future internet

MDPI

Article

Privacy-Preserving Authentication Based on PUF for VANETs

Lihui Li [1,2], Hanwen Deng [1], Zhongyi Zhai [1] and Sheng-Lung Peng [3,*]

[1] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China; llhljz@mails.guet.edu.cn (L.L.); coderkled@163.com (H.D.); zhaizhongyi@guet.edu.cn (Z.Z.)
[2] Guangxi Universities Key Laboratory of Application Technology of Intelligent Connected Vehicle, Guangxi Vocational Normal University, Nanning 530007, China
[3] Department of Creative Technologies and Product Design, National Taipei University of Business, Taoyuan City 324022, Taiwan
* Correspondence: slpeng@ntub.edu.tw

Abstract: The secret key is stored in an ideal tamper-proof device so that a vehicle can implement a secure authentication with the road-side units (RSUs) and other drivers. However, some adversaries can capture the secret key by physical attacks. To resist physical attacks, we propose a physical-preserving authentication based on a physical unclonable function for vehicular ad hoc networks. In the proposed scheme, a physical unclonable function is deployed on the vehicle and the RSU to provide a challenge–response mechanism. A secret key is only generated by the challenge–response mechanism when it is needed, which eliminates the need to store a long-term secret key. As a result, this prevents secret keys from being captured by adversaries, improving system security. In addition, route planning is introduced into the proposed scheme so that a vehicle can obtain the authentication key of RSUs on its route before vehicle-to-infrastructure authentication, which greatly speeds up the authentication when the vehicle enters the RSUs' coverage. Furthermore, a detailed analysis demonstrates that the proposed scheme achieves security objectives in vehicular ad hoc networks. Ultimately, when contrasted with similar schemes, the performance assessment demonstrates that our proposed scheme surpasses others in terms of computational overhead, communication overhead and packet loss rate.

Keywords: authentication; physical-preserving; physical unclonable function; route planning

Citation: Li, L.; Deng, H.; Zhai, Z.; Peng, S.-L. Privacy-Preserving Authentication Based on PUF for VANETs. *Future Internet* **2024**, *16*, 326. https://doi.org/10.3390/fi16090326

Academic Editor: Gianluigi Ferrari

Received: 10 May 2024
Revised: 23 August 2024
Accepted: 30 August 2024
Published: 8 September 2024

1. Introduction

With the escalating number of vehicles, vehicular ad hoc networks (VANETs) are poised to enhance the quality of travel and traffic conditions [1]. Typically, VANETs are comprised of three primary components: certification authority (CA), vehicles equipped with on-board units (OBUs), and roadside units (RSUs). These networks primarily utilize two principal communication modes: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), both communication modes adhere to the dedicated short-range communication protocol for wireless access in the vehicular environment [2,3].

However, due to the inherent openness of wireless channels within VANETs, adversaries can readily execute a range of attacks, including a denial of service attack (DoS), a Sybil attack, and so on. Hence, in wireless channel communication scenarios, security and privacy-preserving are critical challenges [4,5]. In particular, traditional security standards such as confidentiality, authentication, and integrity serve to guarantee that transmitted messages are accessible solely to authorized entities, thereby upholding the integrity and security of the communication process [6,7]. However, most traditional solutions are based on the assumption that the secret key of a vehicle is securely stored in an ideal tamper-proof device (TPD). However, as physical attacks, such as side-channel attacks, become more powerful, an adversary can retrieve the secret key from TPD.

In recent years, the physical unclonable function (PUF) has been regarded as a promising tool for protecting against physical attacks [8]. The PUF is easy to evaluate but hard to predict, and it is impossible to replicate. Due to random uncontrollable variables in the manufacturing process, the PUF can generate a challenge–response pair (CRP) that is also unique. Therefore, the PUF has significantly higher physical security by generating unique CRP rather than storing secret keys in memories [9]. In addition, the CRPs need to be updated.

In recent developments, route planning has emerged as a prevalent practice within the self-driving vehicle landscape [10]. During the route planning phase, a vehicle strategically chooses its preferred route, subsequently communicating this decision to the RSUs along its trajectory with the assistance of the CA. Upon entering the coverage area of the designated RSUs, swift and highly effective authentication is facilitated between the vehicle and RSUs through the exchange of shared messages [11].

Hence, in order to prevent the secret key from physically being stolen by adversaries and speed up authentication, we proposed a physical-preserving authentication based on PUF for VANETs in this paper, and we summarized the main contributions as follows.

- The proposed scheme provides physical security through PUF. In particular, the secret key of the vehicle is generated by a challenge–response mechanism based on PUF, instead of storing the secret key in TPD's physical memory. Therefore, adversaries cannot obtain the secret key. And fuzzy extractor (FE) technology is introduced to enhance the stability of PUF and mitigate various electrical noise interferences. Furthermore, aside from the unlikability of pseudonyms, we also propose an update mechanism for CRPs to ensure that adversaries cannot attack user privacy by analyzing CRPs;
- In the proposed scheme, a vehicle plans its route in advance, knowing which RSUs it will pass by. Then, Before V2I authentication, the vehicle requests the CA for the secret authentication keys of all RSUs on its path at once. The request process introduces oblivious transfer (OT) technology to avoid leaking the vehicle's driving trajectory. Next, the V2I authentication will be sped up when the vehicle enters the RSUs' coverage. As a result, this improves the authentication efficiency when the vehicle is roaming among different RSUs' coverage;
- A comprehensive analysis demonstrates that the proposed scheme effectively fulfills security objectives within VANETs. Furthermore, through comparative analysis with related schemes, our evaluation indicates superior performance in terms of time consumption and communication overhead.

