{rj}$'s angle of the vector within a horizon line. The location verification check is verified if $\mathcal{X}_{NP}$ and $\mathcal{Y}_{NP}$ align with the values of LI_{rj}.

D. Physical Unclonable Function

In this work, we utilize an SRAM-based Physical Unclonable Function (PUF) design, as outlined in the reference [29]. Our mathematical framework for PUFs is built upon this established design, ensuring the robustness of the challenge-response mechanism and facilitating its use in secure applications. Primarily, an SRAM-based PUF operates as a distinctive mathematical relationship that transforms a set of challenges, $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$, into a corresponding set of responses $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$. This process relies on inherent physical properties, ensuring both uniqueness and unpredictability. The function is formally expressed as:

$$\mathcal{F} : \mathcal{C} \rightarrow \mathcal{R}$$

where each challenge $\mathcal{C}$ generates a specific response $\mathcal{R}$. For a challenge input $\mathcal{C}$, the PUF generates a corresponding response $\mathcal{R}$ as:

$$\mathcal{R} = \text{PUF}(\mathcal{C})$$

where PUF$(\cdot)$ denotes the unique physical characteristics of each PUF instance. The response $\mathcal{R}$ is then employed in cryptographic operations to enable secure authentication and efficient key management within the proposed protocol.

III. Proposed Protocol

This section provides a comprehensive overview of our PUF-enabled key exchange protocol designed to thwart physical and cyber attacks in VANETs. Our protocol leverages PUF, AES symmetric encryption/decryption, and hash functions. We implement Static Random-Access Memory (SRAM) PUF using monostable SRAM cells, which inherently tolerate potential noise in the PUF's output, eliminating the need for additional error-tolerance techniques. Additionally, we deploy PUF at RSUs, which are unmanned and susceptible to tampering attacks. The following subsections offer a detailed review of the developed protocols.

A. Vehicle Registration

$\mathcal{TPA}_t$ initiates the vehicle registration process selecting a unique identifier ID_{vi} for the vehicle. It then generates a challenge-response pair $< \mathcal{C}_{vi}, \mathcal{R}_{vi} >$ and exploits $\mathcal{GEN}(.)$ to determine ξ_{vi} for noise handling in succeeding phases. $\mathcal{TPA}_t$ computes $\alpha_{vi} = h(ID_{vi}||\mathcal{K}_t)$. Next, it calculates $\omega_{vi} = h(\alpha_{vi}||\mathcal{R}_{vi}||\mathcal{K}_t)$. Additionally, it creates a masked identifier $\mathcal{MID}_{vi}^{new} = Enc_{\mathcal{K}_t}(ID_{vi}||\mu_t||\alpha_{vi}||\mathcal{R}_{vi})$. The calculated values $< ID_{vi}, \alpha_{vi}, \omega_{vi}, \mathcal{C}_{vi}, \mathcal{MID}_{vi} >$ are then written into the vehicle's $\mathcal{OBU}_i$, completing the registration phase.

B. RSU Registration

During the registration phase of $\mathcal{RSU}_{rj}$, the $\mathcal{TPA}_t$ selects a unique identifier ID_{rj} for $\mathcal{RSU}_{rj}$ and provides the location information LI_{rj} where it is supposed to be installed. It then generates a challenge-response pair $< \mathcal{C}_{rj}, \mathcal{R}_{rj} >$ exploits $\mathcal{GEN}(.)$ to determine ξ_{rj} for noise handling in succeeding phases. $\mathcal{RSU}_{rj}$ then gets $\alpha_{rj} = h(ID_{rj}||\mathcal{K}_t)$. Next, $\mathcal{TPA}_t$ calculates $\omega_{rj} = h(\alpha_{rj}||\mathcal{R}_{rj}||\mathcal{K}_t)$. Additionally, it creates a ciphertext $CT_{rj} = ENC_{\mathcal{K}_t}\{\mathcal{R}_{rj}, \alpha_{rj}, LI_{rj}\}$. This ciphertext is stored in the database against ID_{rj}, and the calculated values $< \alpha_{rj}, \omega_{rj}, \mathcal{C}_{rj}, \mathcal{ID}_{rj} >$ are shared with $\mathcal{RSU}_{rj}$.

C. Authentication and Key Exchange

Our authentication protocol is specifically designed to securely exchange a session key between $\mathcal{V}_{vi}$, $\mathcal{RSU}_{rj}$, and $\mathcal{TPA}_t$. The protocol follows a detailed procedure to ensure the security and integrity of the communication among these entities. A detailed overview of our protocol is displayed in Figure 2 while a brief description is given below:

- To constructs an initial login message $\mathbb{MSG}_1$, $\mathcal{V}_{vi}$ generates $\mu_1 \in \mathbb{Z}_q^*$ and computes: $\mathcal{V}_1 = ID_{vi} \oplus \mu_1$, $\mathcal{R}_{vi} = \mathcal{REP}(PUF_{vi}(\mathcal{C}_{vi}), \xi_{vi})$ and $\mathcal{V}_2 = h(ID_{vi}||ID_{rj}||\mathcal{R}_{vi}||\omega_{vi}||\mu_1)$. This message is then sent to the $\mathcal{RSU}_{rj}$.
- Upon receiving $\mathbb{MSG}_1$, $\mathcal{RSU}_{rj}$ firstly furnishes its $\mathcal{X}_{vj}$ and $\mathcal{Y}_{vj}$ coordinates. $\mathcal{RSU}_{rj}$ then construct a message $\mathbb{MSG}_2$ by computing $\mathcal{R}_{rj} = \mathcal{REP}(PUF_{rj}(\mathcal{C}_{rj}), \xi_{rj})$, $\mathcal{V}_3 = \omega_{rj} \oplus (\mu_2||\mathcal{X}_{rj}||\mathcal{Y}_{rj})$ and $\mathcal{V}_4 = h(ID_{rj}||\omega_{rj}||\mathcal{R}_{rj}||\mu_2)$. Next, $\mathcal{RSU}_{rj}$ forwards $\mathbb{MSG}_2$ to $\mathcal{TPA}_k$.
- On receiving $\mathbb{MSG}_2$, $\mathcal{TPA}_k$ computes: $(ID_{vi}||\mu_t||\alpha_{vi}||\mathcal{R}_{vi}) = Dec_{\mathcal{K}_t}(\mathcal{MID}_{vi})$, $\mu_1 = ID_{vi} \oplus \mathcal{V}_1$, $\omega_{vi} = h(\alpha_{vi}||\mathcal{R}_{vi}||\mathcal{K}_t)$ and verifies: $\mathcal{V}_2' \stackrel{?}{=} h(ID_{vi}||ID_{rj}||\mathcal{R}_{vi}||\omega_{vi}||\mu_1)$. If it is equal, $\mathcal{TPA}_k$ finds $\{\mathcal{R}_{rj}, LI_{rj}\}$ against ID_{rj} and computes: $\omega_{rj} = h(ID_{rj}||\mathcal{K}_t||\mathcal{R}_{rj})$, $(\mu_2||\mathcal{X}_{rj}||\mathcal{Y}_{rj}) = \mathcal{V}_3 \oplus \omega_{rj}$, $\mathcal{X}_{NP} = \mathcal{V}_p * Cos\theta + \mathcal{X}_{rj}$ and $\mathcal{Y}_{NP} = \mathcal{V}_p * Sin\theta + \mathcal{Y}_{rj}$. Afterwards, $\mathcal{TPA}_k$ checks for $\mathcal{X}_{NP}$ and $\mathcal{Y}_{NP}$ against LI_{rj}. $\mathcal{TPA}_k$ then authenticates $\mathcal{RSU}_{rj}$ by evaluating $\mathcal{V}_4' \stackrel{?}{=} h(ID_{rj}||\omega_{rj}||\mathcal{R}_{rj}||\mu_2)$. If it returns true, $\mathcal{TPA}_k$ further generates $\mu_t^{new} \in \mathbb{Z}_q^*$ and computes $SK = h(ID_{rj}||ID_t||\mu_2||\mu_t^{new})$, $\mathcal{MID}_{vi}^{new} = Enc_{\mathcal{K}_t}(ID_{vi}||\mu_t^{new}||\alpha_{vi}||\mathcal{R}_{vi})$, $\mathcal{V}_5 = \mu_2 \oplus \mu_t^{new}$, $\mathcal{V}_6 = \mathcal{R}_{vi} \oplus (\mathcal{MID}_{vi}^{new}||\mu_2||\mu_t^{new})$, $\mathcal{V}_7 = h(ID_{rj}||\omega_{rj}||\mu_2||SK)$ and $\mathcal{V}_8 = h(ID_{vi}||\omega_{vi}||\mu_1||\mathcal{R}_{vi}||SK)$. Finally, $\mathcal{TPA}_k$ sends $\mathbb{MSG}_3$ to $\mathcal{RSU}_{rj}$.
- On getting $\mathbb{MSG}_3$, $\mathcal{RSU}_{rj}$ computes: $\mu_t^{new} = \mu_2 \oplus \mathcal{V}_5$, $SK = h(ID_{rj}||ID_t||\mu_2||\mu_t^{new})$ and verifies $\mathcal{V}_7' \stackrel{?}{=} h(ID_{rj}||\omega_{rj}||\mu_2||SK)$. Upon successful verification of the $\mathcal{TPA}_t$, $\mathcal{RSU}_{rj}$ sends $\mathbb{MSG}_4$ to $\mathcal{V}_{vi}$.
- $\mathcal{V}_{vi}$ computes: $(\mathcal{MID}_{vi}^{new}||\mu_2||\mu_t^{new}) = \mathcal{R}_{vi} \oplus \mathcal{V}_6$, $SK = h(ID_{rj}||ID_t||\mu_2||\mu_t^{new})$, and checks for the equality $\mathcal{V}_8' \stackrel{?}{=} h(ID_{vi}||\omega_{vi}||\mu_1||\mathcal{R}_{vi}||SK)$. Once $\mathcal{V}_{vi}$ successfully verifies $\mathcal{RSU}_{rj}$ and establishes a session key, it can start exchanging beacon messages with $\mathcal{TPA}_t$.