The subsequent sections of this paper are structured as follows: Section 2 outlines the related works, while Section 3 provides the preliminaries. The proposed scheme is elaborated upon in Section 4, followed by security analysis and performance analysis in Sections 5 and 6. Ultimately, we conclude the paper in Section 7 and point out future research directions in Section 8, respectively.

2. Related Works

VANETs have become a prevalent research field in the intelligent transportation system in order to avoid traffic congestion and accidents and enhance the driving experience. Several methods have been proposed for the security and privacy issues of VANETs.

Zhu et al. [12] introduced a privacy-preserving authentication and data aggregation framework for fog-based smart grid systems. Their study outlines the architecture of a fog-based smart grid, addressing its applications, security, and privacy challenges. Additionally, they propose a privacy-preserving authentication and data aggregation scheme tailored for fog-based smart grids. This scheme leverages short randomizable and blind signatures to offer anonymous authentication under specific conditions. Furthermore, the integration of fog nodes addresses billing issues subsequent to anonymous authentication.

To preserve privacy, Zhang et al. [13] introduced an authentication framework that integrates fifth-generation communication technology (5G) with edge computing, diverging from the architectural conventions of previous 802.11p-based inter-vehicle communication

networks. Within this proposed framework, device-to-device technology serves as the conduit for communication between vehicles, deviating from the traditional model for VANETs. However, achieving secure communication within a 5G-enabled model poses a formidable challenge. To address this challenge, the proposed framework adopts a two-step approach to security authentication. Initially, authentication and the selection of an edge computing vehicle are imperative, leveraging a fuzzy logic mathematical method during the selection process. Subsequently, mutual authentication between edge computing and ordinary vehicles is executed. The procedural sequence facilitates the exchange of security information among vehicles within a group while simultaneously safeguarding the identity privacy and traceability of each vehicle [7].

To circumvent the redundant authentication of identical messages and identify invalid messages within a batch, Cui et al. [14] innovatively integrated an edge-computing concept into the message-authentication process of VANETs. Within their framework, an RSU can adeptly authenticate messages from neighboring vehicles and subsequently disseminate the results to vehicles within its communication range. This approach effectively mitigates redundant authentication procedures and enhances the overall system efficiency.

Hathal et al. [15] put forward a certificateless and lightweight authentication scheme aimed at furnishing secure communication avenues for vehicle communication systems. In their study, they introduce authentication tokens as substitutes for digital certificates, thereby alleviating the administrative load associated with certificate management for a trusted authority (TA). Furthermore, the adoption of tokens guarantees the attainment of mutual authentication for V2I communication.

Cui et al. [16] introduced a lightweight message authentication framework based on a reputation system tailored for 5G-enabled VANETs. Within this authentication framework, the TA assumes responsibility for reputation management. Notably, if a vehicle's reputation score falls below the specified threshold, it becomes ineligible to receive a credit reference from the TA, thus mitigating the influx of untrusted messages within VANETs at the source.

However, none of the above schemes take into account the possibility that the vehicle could be physically attacked, leading to the disclosure of the secret key. In fact, with side-channel attacks, such as power, electromagnetic, and time usage attacks on vehicles, the attacker could still access the secret key stored in TPD. To combat this issue, a physical-preserving authentication based on PUF for VANETs is proposed by us. In the proposed scheme, we use PUF to generate a secret key only when it is needed for V2I authentication without ever storing the secret key in TPD's permanent storage, which ensures the physical security of the system. In addition, when the drivers enter the coverage of RSUs, route planning is also introduced into the proposed scheme to speed up the V2I authentication.

3. Preliminaries

In this section, we introduce some basic knowledge, including the system model, the design goals of the authentication protocol for VANETs and the Physical Unclonable Function.

3.1. System Model

There are usually three entities in VANETs: Certification Authority (CA), roadside unity (RSU), and vehicle, as shown in Figure 1.

CA: CA can perform well in computing and storing. It is responsible for managing the entire VANETs and tracing and revoking the real identity of any misbehaving vehicle. Usually, vehicles and RSUs need to be registered in the CA. Then CA stores RSU's and vehicle's vital information such as RSU's secret key of authentication. Therefore, it can help the mutual authentication between vehicles and RSUs. Finally, CA is assumed to be fully trusted.

RSU: RSU is deployed at the roadsides and is embedded with a PUF chip. The communication method between the vehicle and RSU is via wireless channels, while a

stable wired channel is between RSU and CA. In particular, RSU generates its authentication key only when authenticating with the vehicle. RSU is honest but curious.

Vehicle: OBU and PUF chips are usually equipped on the vehicle. The OBU is responsible for facilitating communication with other OBUs, RSUs, and CA. Conversely, the primary role of the PUF chips is to generate CRPs. Ultimately, the vehicle is characterized as honest yet curious in its behavior.

Adversary: Anyone who intends to change, manipulate, hide the data, or gain physical access to vehicles and RSUs for secret keys is regarded as the active adversary. The adversary may inject new packets, store old messages, initiate a session, or pretend to be a valid device.

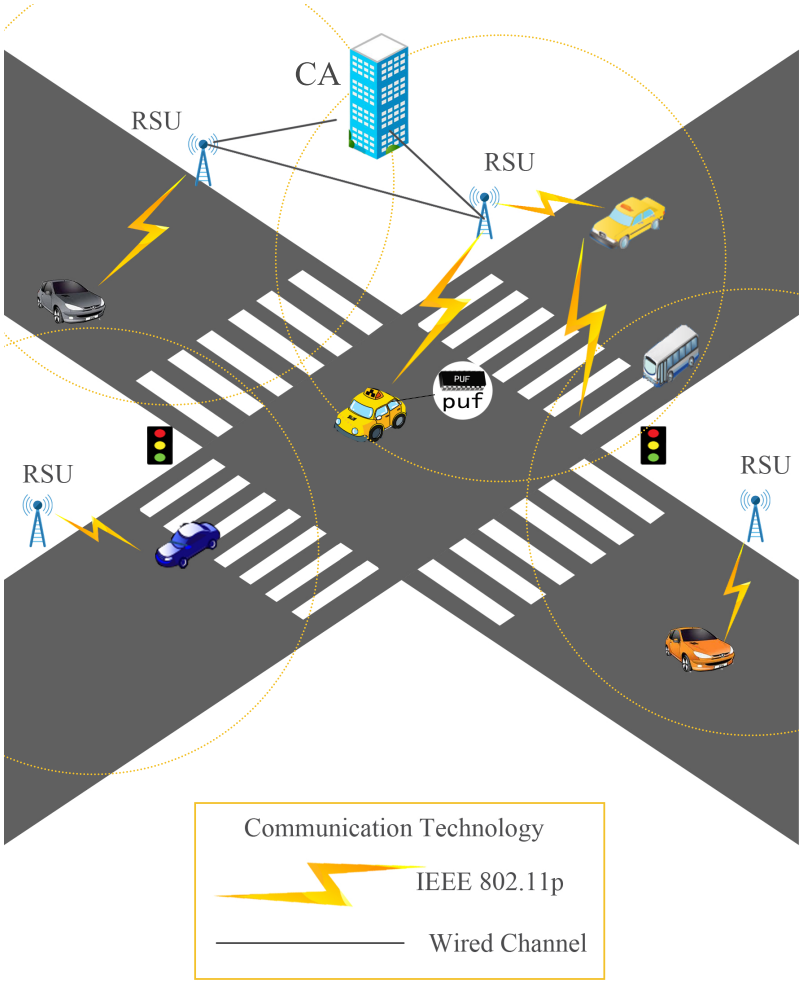

Figure 1. System model.

3.2. Threat Model

The proposed scheme's initial phase and registration phase occur within a secure channel. However, it is important to note that the security of the communication between vehicles and other entities, including RSUs and other vehicles, cannot be fully guaranteed. A crucial aspect to consider is that, despite adhering to strict protocols, RSUs often exhibit curiosity towards sensitive vehicle information, such as travel routes and speeds. Furthermore, all algorithms of this scheme are discussed in a CA domain, where each domain has a limited geographical coverage, typically corresponding to a single city. Consequently, it is assumed that all entities participating in this scheme maintain synchronous time. And the identities of all RSUs are publicly accessible. The adversary model assumes the following:

Firstly, it is presumed that an attacker possesses the capability to intercept, manipulate, delete, and replay any information transmitted over unsecured public channels. This encompasses all forms of communication that lack robust encryption or protection.

Secondly, the system's entities are vulnerable to physical attacks, making it likely that their secret parameters could be stolen. Additionally, apart from executing the protocol with integrity, RSUs might attempt to decipher the privacy of individual vehicles by analyzing legitimately received messages.

3.3. Design Goals

Authentication and Integrity: Upon receiving a message, both vehicles and RSUs must possess the capability to ascertain its validity. Should the message be falsified or altered during transmission by unauthorized parties, the receiver should be equipped to detect such tampering.

Physical protection: In order to ensure the security of the vehicles, the secret keys of the vehicles must not be physically stolen.

Anonymity and Traceability: There is nobody else but CA who can obtain the vehicles' real identity through the messages from the given vehicles.

3.4. Physical Unclonable Function

PUF offers a challenge–response mechanism that outputs a response for a challenge as an input. It can be represented as follows: $R = PUF(C)$. Exploiting the singularity of the integrated circuit's physical micro-structure in the manufacturing process, it ensures that each PUF is unique. Because the operation of a PUF relies on the intrinsic physical characteristics of the integrated circuit, any endeavor to tamper with the PUF disrupts its functionality, rendering it ineffective [17].

3.5. Fuzzy Extractor

Fuzzy extractor (FE) [18] is a cryptographic tool designed to convert imperfect, noisy data (such as fingerprints or PUFs) into secure keys. In cryptographic mechanisms, secret values are typically required to be evenly distributed and precisely regenerable when needed. However, in real-world applications, such as fingerprints or PUF values, these requirements are challenging to fulfill due to physiological and environmental factors that introduce variations in the detected data across different measurements. FE can address this issue by correcting certain differences in the input data. It allows for a certain level of noise in the input, and as long as the input is sufficiently similar, it can output an identical, uniformly random string. As depicted in Figure 2, the FE consists of two main parts:

Generator: $(P_F, R_F) \leftarrow Gen(W)$. This function takes a string W (a one-time sampling of a random noise source) as input and outputs two strings: R_F, a random string, and P_F, an exposed auxiliary string.

Regenerator: $R'_F \leftarrow Rep(W', P_F)$. The regeneration algorithm takes another sampling of the noise source, W', and the exposed auxiliary string P_F as inputs, and outputs a string R'_F.

Correctness: The correctness requirement for the FE is that if the distance between the two samples W and W' is close enough, then $R'_F = R_F$, ensuring that R_F can be accurately reproduced.

Security: The security requirement is that if the random source has sufficient entropy, then R_F will be uniformly random, providing a high level of security for the generated keys.

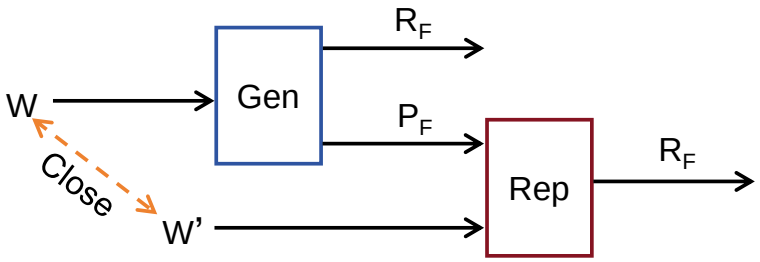

Figure 2. Fuzzy extractor.