IV. Security Analysis

This section solicits security analysis of the proposed key-exchange protocol for VANETs, employing informal and

formal methods to confirm its robustness against all potential security threats.

A. Informal Analysis

The proposed key-exchange protocol is analyzed using informal analysis on the basis of logical reasoning and threat modeling as discussed in [30] and [31]. We perform an informal analysis to check how the proposed protocol resists potential attacks and offers security features.

1) Offers $\mathcal{V}_{vi}$ Anonymity and Untraceability: In the devised protocol, $\mathcal{V}_{vi}$ keeps its ID_{vi} private. The transmitted public message $\mathbb{MSG}_1 = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{MID}_{vi}\}$, where $\mathcal{MID}_{vi} = Enc_{\mathcal{K}_t}(ID_{vi}||\mu_t||\alpha_{vi}||\mathcal{R}_{vi})$. This message $\mathbb{MSG}_1$ does not reveal any information about ID_{vi} to $\mathring{\mathcal{A}}$. This ensures that $\mathring{\mathcal{A}}$ cannot deduce the real ID_{vi} from $\mathbb{MSG}_1$. Moreover, if $\mathring{\mathcal{A}}$ observes different communication sessions, it cannot correlate multiple sessions to trace $\mathcal{V}_{vi}$ because $\mathcal{MID}_{vi}^{new}$ is dynamic and unique to each session. $\mathcal{MID}_{vi}^{new}$ is updated at the end of each session establishment. Thus, our protocol ensures anonymity and untraceability.

2) Resists $\mathcal{V}_{vi}$ Impersonation Attacks: Å may try to impersonate $\mathcal{V}_{vi}$ by forging the message $\mathbb{MSG}_1 = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{MID}_{vi}\}$. This requires having the actual ID_{vi}, and ω_{vi} to create a valid $\mathbb{MSG}_1$. However, since ID_{vi}, and ω_{vi} are neither stored nor shared publicly, $\mathring{\mathcal{A}}$ cannot obtain these parameters. Therefore, it is impossible to accurately generate real $\mathbb{MSG}_1$ for $\mathring{\mathcal{A}}$. It affirms the robust defense of the protocol against impersonation attacks in $\mathcal{V}_{vi}$.

3) Resists $\mathcal{RSU}_{rj}$ Impersonation Attacks: In the proposed protocol, if $\mathring{\mathcal{A}}$ attempts to create the message $\mathbb{MSG}_2 = \{\mathbb{MSG}_1, ID_{rj}, \mathcal{V}_3, \mathcal{V}_4\}$ to carry out an $\mathcal{RSU}_{rj}$ impersonation attack, it must determine $\mathcal{V}_3$ and $\mathcal{V}_4$. For this, $\mathring{\mathcal{A}}$ needs the ω_{rj} and $\mathcal{R}_{rj}$ for corresponding $\mathcal{RSU}_{rj}$. Since these parameters are not available to $\mathring{\mathcal{A}}$, the proposed protocol effectively resists $\mathcal{RSU}_{rj}$ impersonation attacks.

4) Resists $\mathcal{TPA}_t$ Impersonation Attacks: If $\mathring{\mathcal{A}}$ wants to impersonate $\mathcal{TPA}_t$, it needs to generate a valid message $\mathbb{MSG}_3 = \{\mathcal{V}_5, \mathcal{V}_6, \mathcal{V}_7, \mathcal{V}_8\}$. This message requires the actual values of ID_{vi} and ω_{rj}. Furthermore, generating these values also requires the master key $\mathcal{K}_t$, which is exclusive to $\mathcal{TPA}_t$ and is inaccessible to $\mathring{\mathcal{A}}$. Consequently, these rigorous requirements effectively protect the protocol against $\mathcal{TPA}_t$ impersonation attacks.

5) Resists Physical or Cloning Attacks: If $\mathring{\mathcal{A}}$ compromises $\mathcal{V}_{vi}$ or $\mathcal{RSU}_{rj}$ and attempts to alter or clone them, the integrated PUF changes its response, making tampering ineffective. During protocol execution, a compromised PUF fails to generate the correct $\mathcal{R}_{vi}$ or $\mathcal{R}_{rj}$, which are essential for verification. Consequently, $\mathcal{TPA}_t$ can detect and prevent these intrusion attempts by $\mathring{\mathcal{A}}$, demonstrating the protocol's resilience to physical or cloning attacks.