3.6. Oblivious Transfer

Oblivious transfer (OT) is a classic cryptographic primitive that is widely utilized in multi-party secure computing and other related fields. There exist numerous different OT schemes, but they can generally be categorized as k-out-n OT_n^k schemes [11]. Typically, two entities are involved in an OT_n^k protocol: the sender, who possesses n messages, and the receiver, who wishes to obtain k messages from the sender. Specifically, the sender encrypts all n messages without knowing which k messages the receiver intends to obtain and then sends all the encrypted messages to the receiver. The receiver is then able to decrypt only the k messages it needs. For ease of understanding, the implementation principle of OT_n^1 is illustrated in Figure 3.

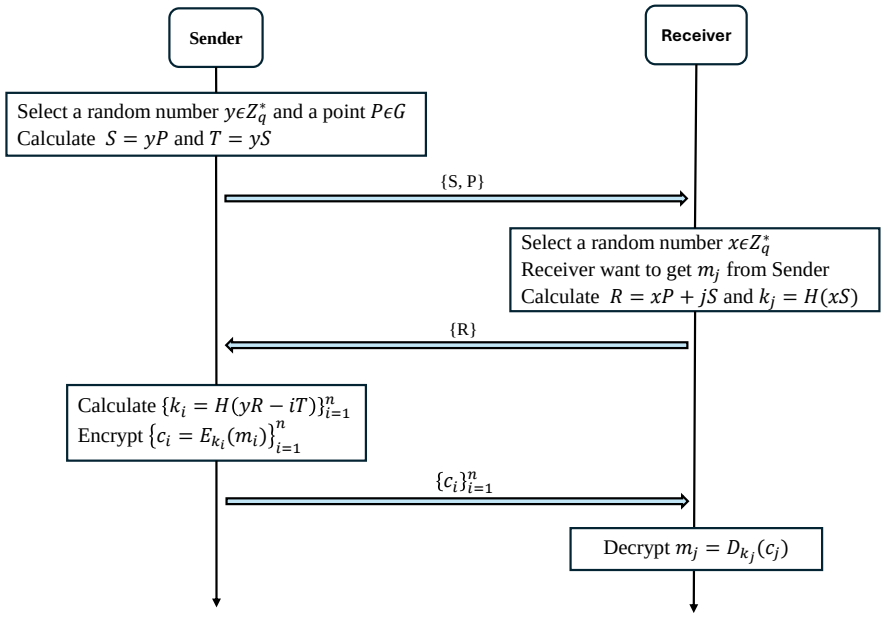

Figure 3. Oblivious transfer.

4. Proposed Scheme

In this section, we outline the proposed scheme, comprising seven distinct phases: the initial phase, RSU registration phase, vehicle registration phase, route planning phase, authentication phase, CRP update phase, and pseudonym update phase.

4.1. Initial

Let F_p be a finite field, and p is a large prime number to represent the size of the finite field. And E is an elliptic curve, CA chooses a group G from E where q is the order and G is its generator. Then, CA generates a public and secret key pair $PK_c = sk_cG$, where PK_c is a public key and sk_c is a secret key. CA generates a revocation list $REV = \{ID_v, AID_v, t\}$, where ID_v represents the real identity of a malicious vehicle, AID_v represents the pseudonym of the malicious vehicle, and t represents the timestamp. Once a

vehicle is detected as malicious, its real identity, pseudonym, and the current timestamp are added to the revocation list. Finally, CA selects the hash function $H(\cdot)$.

4.2. RSU Registration

CA generates an identity ID_u and a sequence number n_u for a RSU and sends them to RSU securely. Additionally, the RSU generates an initial CRP (C_u, R_u) by PUF chips employed on RSU, where $R_u = PUF(C_u)$. It is processed by the *Gen* function of FE, $(P_{Fu}, R_{Fu}) = Gen(R_u)$, Then, the RSU calculates its symmetric encryption key $K_u = H(R_{Fu})$ as the authentication key, and CRP (C_u, R_{Fu}) and K_u both need to be updated periodically. Next, the RSU sends the K_u to CA via a secure way. Last, $\{ID_u, K_u\}$ is stored by CA, and $\{ID_u, C_u, P_{Fu}\}$ is held by the RSU.

4.3. Vehicle Registration

CA generates an identity ID_v, selects a random number $r_v \in Z_q^*$ for a vehicle, and generates the pseudonym AID_v for the vehicle, where $AID_v = H(sk_c||r_v) \oplus ID_v$. Then, select a random number $y_v \in Z_q^*$, calculate $S_v = y_v P$ and $T_v = y_v S_v$, CA sends $\{AID_v, ID_v, S_v\}$ to the vehicle via a secure method. In addition, the vehicle generates an initial CRP (C_v, R_v) using PUF chips employed on the vehicle, where $R_v = PUF(C_v)$. Calculate $(P_{Fv}, R_{Fv}) = Gen(R_v)$. Then, the vehicle sends the R_{Fv} to CA via a secure method. Last, $\{ID_v, AID_v, r_v, y_v, T_v, R_{Fv}\}$ is stored in CA's database, $\{AID_v, C_v, P_{Fv}, S_v\}$ is held by the vehicle.

4.4. Route Planning

To improve the V2I authentication efficiency when a vehicle is roaming among different RSUs' coverage, the vehicle plans its route in advance before driving, knowing the RSU set that it will pass by, namely $\overrightarrow{PA_v} = \{RSU_{n_1}^1, RSU_{n_2}^2, \cdots, RSU_{n_a}^a, \cdots, RSU_{n_k}^k\}$; here, the superscript of each RSU represents its sequence number in the path, and the subscript represents its sequence number in the CA. Then, the vehicle sends a message to request for these RSUs' authentication key from CA. In order to prevent CA from tracking the vehicle's driving path, OT technology is used here. The interactive implementation process between the vehicle and CA is shown in Figure 4, and the details are as follows.

(1) The vehicle generates a timestamp t_1, calculates $Auth_v = H(Rep(PUF(C_v), P_{Fv}))$ (here, the *Rep* function of FE is used to eliminate the noise of PUF), and selects k random numbers $\overrightarrow{X_v} = \{x_a \in Z_q^*\}_{a=1}^k$; it is necessary to note that there is a one-to-one correspondence between the elements of $\overrightarrow{X_v}$ and $\overrightarrow{PA_v}$, respectively. Calculate symmetry session keys $\overrightarrow{SK_v} = \{sk_a = H(x_a S_v)\}_{a=1}^k$ and relevant transmitted auxiliary parameters $\overrightarrow{R_v} = \{R_a = n_a S_v + x_a P\}_{a=1}^k$ and send $M_1 : \{\phi_1 = H(Auth_v||\overrightarrow{R_v}||AID_v||t_1), \overrightarrow{R_v}, AID_v, t_1\}$ to CA via an RSU that has previously been mutually authenticated with the vehicle;

(2) On receiving M_1, CA verifies t_1. If it is valid, CA first detects whether AID_v is in the revocation list REV. If it is in the list, it indicates that the vehicle is a malicious vehicle and rejects the request of the vehicle. Then CA utilizes AID_v to retrieve R_{Fv}, which was generated during the vehicle's registration process, and verifies ϕ_1 by checking whether the equation $\phi_1 = H(H(R_{Fv})\|\overrightarrow{R_v}\|AID_v\|t_1)$ holds true. If this equation holds, it signifies that the vehicle is legitimate and that M_1 has not been tampered with; otherwise, all subsequent operations are abandoned;

(3) If the vehicle is legitimate, CA calculates a $k * n$ symmetrical keys matrix $SKM_v = \{sk_a^i = H(y_v R_a - iT_v)\}_{a\in\{1,2,\cdots,k\}, i\in\{1,2,\cdots,n\}}$. Here, k denotes the number of RSUs located along the vehicle's path, and n denotes the total number of all RSUs within a given CA domain. The a-th row of the matrix SKM_v contains n elements. It is

important to note that the n_a-th element of a-th row is specifically equal to the session key sk_a of the vehicle side, and the proof process is as follows.

$$SKM_v[a, n_a] = sk_a^{n_a} = H(y_v R_a - iT_v) = H(y_v(n_a S_v + x_a P) - iT_v)$$
$$= H(n_a T_v + x_a S_v - iT_v) = H((n_a - i)T_v + x_a S_v) = H(x_a S_v) = sk_a$$

From the perspective of CA, it is unaware of the value n_a that corresponds to the a-th row of the SKM_v matrix. Consequently, it does not know which RSU's authentication key is being correctly transmitted. Assuming that $n = 6$, meaning there are 6 RSUs in the entire CA domain, the vehicle needs to traverse 4 RSUs during a specific trip, that is, $k = 4$. The serial numbers of the RSUs that the vehicle will traverse are ($n_1 = 3$, $n_2 = 2$, $n_3 = 6$, $n_4 = 1$) in CA. Then, the SKM_v matrix can be derived, as illustrated in Equation (1). The four blue elements (sk_1^3, sk_2^2, sk_3^6, sk_4^1) are equal to the four session keys (sk_1, sk_2, sk_3, sk_4) at the vehicle end, respectively;

$$SKM_v = \begin{pmatrix} sk_1^1 & sk_1^2 & sk_1^3 & sk_1^4 & sk_1^5 & sk_1^6 \\ sk_2^1 & sk_2^2 & sk_2^3 & sk_2^4 & sk_2^5 & sk_2^6 \\ sk_3^1 & sk_3^2 & sk_3^3 & sk_3^4 & sk_3^5 & sk_3^6 \\ sk_4^1 & sk_4^2 & sk_4^3 & sk_4^4 & sk_4^5 & sk_4^6 \end{pmatrix} \quad (1)$$

(4) CA uses the elements of the matrix SKM_v to encrypt the authentication keys of all n RSUs by calculating $EMM_v = \{c_a^i = Eyn(sk_a^i, K_u^i)\}_{a\in\{1,2,\ldots,k\}, i\in\{1,2,\ldots,n\}}$. The results for each element in EMM_v are shown in Equation (2);

$$EMM_v = \begin{pmatrix} Eyn(sk_1^1, K_u^1) & Eyn(sk_1^2, K_u^2) & Eyn(sk_1^3, K_u^3) & Eyn(sk_1^4, K_u^4) & Eyn(sk_1^5, K_u^5) & Eyn(sk_1^6, K_u^6) \\ Eyn(sk_2^1, K_u^1) & Eyn(sk_2^2, K_u^2) & Eyn(sk_2^3, K_u^3) & Eyn(sk_2^4, K_u^4) & Eyn(sk_2^5, K_u^5) & Eyn(sk_2^6, K_u^6) \\ Eyn(sk_3^1, K_u^1) & Eyn(sk_3^2, K_u^2) & Eyn(sk_3^3, K_u^3) & Eyn(sk_3^4, K_u^4) & Eyn(sk_3^5, K_u^5) & Eyn(sk_3^6, K_u^6) \\ Eyn(sk_4^1, K_u^1) & Eyn(sk_4^2, K_u^2) & Eyn(sk_4^3, K_u^3) & Eyn(sk_4^4, K_u^4) & Eyn(sk_4^5, K_u^5) & Eyn(sk_4^6, K_u^6) \end{pmatrix} \quad (2)$$