6) Resists Desynchronization Attacks: The $\mathcal{TPA}_t$ records the $\mathcal{RSU}_{rj}$'s parameters exchanged during its registration phase. Conventionally, these parameters are stored in the verifier table along with pseudonyms, allowing the $\mathcal{TPA}_t$ to retrieve them during authentication using the pseudonym provided by $\mathcal{RSU}_{rj}$. After each successful authentication round, the $\mathcal{TPA}_t$ must update the pseudonym and share the updated value with $\mathcal{RSU}_{rj}$ to prevent traceability attacks. Network failures can pose challenges in updating and sharing pseudonyms, potentially leading to desynchronization. This desynchronization occurs when the $\mathcal{TPA}_t$ has a pseudonym that differs from the one in $\mathcal{RSU}_{rj}$'s memory. To avoid such issues, our protocol does not use pseudonyms. Instead, we store data at the $\mathcal{TPA}_t$ against the actual ID_{rj} to efficiently derive $\mathcal{RSU}_{rj}$'s credentials. This eliminates the need to update and share pseudonyms, making our protocol resistant to desynchronization attacks.

7) Resists Ephemeral Secret Leakage (ESL) Attacks: If $\mathring{\mathcal{A}}$ obtains the ephemeral secrets for that session, then he can reconstruct the session key $SK = h(ID_{rj}||ID_t||\mu_2||\mu_t^{new})$. In our protocol, SK is derived from ID_{rj}, ID_t, and the ephemeral secrets μ_2 and μ_t^{new}. These secrets remain private and are never transmitted publicly. Furthermore, the use of a hash function ensures that $\mathring{\mathcal{A}}$ cannot reconstruct the session key. Even in the worst-case scenario where $\mathring{\mathcal{A}}$ obtains the session key for a specific session, it does not affect any previously established session keys. This is because each session key is computed using fresh, session-specific values, and no long-term secrets are reused across sessions. As a result, the compromise of one session does not enable the adversary to infer or derive the session keys of any past or future communications. Consequently, the proposed key exchange protocol resists ephemeral secret leakage attacks.

8) Resists Stolen Verifier Attacks: In a stolen verifier attack, $\mathring{\mathcal{A}}$ aims to retrieve secret credentials by compromising the verifier's stored information. However, in the devised protocol, the $\mathcal{TPA}_t$ does not maintain any verifier table or store static credentials related to $\mathcal{V}_{vi}$ or $\mathcal{RSU}_{rj}$. Instead, session-specific values, such as ω_{vi}, ω_{rj}, and PUF-generated responses, are dynamically determined during the execution of the protocol. Furthermore, even if $\mathring{\mathcal{A}}$ manages to extract stored data such as $\mathcal{MID}_{vi}$ or CT_{rj}, these are encrypted using the long-term symmetric key $\mathcal{K}_t$ and do not reveal useful information without breaking the AES encryption. Hence, without having reusable verifiers and the use of PUFs and encryption, collectively ensures strong resistance against stolen verifier attacks.

9) Ensures Perfect Forward Secrecy: Perfect forward secrecy ensures that compromise of long-term keys does not reveal previous session keys. Our protocol ensures it by generating session keys using fresh and dynamic random numbers (μ_1, μ_2, μ_t^{new}) for each session. Specifically, the session key $SK = h(ID_{rj}||ID_t||\mu_2||\mu_t^{new})$ is determined using freshly selected nonces that are not reused or stored. Therefore, even if the long-term key $\mathcal{K}_t$ is later compromised, it does not help to reconstruct any previous session keys. Moreover, session-specific randomness combined with hash functions ensures that each session remains cryptographically isolated. It fulfills the requirement for perfect forward secrecy.

B. Formal Analysis

This section outlines the formal security validation for our proposed protocol using the random oracle model [32], [33], focusing on proving its semantic security.

1) Participants and Partnering: The interaction involves entities such as $\mathcal{V}_{vi}$, $\mathcal{RSU}_{rj}$ and $\mathcal{TPA}_t$. Each participant instance is labeled as $Fm^k_{\mathcal{V}_{vi}}$ for vehicles, $Fm^k_{\mathcal{RSU}_{rj}}$ for roadside units and $Fm^k_{\mathcal{TPA}_t}$ for trusted party. Participant is marked 'Accept' if it successfully processes a correct message. Two participants are considered partnered if they share the same session key, session identities, and participant identities.

2) Queries: The queries model the potential actions of attackers.

- $Execute(Fm^k_{\mathcal{V}_{vi}}, Fm^k_{\mathcal{RSU}_{rj}}, Fm^k_{\mathcal{TPA}_t})$: $\mathring{\mathcal{A}}$ can intercept all messages transmitted through public channel.
- $Send(Fm^k_{\mathcal{V}_{vi}}, Fm^k_{\mathcal{RSU}_{rj}}, Fm^k_{\mathcal{TPA}_t}, \mathbb{MSG})$: $\mathring{\mathcal{A}}$ creates and sends a message $\mathbb{MSG}$ to $Fm^k_{\mathcal{V}_{vi}}$, $Fm^k_{\mathcal{RSU}_{rj}}$, or $Fm^k_{\mathcal{TPA}_t}$. If the message $\mathbb{MSG}$ is correct, $Fm^k_{\mathcal{V}_{vi}}$, $Fm^k_{\mathcal{RSU}_{rj}}$, or $Fm^k_{\mathcal{TPA}_t}$ will answer to $\mathring{\mathcal{A}}$.
- $Reveal(Fm^k_{\mathcal{V}_{vi}}, Fm^k_{\mathcal{RSU}_{rj}}, Fm^k_{\mathcal{TPA}_t})$: The attacker $\mathring{\mathcal{A}}$ can obtain the session keys shared among $Fm^k_{\mathcal{V}_{vi}}$, $Fm^k_{\mathcal{RSU}_{rj}}$ and $Fm^k_{\mathcal{TPA}_t}$.
- $Test(Fm^k_{\mathcal{V}_{vi}}, Fm^k_{\mathcal{RSU}_{rj}}, Fm^k_{\mathcal{TPA}_t}, r_b)$: $\mathring{\mathcal{A}}$ can run this query only once. It generates a random bit r_b. If $r_b = 1$, it returns the real session key; otherwise, it returns an arbitrary number.
- $Corrupt(Fm^k_{\mathcal{V}_{vi}})$: This simulates a side-channel attack on the OBUi and retrieves the stored information: $\mathcal{MID}^{new}_{vi} = Enc_{\mathcal{K}_t}(ID_{vi}||\mu_t||\alpha_{vi}||\mathcal{R}_{vi})$.
- $Corrupt\mathcal{RSU}_{rj}(Fm^k_{\mathcal{RSU}_{rj}})$: This simulates an attack where $\mathring{\mathcal{A}}$ captures the $\mathcal{RSU}_{rj}$ and retrieves the stored information $CT_{rj} = ENC_{\mathcal{K}_t}\{\mathcal{R}_{rj}, \alpha_{rj}, LI_{rj}\}$.

3) Freshness: You can consider an instance as fresh if it meets these conditions:

1) $Fm^k_{\mathcal{V}_{vi}}$, $Fm^k_{\mathcal{RSU}_{rj}}$ and $Fm^k_{\mathcal{TPA}_t}$ are in the accept state.
2) No one has executed the $Reveal(Fm^k_{\mathcal{V}_{vi}}, Fm^k_{\mathcal{RSU}_{rj}}, Fm^k_{\mathcal{TPA}_t})$ query.
3) The $Corrupt(Fm^k_{\mathcal{V}_{vi}})$ and $Corrupt\mathcal{RSU}_{rj}(Fm^k_{\mathcal{V}_{vi}})$ queries have been executed no more than once.