(5) CA generates a timestamp t_2 and a temporary session key K_v with the vehicle, where $K_v = H(R_{Fv}||t_2)$, $\phi_2 = H(E_{K_v}(EMM_v)||H(R_{Fv})||t_2||AID_v)$. Next, CA sends M_2 : $\{E_{K_v}(EMM_v), t_2, AID_v, \phi_2\}$ to the vehicle;

(6) On receiving M_2, the vehicle verifies t_2. If it is correct, the vehicle then verifies ϕ_2 by checking whether the equation $\phi_2 = H(E_{K_v}(EMM_v)||H(Rep(PUF(C_v), P_{Fv}))||t_2||AID_v)$ holds. If the checking process passes, the vehicle accepts M_2. In the end, the vehicle gets $E_{K_v}(EMM_v)$;

(7) The vehicle calculates $K_v = H(Rep(PUF(C_v), P_{Fv})||t_2)$ and uses K_v to decrypt $E_{K_v}(EMM_v)$ in order to retrieve EMM_v, and then retrieve the needed authentication keys by calculating $\{Dec(sk_a, Eyn(sk_a^{n_a}, K_u^{n_a}))\}_{a=1}^k$. In the above example, in order to obtain k desired RSUs, only 4 decryption operations need to be performed, i.e., $K_u^3 = Dec(sk_1, Eyn(sk_1^3, K_u^3))$, $K_u^2 = Dec(sk_2, Eyn(sk_2^2, K_u^2))$, $K_u^6 = Dec(sk_3, Eyn(sk_3^6, K_u^6))$, $K_u^1 = Dec(sk_4, Eyn(sk_4^1, K_u^1))$;

(8) In the event of physical attacks, such as side-channel attacks, the long-stored authentication keys $\{K_u^a\}_{a=1}^k$ may be compromised. To prevent this, we employ a K_v to individually encrypt each authentication key, namely, $\{E_{K_v}(K_u^a)\}_{a=1}^k$.

In the aforementioned process of this section, the vehicle simultaneously requests authentication keys from all the RSUs it encounters. To avoid path leakage, it employs OT. As a result, the entire process becomes relatively complicated. Now, we will analyze the time complexity. In the above process, the primary time consumption stems from the computation of $SKM_v$and EMM_v. Assuming there are n RSUs under the CA domain and the vehicle requires obtaining k RSUs prior to a specific trip, then SKM_v comprises $n * k$ elements. When calculating each element in SKM_v, Only two scalar multiplication operations on ECC, one addition operation on ECC and one hash operation are necessary, Consequently, the time complexity of computing SKM_v is approximately $\mathcal{O}(n * k)$, and the same time complexity applies to calculating EMM_v, which is also about $\mathcal{O}(n * k)$. All other

operations within this section exhibit either constant or linear time complexity with respect to k. Therefore, the total time complexity of this section is dominated by the $\mathcal{O}(n * k)$. Since the number n of RSUs in a CA domain is not very large, and the number k of RSUs on the path of a vehicle during a certain trip is relatively small, generally less than 50. The time complexity in this section falls within an acceptable range.

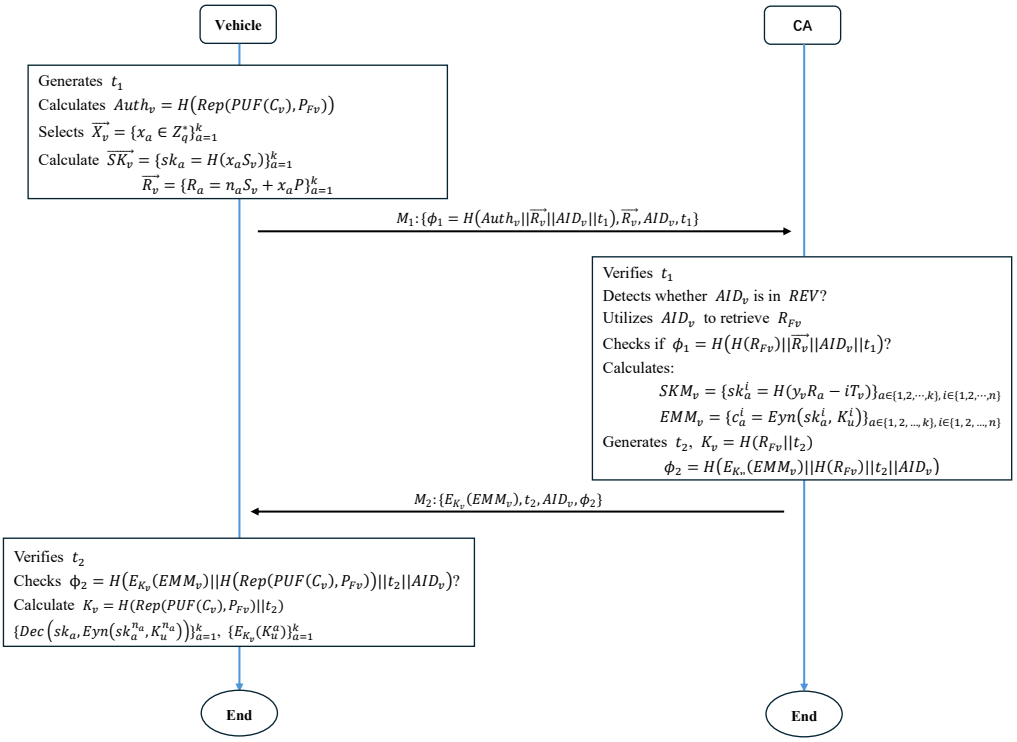

Figure 4. Route planning.