4) Accepted State: Fm^k enters the accepted state after the last message exchange is completed.

5) Adversary ($\mathring{\mathcal{A}}$): In this model, $\mathring{\mathcal{A}}$'s task is constrained to querying the Oracles and operating the simulator. Taking advantage of the gathered responses, $\mathring{\mathcal{A}}$ aims to break the protocol. The above-listed queries are the ones $\mathring{\mathcal{A}}$ can initiate.

6) Semantic Security: An $\mathring{\mathcal{A}}$ is allowed to execute this only once.

$\mathring{\mathcal{A}}$ can use the Test query and multiple other queries to check if the returned value is correct or not. This means that $\mathring{\mathcal{A}}$ tries to guess the random bit r_b generated by the Test query. The probability is represented as $(Advg^P_{\mathring{\mathcal{A}}} = |2Pr[\text{suc}(A)] - 1|)$. If $(Advg^P_{\mathring{\mathcal{A}}})$ is less than η, where η is a very small value, the protocol is considered secure.

Theorem 1: The advantage that $\mathring{\mathcal{A}}$ has in obtaining the session key within polynomial time is expressed as:

$$Advg^P_{\mathring{\mathcal{A}}} \leq \frac{q^2_{HSH}}{2^{l_{HSH}}} + \frac{(q_{SND} + q_{EXE})^2}{n} + 2q_{SND}Advg^{PUF}_{\mathring{\mathcal{A}}}$$

where q_{HSH}, q_{SND}, and q_{EXE} represent that how many times these operations are performed. l_{HSH} and n denote the length of hash output and transcripts, respectively. The advantage of $\mathring{\mathcal{A}}$ to break the PUF are represented by $Adv^{PUF}_{\mathring{\mathcal{A}}}$.

Proof: The games Game_a (where $0 \leq a \leq 3$) are designed to simulate attacks launched by $\mathring{\mathcal{A}}$. Win_a (for $0 \leq a \leq 4$) indicates that $\mathring{\mathcal{A}}$ successfully guesses the random bit r_b in Game_a. The descriptions of these games is defined as follows:

$Game_0$: This game imitate the actual attack initially carried out by $\mathring{\mathcal{A}}$. Based on this definition, we obtain the following:

$$Advg^P_{\mathring{\mathcal{A}}} = |2\Pr[\text{Sucss}_0] - 1| \tag{1}$$

$Game_1$: The eavesdropping attack is simulated in this game. $\mathring{\mathcal{A}}$ intercepts all messages that are publicly transmitted. Furthermore, $\mathring{\mathcal{A}}$ attempts to guess the random bit r_b. However, due to the properties of the hash function and secret credentials i.e.$(\mathcal{MID}_{vi}, \mathcal{V}_1, \mathcal{R}_{vi})$, $\mathring{\mathcal{A}}$ is not capable to know the relationship of intercepted messages and the session keys. Therefore, we get:

$$\Pr[\text{Sucss}_0] = \Pr[\text{Sucss}_1] \tag{2}$$

$Game_2$: This game imitate a collision attack on the transcripts and hash results. According to the birthday paradox, the chance of a hash collision happening is less than $\frac{q^2_{HSH}}{2^{l_{HSH}+1}}$, and the chance of collisions in other transcripts is less than $\frac{(q_{SND}+q_{EXE})^2}{2n}$. Therefore, we conclude:

$$\Pr[\text{Sucss}_2] - \Pr[\text{Sucss}_1] \leq \frac{q^2_{HSH}}{2^{l_{HSH}+1}} + \frac{(q_{SND} + q_{EXE})^2}{2n} \tag{3}$$

$Game_3$: This game shows how an $\mathring{\mathcal{A}}$ carries out the $Corrupt$ operations on $(Fm^k_{\mathcal{V}_{vi}})$, and $Corrupt\mathcal{RSU}_{rj}(Fm^k_{\mathcal{RSU}_{rj}})$ to access the stored secret credentials $\mathcal{MID}^{new}_{vi} = Enc_{\mathcal{K}_t}(ID_{vi}||\mu_t||\alpha_{vi}||\mathcal{R}_{vi})$ in the OBUi and $CT_{rj} = ENC_{\mathcal{K}_t}\{\mathcal{R}_{rj}, \alpha_{rj}, LI_{rj}\}$ in the RSU. Here, α_{vi} is calculated as $\alpha_{vi} = h(ID_{vi}||\mathcal{K}_t)$ and ω_{rj} equals $h(\alpha_{rj}||\mathcal{R}_{rj}||\mathcal{K}_t)$. If $\mathring{\mathcal{A}}$ wants to get these important parameters, he must either guess α_{vi}, ω_{rj} or crack PUF. Assume that the likelihood of $\mathring{\mathcal{A}}$ breaking PUF is $Advg^{PUF}_{\mathring{\mathcal{A}}}$. Therefore, we get the following result:

$$\Pr[\text{Sucss}_3] - \Pr[\text{Sucss}_2] \leq q_{SND}\left(Adv^{PUF}_{\mathring{\mathcal{A}}}\right) \tag{4}$$

The session keys are computed independently and at random. Therefore, the likelihood of correctly guessing r_b is the same as the likelihood of guessing the SK. We obtain:

$$\Pr[\text{Sucss}_3] = \frac{1}{2} \tag{5}$$

Combining the above formulas, we arrive at the following conclusion:

$$\begin{aligned}\frac{1}{2}Advg^P_{\mathring{\mathcal{A}}} &= \left|\Pr[\text{Sucss}_0] - \frac{1}{2}\right| \\ &\leq \frac{q^2_{HSH}}{2^{l_{HSH}+1}} + \frac{(q_{SND} + q_{EXE})^2}{2n} + \frac{q_{SE}}{2} \\ &\quad + q_{SND}Advg^{PUF}_{\mathring{\mathcal{A}}}\end{aligned} \tag{6}$$

The final result is derived as follows:

$$Advg^P_{\mathring{\mathcal{A}}} \leq \frac{q^2_{HSH}}{2^{l_{HSH}}} + \frac{(q_{SND} + q_{EXE})^2}{n} + 2q_{SND}Advg^{PUF}_{\mathring{\mathcal{A}}}$$

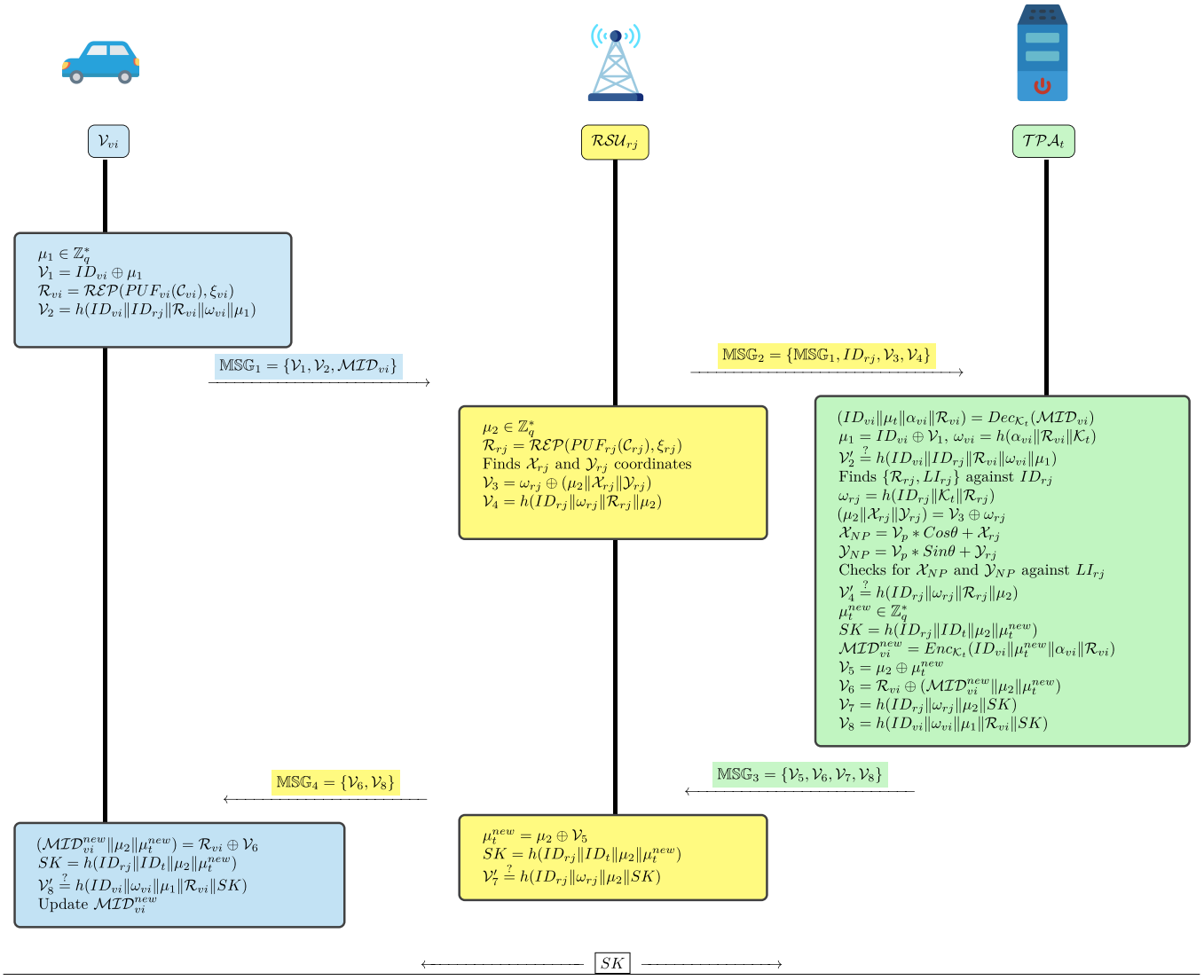

Fig. 2. PUF-enabled key-exchange protocol.

C. Formal Security Verification

In addition to conducting a formal security analysis of the proposed protocol within the Random Oracle Model (ROM), we also verified its security properties using the Scyther verification tool. The protocol was modeled using the Security Protocol Description Language (SPDL). After encoding the protocol specification in SPDL, the Scyther tool assessed it based on a set of predefined security claims, as outlined in [34]. For example, it is assumed that secret information remains protected from adversaries throughout the execution of protocols. As a result, Scyther does not identify any vulnerabilities that rely on such assumptions.

Each participant in the protocol namely $\mathcal{V}_{vi}$, $\mathcal{RSU}_j$, and $\mathcal{TPA}_t$ represented as a distinct role within the model. A role captures all relevant steps that an entity performs as part of the authentication process. We specified auxiliary functions associated with each role and defined protocol specific elements such as timestamp and nonce using the *timestamp* and *fresh* declarations, respectively.

To streamline the protocol definition, we employed macros and used *send* and *recieve* commands to represent the transmission and reception of messages. Additionally, we incorporated claim statements within each role to capture the intended security guarantees. Scyther supports various claims, including *Alive*, *Niagree*, etc. These claims were automatically verified by the tool, which then produced the corresponding

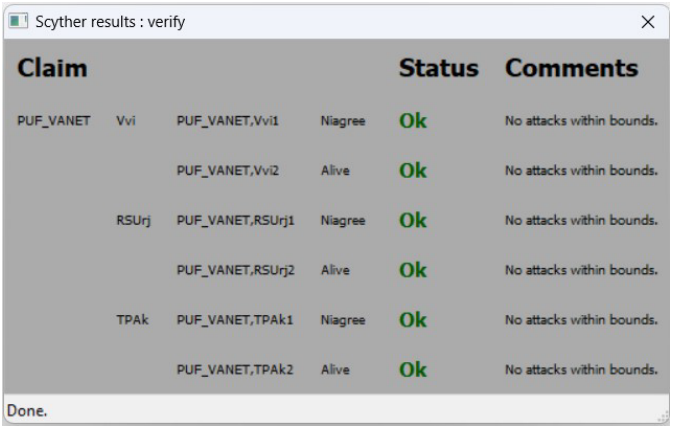

Fig. 3. Results of Scyther simulation.

TABLE II
NOTATION

Notations	Explanation
$\mathcal{TPA}_t$	Trusted-Party Agent
$\mathcal{K}_t$	Master key of $\mathcal{TPA}_t$
$\mathcal{V}_{vi}$	ith Vehicle
$ID_{vi}, \mathcal{MID}_{vi}$	Real and Masked identities of $\mathcal{V}_{vi}$
$\langle \mathcal{C}_{vi}, \mathcal{R}_{vi} \rangle$	Challenge-response pair of $\mathcal{V}_{vi}$
$\mathcal{OBU}_{vi}$	On-board unit of $\mathcal{V}_{vi}$
$\mathcal{RSU}_{rj}$	jth roadside Unit
ID_{rj}	Real Identity of $\mathcal{RSU}_{rj}$
$\langle \mathcal{C}_{rj}, \mathcal{R}_{rj} \rangle$	Challenge-response pair of $\mathcal{RSU}_{rj}$
LI_{rj}	Location identifier of $\mathcal{RSU}_{rj}$
N	Target point calculated using V_p
$\mathcal{X}_{NP}$	X-coordinate of point N
$\mathcal{Y}_{NP}$	Y-coordinate of point N
$\cos\theta$	Cosine of the angle θ
$\sin\theta$	Sine of the angle θ
$\mathring{\mathcal{A}}$	Adversary
μ_1, μ_2	Arbitrary Value
$Enc_{\mathbb{K}}, Dec_{\mathbb{K}}$	Encryption/ Decryption algorithm
Z_p^*	Positive set of integers
$\stackrel{?}{=}$	Either equal to or not
$\oplus$	Bitwise XOR Operator
$\|$	Concatenation operator

analysis results. As illustrated in Figure 3, the verification outcomes affirm the security robustness of the proposed protocol.