4.5. Authentication

Given that the vehicle has previously acquired the ciphertext of the RSUs' authentication key during route planning, the vehicle simply needs to decrypt the ciphertext before entering the coverage of the RSU to obtain the RSU's authentication key. Subsequently, the vehicle and RSU mutually authenticate each other. The interactive implementation process between the vehicle and RSU is shown in Figure 5, and the details are as follows.

(1) When the vehicle drives into the coverage range of RSU, the vehicle computes the temporary session key $K_v = H(Rep(PUF(C_v), P_{Fv})||t_2)$. Next, the vehicle is able to obtain the authentication key of the RSU by computing $K_u^a = D_{K_v}(E_{K_v}(K_u^a))$. Next, the vehicle generates a timestamp t_3 and a random number rn. And the vehicle sends $M_3 : \{E_{K_u^a}(AID_v||rn||t_3), t_3, \phi_3 = H(E_{K_u^a}(AID_v||rn||t_3)||t_3)\}$ to the RSU;

(2) On receiving M_3, the RSU verifies t_3 and ϕ_3. If both are correct, the RSU computes $K_u^a = H(Rep(PUF(C_u), P_{Fu}))$, then decrypts the $E_{K_u^a}(AID_v||rn||t_3)$ to obtain AID_v and rn using K_u^a. Last, the RSU sends $H(rn+1)$ to the vehicle;

(3) On receiving the response, the vehicle checks whether $H(rn+1)$ is correct. If the checking process passes, the authentication is successful. Otherwise, it fails.

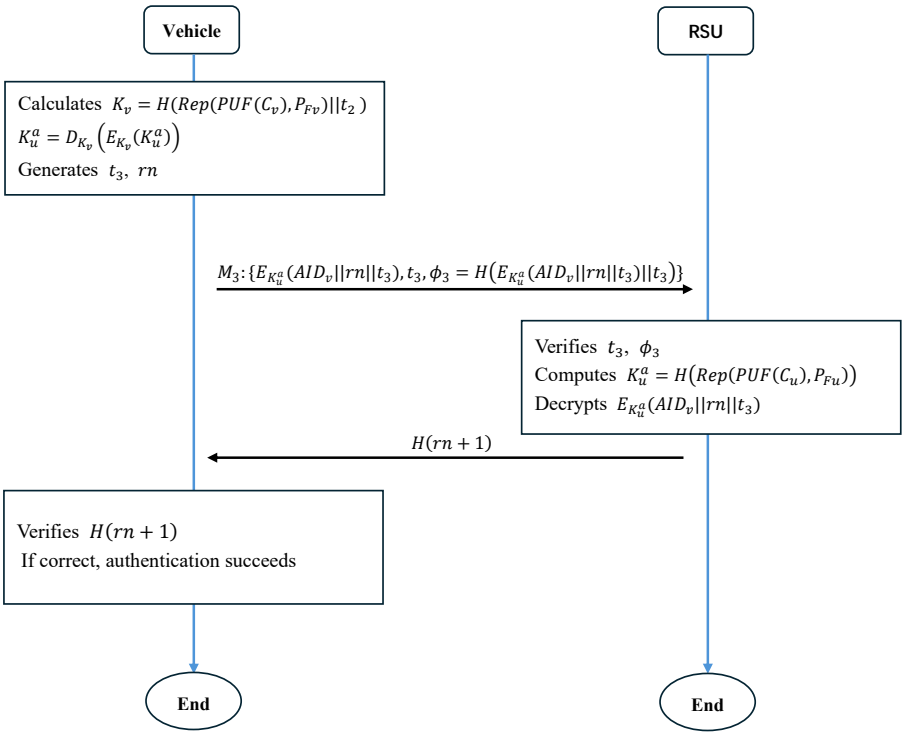

Figure 5. Authentication.

4.6. CRP Update

To ensure freshness, the CRPs of vehicles and RSUs need be updated after a certain time. Here we use the vehicle as an example to describe how to update, and the CRPs of RSUs are updated in the same way. The detailed update steps are as follows.

(1) The vehicle generates a new CRP (C_v^n, R_v^n) by PUF chips employed on the vehicle, where $R_v^n = PUF(C_v^n)$. Calculate $(P_{Fv}^n, R_{Fv}^n) = Gen(R_v^n)$. Next, the vehicle calculates $R_{Fv} = Rep(PUF(C_v), P_{Fv}), \overline{C}_v^n = C_v^n \oplus R_{Fv}$ and $\overline{R}_{Fv}^n = R_{Fv}^n \oplus R_{Fv}$. And the vehicle generates a timestamp t_4 and $\phi_4 = H(AID_v||R_{Fv}||t_4)$. Finally, the vehicle sends $M_4 : \{AID_v, \overline{C}_v^n, \overline{R}_{Fv}^n, t_4, \phi_4\}$ to CA;

(2) On receiving M_4, CA verifies ϕ_4 and t_4. If both are correct, CA calculates $C_v^n = \overline{C}_v^n \oplus R_{Fv}, R_{Fv}^n = \overline{R}_{Fv}^n \oplus R_{Fv}$. Then, CA deletes the old CRP (C_v, R_{Fv}) from its database and updates it with a new CRP (C_v^n, R_{Fv}^n).

4.7. Pseudonym Update

To prevent attackers from contacting multiple messages from the same vehicle, the pseudonym of a vehicle needs to be updated periodically. The detailed steps for updating are as follows. Firstly, CA selects a new random number r_v^n for a vehicle. And CA generates the pseudonym AID_v^n for the vehicle, where $AID_v^n = H(sk_c||r_v^n) \oplus ID_v$. Then, CA deletes the old random r_v and the old pseudonym AID_v from its database and updates with the new random number r_v^n and the new pseudonym AID_v^n. Finally, CA sends $\{AID_v^n, ID_v\}$ to the vehicle via a secure method.