V. PERFORMANCE EVALUATION

This section evaluates the performance of our PUF-enabled key exchange protocol within VANETs. We compare our protocol against related protocols of [11], [12], [13], [14], [35], and [36]. To ensure a fair comparison, we have selected benchmark protocols that are specifically designed for the VANET environment. These protocols address key security requirements in VANETs, such as authentication and key agreement. Furthermore, we have prioritized recent protocols to ensure that our analysis captures the most up-to-date advancements in VANET security. By including these benchmark protocols, we aim to evaluate our proposed protocol against current solutions and emphasize its strengths in terms of resilience, scalability, and efficiency. This comparative analysis provides a comprehensive assessment of the protocol's performance within the dynamic VANET environment. The performance evaluation focuses on different metric including computation, energy, and communication costs. Furthermore, we evaluate the security features of the proposed protocol in comparison with related protocols to demonstrate its enhanced security capabilities. The details are discussed in subsequent subsections:

A. Experimental Configuration and Simulation Setup

An experimental configuration and simulation setup have been created to measure the execution time of cryptographic operations in the proposed and related protocols. To replicate a real-world VANET environment, we utilize a Raspberry Pi, simulating the cryptographic computations performed by $\mathcal{V}_{vi}$ and $\mathcal{RSU}_{rj}$. On the other hand, we utilized a dedicated desktop system to implement the operations of $\mathcal{TPA}_t$ side. The specifications of Raspberry Pi and dedicated desktop system are detailed in Table III, outlining the simulation environment's configuration. The empirical results and running times for key cryptographic operations are obtained through extensive implementation and testing. The key cryptographic operations from the proposed and related protocols have been implemented multiple times in accordance with their implementation environment and specifications. While implementing, we emphasized the more demanding cryptographic operations, deliberately omitting operations such as XOR and concatenation due to their minimal costs. The running time of various operations is listed in Table IV.

TABLE III
HARDWARE SPECIFICATIONS FOR IMPLEMENTATION

Features	Raspberry pi	System
Model	Raspberry pi 3 B+	Lenovo ThinkBook 16 Gen 6
Operating System	Debian based GNU/Linux	Windows 11 Home 64
CPU	BCM2837B0	Core i5-13420H
Speed	1.4 GHz	4.60 GHz
Memory	1GB LPDDR2 SDRAM	8GB DDR5-5200MHz
Language	Python	Python
Library	PyNaCl	PyNaCl

TABLE IV
RUNNING TIME OF OPERATIONS

Operation	Symbol	Running Time (ms)	
		Raspberry pi	System
Point multiplication	T_{PM}	2.810	0.935
Encryption/Decryption	$T_{ENC/DEC}$	2.356	0.815
Hash function	$T_{h(.)}$	1.990	0.660
SRAM PUF	T_{PUF}	0.780	0.112

B. Computation Overhead

This subsection presents the computation overhead comparison between the proposed PUF-enabled key exchange and related protocols [11], [12], [13], [14], [35], [36]. We determine the computation overhead using the running time of various operations outlined in Table IV. Since the registration phase is a one-time process, we have only considered the operations that are used in the authentication process to determine the computation overheads of the proposed and related protocols.

In proposed protocol, $\mathcal{V}_{vi}$ uses $(3T_{h(.)} + 1T_{PUF})$, and $\mathcal{RSU}_{rj}$ also uses $(3T_{h(.)} + 1T_{PUF})$, which collectively become $(6T_{h(.)} + 2T_{PUF})$. As mentioned in Section V-A, the cryptographic operations of $\mathcal{V}_{vi}$ and $\mathcal{RSU}_{rj}$ sides are implemented on Raspberry pi. Therefore, while considering the running time of cryptographic operations implemented on Raspberry pi from Table IV, the computation overhead of $\mathcal{V}_{vi}$ and $\mathcal{RSU}_{rj}$ sides is $6T_{h(.)} + 2T_{PUF} \approx 13.5\ ms$. Likewise, $\mathcal{TPA}_t$ executes $(7T_{h(.)} + 2T_{ENC/DEC})$ to complete the authentication process. The operations at $\mathcal{TPA}_t$ side are performed on a system. Thus, considering the running time of operations implemented on a system from Table IV, the computation overhead of $\mathcal{TPA}_t$ side is $7T_{h(.)} + 2T_{ENC/DEC} \approx 6.25\ ms$. The accumulative computation overhead of our protocol is $13.5 + 6.25 \approx 19.75\ ms$.

The computation overheads of related protocols [11], [12], [13], [14], [35], [36] are assessed using the same method. Table V and Figure 4(a) present the computation overhead comparison between the proposed and related protocols. Comparatively, our protocol takes 42.95%, 73.62%, 20.81%,

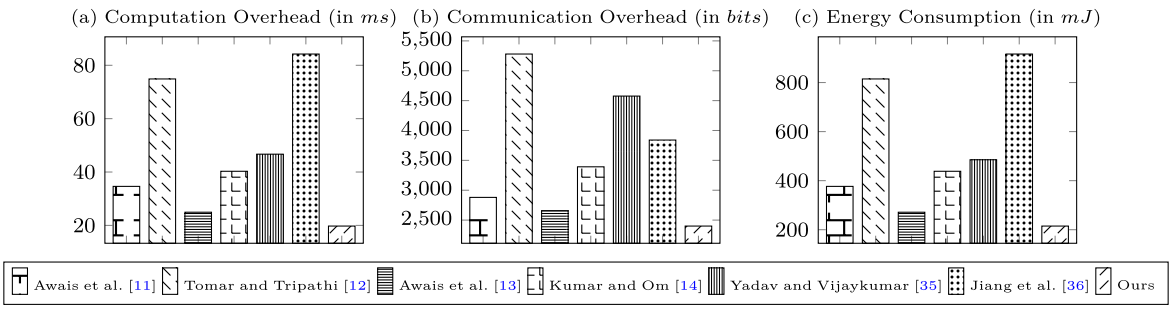

Fig. 4. Visual Representation: Computation overhead, communication overhead and energy consumption.

TABLE V
A DETAILED COMPARISON: COMPUTATION OVERHEAD, ENERGY CONSUMPTION AND COMMUNICATION OVERHEAD

Protocols	Computation overhead (ms)			Communication Overhead ($bits$)	Energy Consumption (mJ)
	$\mathcal{V}_{vi}$ & $\mathcal{RSU}_{rj}$	$\mathcal{TPA}_t$	Total		
[11]	$5T_{h(.)} + 5T_{PM} \approx 24.00$	$9T_{h(.)} + 5T_{PM} \approx 10.62$	34.62	2880	376.67
[12]	$13T_{h(.)} + 12T_{PM} \approx 59.59$	$9T_{h(.)} + 10T_{PM} \approx 15.29$	74.88	5280	814.69
[13]	$9T_{h(.)} + 1T_{PUF} \approx 18.69$	$7T_{h(.)} + 2T_{ENC/DEC} \approx 6.25$	24.94	2656	271.34
[14]	$11T_{h(.)} + 4T_{PM} \approx 33.13$	$8T_{h(.)} + 2T_{PM} \approx 7.15$	40.28	3392	438.25
[35]	$9T_{h(.)} + 7T_{PM} \approx 37.58$	$5T_{h(.)} + 6T_{PM} \approx 8.91$	46.69	4576	485.25
[36]	$32T_{h(.)} + 3T_{PM} + 1T_{PUF} + 1T_{ENC/DEC} \approx 75.25$	$12T_{h(.)} + 1T_{PM} + 1T_{PUF} \approx 8.97$	84.22	3840	916.31
Ours	$6T_{h(.)} + 2T_{PUF} \approx 13.5$	$7T_{h(.)} + 2T_{ENC/DEC} \approx 6.25$	19.75	2400	214.88

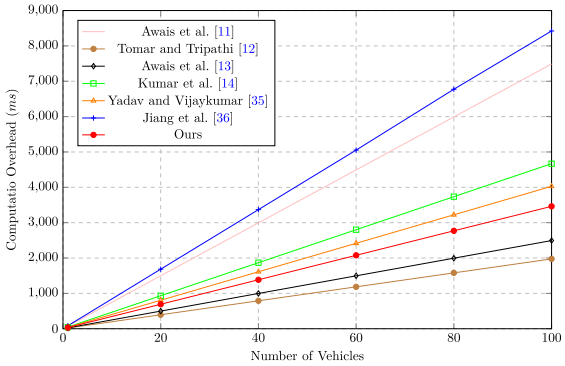

Fig. 5. Comparative analysis on scalability.

50.97%, 57.70%, and 76.55% lower computation overhead as compared to [11], [12], [13], [14], [35], and [36], respectively. This comparison reveals that our protocol achieves a 53.77% average reduction in computation overhead compared to related ones.

To evaluate the scalability of our protocol, we present an analysis in Figure 5, which reveals how computation overhead varies with an increasing number of vehicles in the VANET environment. The results show that, while the computation overhead of related protocols grows significantly with network size, our protocol exhibits a comparatively slower increase. This efficient growth trend demonstrates that the protocol scales effectively and remains suitable for large-scale vehicular networks.

C. Communication Overhead

The following assumptions have been taken to determine the communication overhead of the proposed PUF-enabled key exchange protocol: Identity, concatenation, XoR, timestamp, and random numbers take 160 *bits*, respectively. AES encryption/decryption, hash function, and ECC point multiplication take 128, 256, and 320 *bits*, respectively.

In the proposed protocol, the entities exchange four messages to complete the authentication process. 1: $\mathcal{V}_{vi}$ initiates a message $\mathbb{MSG}_1 = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{MID}_{vi}\}$ towards $\mathcal{RSU}_{rj}$, which takes $(160 + 160 + 256) = 576$ *bits*. 2: $\mathcal{RSU}_{rj}$ transmits two messages $\mathbb{MSG}_2 = \{\mathbb{MSG}_1, ID_{rj}, \mathcal{V}_3, \mathcal{V}_4\}$ and $\mathbb{MSG}_4 = \{\mathcal{V}_6, \mathcal{V}_8\}$ towards $\mathcal{TPA}_t$ and $\mathcal{V}_{vi}$, respectively. The messages $\mathbb{MSG}_2$ and $\mathbb{MSG}_4$ take $(160+160+256)+(160+256) = 992$ *bits* for transmission. 3: $\mathcal{TPA}_t$ sends a message $\mathbb{MSG}_3 = \{\mathcal{V}_5, \mathcal{V}_6, \mathcal{V}_7, \mathcal{V}_8\}$ towards $\mathcal{RSU}_{rj}$, which requires $(160 + 160 + 256 + 256) = 832$ *bits*. Therefore, the aggregated communication overhead is $(576 + 992 + 832) = 2400$ *bits*.

We determine the communication overheads of related protocols [11], [12], [13], [14], [35], [36] in similar way. The detailed communication overheads comparison between the proposed and related protocols is shown in Table V and Figure 4(b). In comparison to related protocols [11], [12], [13], [14], [35], [36], our protocol takes 16.67%, 54.55%9.64%, 29.25%, 47.55%, and 37.5% lower communication overhead, respectively. Therefore, our protocol achieves a 35.53% average reduction in communication overhead compared to related ones.

D. Energy Consumption

During the implementation, the system expends battery power to compute and transmit data among the participants. This consumed energy is quantitatively expressed as $CE = T_e \times CP$ within the wireless communication channel. Here, T_e represents the required execution/computation time (as shown in Table V), and CP denotes CPU power, set at 10.88 W for data transmission over the wireless channel. The consumed energy is directly proportional to the execution/computation time. Consequently, if the data transmission protocol requires lower computational time, it will accordingly consume less battery power. As indicated in Table V, the

TABLE VI
SECURITY ATTRIBUTES COMPARISON

Protocols → Attributes ↓	[11]	[12]	[13]	[14]	[35]	[36]	Ours
$A1$	✓	✓	✓	✓	✓	✓	✓
$A2$	✓	✓	✓	✓	✓	✓	✓
$A3$	✓	✓	✓	✓	✓	✓	✓
$A4$	✓	✓	✓	✓	✓	✓	✓
$A5$	⊗	⊗	✓	⊗	✓	✓	✓
$A6$	⊗	⊗	✓	⊗	⊗	✓	✓
$A7$	✓	⊗	⊗	✓	⊗	⊗	✓
$A8$	⊗	⊗	⊗	⊗	⊗	⊗	✓
$A9$	✓	✓	✓	✓	⊗	⊗	✓
$A1$: Offers $\mathcal{V}_{vi}$ Anonymity and Untraceability, $A2$: Resists $\mathcal{V}_{vi}$ Impersonation							
$A3$: Resists $\mathcal{RSU}_{rj}$ Impersonation, $A4$: Resists $\mathcal{TPA}_t$ Impersonation							
$A5$: Resists Physical Attack, $A6$: Resists Desynchronization Attacks							
$A7$: Resists ESL Attacks, $A8$: Resists Stolen Verifier Attack							
$A9$: Ensures Forward Secrecy ✓: Offers; ⊗: Does not offer							

proposed PUF-enabled key exchange protocol requires only 19.75 *ms* for execution. Therefore, it takes 214.88 *mJ* as energy Consumption and is proved to be more energy-efficient during message transmission compared to related protocols. The energy Consumption comparison among the proposed and related protocols is shown in Table V and Figure 4(c).

E. Security Attributes

Table VI summaries the comparison of the proposed PUF-enabled key exchange and related protocols of [11], [12], [13], [14], [35], and [36]. It is obvious from Table VI that the related protocols do not provide enough security as most of them are vulnerable to various attacks, including physical or cloning, desynchronization, stolen verifier and ESL. However, the comparison reveals that our protocol renders added security attributes compared to related protocols and achieves higher security.

The discussions of Section V highlight the effectiveness of our proposed PUF-enabled key exchange protocol for VANETs. Through meticulous testing and comparison with related protocols, our protocol enhances security features and optimizes computation, energy, and communication overheads. These results underscore the protocol's potential for practical deployment in vehicular networks where both security and efficiency are paramount.

VI. CONCLUSION

The VANET framework faces security challenges like data interception, manipulation, and unauthorized access due to public communication channels. Data transmission within VANETs is sensitive and must remain secure and available to authorized users only. Therefore, we proposed a PUF-enabled key exchange protocol in this article that effectively counters potential security attacks, emphasizing mutual authentication. We conduct both informal and formal security analyses of our protocol. The informal analysis demonstrates that our protocol resists various security threats, including vehicle, RSU, or TPA impersonation and desynchronization attacks, and also ensures vehicle anonymity and untraceability. The formal analysis confirms that our protocol withstands potential attacks during communication. Additionally, performance comparisons reveal that our protocol outperforms existing protocols, reducing computation overheads by 51.05% and communication overheads by 28.80%. While the proposed protocol ensures strong security and lightweight performance for VANETs, future work will focus on integrating context-aware mechanisms to improve flexibility across diverse application scenarios and also enhancing resilience against machine learning-based and side-channel attacks. We also plan to explore post-quantum cryptographic techniques to further strengthen the protocol's robustness.

REFERENCES

[1] V. Hassija, V. Chamola, S. Garg, D. N. G. Krishna, G. Kaddoum, and D. N. K. Jayakody, "A blockchain-based framework for lightweight data sharing and energy trading in V2G network," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5799–5812, Jun. 2020.

[2] S. A. Chaudhry, "Designing an efficient and secure message exchange protocol for Internet of Vehicles," *Secur. Commun. Netw.*, vol. 2021, pp. 1–9, May 2021.

[3] G. Singh Rawat, K. Singh, M. Shariq, A. K. Das, S. Ashraf Chaudhry, and P. Lorenz, "BTC2PA: A blockchain-assisted trust computation with conditional privacy- preserving authentication for connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 1, pp. 1134–1148, Jan. 2025.

[4] S. S. Moni and D. Manivannan, "A scalable and distributed architecture for secure and privacy-preserving authentication and message dissemination in VANETs," *Internet Things*, vol. 13, Mar. 2021, Art. no. 100350.

[5] T. Alladi, V. Chamola, and Naren, "HARCI: A two-way authentication protocol for three entity healthcare IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 361–369, Feb. 2021.

[6] G. Bansal, N. Naren, and V. Chamola, "RAMA: Real-time automobile mutual authentication protocol using PUF," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2020, pp. 265–270.

[7] S. A. Chaudhry et al., "A lightweight authentication scheme for 6G-IoT enabled maritime transport system," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2401–2410, Feb. 2023.

[8] M. N. Aman, U. Javaid, and B. Sikdar, "A privacy-preserving and scalable authentication protocol for the Internet of Vehicles," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1123–1139, Jan. 2021.

[9] S. M. Karim, A. Habbal, S. A. Chaudhry, and A. Irshad, "BSDCE-IoV: Blockchain-based secure data collection and exchange scheme for IoV in 5G environment," *IEEE Access*, vol. 11, pp. 36158–36175, 2023.

[10] S. Ashraf Chaudhry, A. Irshad, B. A. Alzahrani, A. Alhindi, M. Shariq, and A. Kumar Das, "TS-PAID: A two-stage PUF-based lightweight authentication protocol for Internet of Drones," *IEEE Access*, vol. 13, pp. 1458–1469, 2025.

[11] S. M. Awais et al., "Provably secure and lightweight authentication and key agreement protocol for fog-based vehicular ad-hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 12, pp. 21107–21116, Dec. 2024.

[12] A. Tomar and S. Tripathi, "A Chebyshev polynomial-based authentication scheme using blockchain technology for fog-based vehicular network," *IEEE Trans. Mobile Comput.*, vol. 23, no. 10, pp. 9075–9089, Oct. 2024.

[13] S. M. Awais et al., "PUF-based privacy-preserving simultaneous authentication among multiple vehicles in VANET," *IEEE Trans. Veh. Technol.*, vol. 73, no. 5, pp. 6727–6739, May 2024.

[14] P. Kumar and H. Om, "An anonymous and authenticated V2I communication with a simplified user revocation and re-registration strategy," *J. Supercomput.*, vol. 79, no. 7, pp. 8070–8096, May 2023.

[15] V. Kumar, "RSFVC: Robust biometric-based secure framework for vehicular cloud networking," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 5, pp. 3364–3374, May 2024.

[16] Q. Xie, Z. Ding, and P. Zheng, "Provably secure and anonymous V2I and V2V authentication protocol for VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7318–7327, Jul. 2023.

[17] Z. Wang et al., "An anonymous and revocable authentication protocol for vehicle-to-vehicle communications," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 5114–5127, Mar. 2023.

[18] V. Kumar, R. Kumar, V. Kumar, A. Kumari, and S. Kumari, "RAVCC: Robust authentication protocol for RFID based vehicular cloud computing," *J. Netw. Intell.*, vol. 7, no. 3, pp. 526–543, 2022.

[19] M. Safkhani, C. Camara, P. Peris-Lopez, and N. Bagheri, "RSEAP2: An enhanced version of RSEAP, an RFID based authentication protocol for vehicular cloud computing," *Veh. Commun.*, vol. 28, Apr. 2021, Art. no. 100311.

[20] T. Nandy et al., "An enhanced lightweight and secured authentication protocol for vehicular ad-hoc network," *Comput. Commun.*, vol. 177, pp. 57–76, Sep. 2021.

[21] T.-Y. Wu, Z. Lee, L. Yang, and C.-M. Chen, "A provably secure authentication and key exchange protocol in vehicular ad hoc networks," *Secur. Commun. Netw.*, vol. 2021, pp. 1–17, Jun. 2021.

[22] M. A. Saleem, X. Li, K. Mahmood, T. Tariq, M. J. F. Alenazi, and A. K. Das, "Secure RFID-assisted authentication protocol for vehicular cloud computing environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 12528–12537, Sep. 2024.

[23] T. Alladi, V. Chamola, B. Sikdar, and K.-K.-R. Choo, "Consumer IoT: Security vulnerability case studies and solutions," *IEEE Consum. Electron. Mag.*, vol. 9, no. 2, pp. 17–25, Mar. 2020.

[24] K. Emara, W. Woerndl, and J. Schlichter, "CAPS: Context-aware privacy scheme for VANET safety applications," in *Proc. 8th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jun. 2015, pp. 1–12.

[25] N. Ghosh, S. Chandra, V. Sachidananda, and Y. Elovici, "SoftAuthZ: A context-aware, behavior-based authorization framework for home IoT," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10773–10785, Dec. 2019.

[26] I. Bhattarai, C. Pu, K.-K. Raymond Choo, and D. Korać, "A lightweight and anonymous application-aware authentication and key agreement protocol for the Internet of Drones," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19790–19803, Jun. 2024.

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

[28] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2001, pp. 453–474.

[29] L. Lu and T. T.-H. Kim, "A high reliable SRAM-based PUF with enhanced challenge-response space," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 2, pp. 589–593, Feb. 2022.

[30] S. Ahmed et al., "Signcryption based authenticated and key exchange protocol for EI-based V2G environment," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5290–5298, Nov. 2021.

[31] K. Mahmood, M. N. Fatima, S. Shamshad, Z. Ghaffar, A. K. Das, and M. J. F. Alenazi, "A cost-effective key agreement encryption protocol for securing IIoT-enabled WSN communication," *IEEE Internet Things J.*, vol. 12, no. 5, pp. 5185–5193, Mar. 2025.

[32] N. Koblitz and A. J. Menezes, "The random Oracle model: A twenty-year retrospective," *Designs, Codes Cryptography*, vol. 77, nos. 2–3, pp. 587–610, Dec. 2015.

[33] M. Bellare and P. Rogaway, "Random Oracles are practical: A paradigm for designing efficient protocols," in *Proc. 1st ACM Conf. Comput. Commun. Security*, 1993, pp. 62–73.

[34] M. Eldefrawy, I. Butun, N. Pereira, and M. Gidlund, "Formal security analysis of LoRaWAN," *Comput. Netw.*, vol. 148, pp. 328–339, Jan. 2019.

[35] K. A. Yadav and P. Vijayakumar, "LPPSA: An efficient lightweight privacy-preserving signature-based authentication protocol for a vehicular ad hoc network," *Ann. Telecommun.*, vol. 77, nos. 7–8, pp. 473–489, Aug. 2022.

[36] Q. Jiang, X. Zhang, N. Zhang, Y. Tian, X. Ma, and J. Ma, "Three-factor authentication protocol using physical unclonable function for IoV," *Comput. Commun.*, vol. 173, pp. 45–55, May 2021.