4.8. Trace and Revocation

If an RSU or vehicle detects suspicious behavior from a malicious vehicle, it can submit the corresponding pseudonym AID_v^f to CA. Then, CA calculates $ID_v^f = H(sk_c||r_v^f) \oplus AID_v^f$ using the system secret key sk_c, random number r_v and the pseudonym AID_v^f to obtain ID_v^f of the malicious vehicle.

Afterward, the CA removes $\{ID_v^f, AID_v^f, r_v^f, y_v^f, T_v^f, R_{Fv}^f\}$ from the CA's database, Subsequently, it adds the entity $\{ID_v^f, AID_v^f, t^f\}$ to CA's revocation list REV to ensure that the malicious vehicle is no longer able to participate in the system.

5. Security Analysis

In this section, we analyze the security, the computation and communication overhead of the proposed scheme and compare it with some of the recent existing authentication protocols [19–22].

Authentication and Integrity: Assuming an adversary intercepts or modifies the message $M_1 : \{\phi_1 = H(Auth_v||\overrightarrow{R_v}||AID_v||t_1), \overrightarrow{R_v}, AID_v, t_1\}$, then sends the modified message to CA. CA can detect the attack by the equation $\phi_1 = H(H(R_{Fv})\|\overrightarrow{R_v}\|AID_v\|t_1)$, where $Auth_v = H(R_{Fv})$. Therefore, message integrity is guaranteed.

Physical Protection: In the proposed protocol, instead of storing a secret key in TPD's permanent storage, PUF generates a secret response R for generating a secret key. Furthermore, when a challenge C is inputted, the PUF provides a response R. As the secret response R is exclusively generated by the PUF upon request, attackers are unable to extract any responses from the memory of a vehicle or RSU. Even if an attacker manages to acquire a challenge C, the unclonable nature of the PUF prevents them from deducing the response R from challenge C. Consequently, any endeavor by an adversary to obtain the physical secret key will not be successful.

Anonymity: In the registration phase of the vehicle, the CA generates the pseudonym of a vehicle using the formula $AID_v = H(sk_c||r_v) \oplus ID_v$. Subsequently, the true identity of the vehicle is concealed within this pseudonym. To deduce the real identity from AID_v, RSUs and other vehicles would need access to both sk_c and r_v. However, this crucial information is exclusively stored within the CA's database and is accessible solely by the CA. Consequently, neither RSUs nor other vehicles can obtain this information, rendering them incapable of deducing the real identity from AID_v. Thus, anonymity is effectively ensured.

Traceability: Upon the dispute of a message, the CA possesses the capability to extract the true identity of the vehicle. Given that the secret key sk_c and the random value r_v are stored within the database of CA, the CA is able to ascertain the vehicle's real identity through the computation of $ID_v^f = H(sk_c||r_v^f) \oplus AID_v^f$. After obtaining the real identity ID_v^f of the malicious vehicle, you can revoke the vehicle from the CA; for further details, please refer to Section 4.8. Other entities do not have the ability to revoke and track the malicious vehicle, because they do not have the main private key sk_c of the system and random number r_v^f

Finally, the security comparison results presented in Table 1 demonstrate that our protocol offers superior advantages.

Table 1. Security comparison.

Scheme	Authentication and Integrity	Physical Protection	Anonymity	Traceability
[19]	✓	✓	×	×
[20]	✓	✓	✓	×
[21]	✓	×	✓	✓
[22]	✓	×	✓	×
Ours	✓	✓	✓	✓

✓: Implemented; ×: Not implemented.

6. Performance Evaluation

Since initialization, vehicle registration, RSU registration, and path planning in our scheme are all one-time operations that require less computation and communication overhead throughout the entire implementation process, this section focuses solely on

comparing the computational and communication overhead incurred during the authentication phase.

6.1. Computational Overhead Comparison

To facilitate the comparison of computational overhead, this experiment was conducted on a laptop equipped with an Intel i5-8300H processor (2.3 GHz) and 16 GB of memory. By repeatedly executing operations with different input values and taking the average, we obtained the execution times of common cryptographic operations. Specifically, we represented the time for hash function operations as T_h, the execution time for scalar multiplication on elliptic curve cryptography (ECC) as T_m, and the execution times for encryption and decryption algorithms of AES uniformly as T_a (although typically the decryption function of AES takes longer than the encryption function, the difference in execution time between the encryption and decryption algorithms is negligible for small data volumes). Furthermore, the PUF deployed on vehicles and RSUs adopted the ring oscillator algorithm. The time required to apply a 128-bit challenge to the PUF and generate the corresponding 128-bit response was recorded as 9 microseconds (μs) [23], we represented the execution times for PUF as T_p, the execution times for the function $Rep(W', P_F)$ of FE as T_{gen}, and the execution times for the function $Gen(W)$ of FE as T_{rep}. The times required for these operations are detailed in Table 2, which serves as the basis for subsequent computational overhead. Notably, the execution times for concatenation operations $(||)$ and XOR operations $(\oplus)$ are negligible compared to the times listed in Table 2, so we do not consider the computational time for these two operations.

Table 2. The execution time of the basic operations.

Notation	Operation	Time (ms)
T_h	Hash function	0.001
T_m	Scale multiplication of ECC	0.383
T_a	AES encryption and decryption	0.011
T_p	The 128-bit PUF operation	0.009
T_{gen}	Function $Gen(W)$ of the fuzzy extractor with a 128-bit input	0.023
T_{rep}	Function $Rep(W', P_F)$ of the fuzzy extractor with a 128-bit input	0.010

Moreover, in the authentication phase of the proposed scheme, there are 6 hash operations, an AES encryption operation, 2 AES decryption operations, 2 function $Rep(W', P_F)$ operations of the fuzzy extractor, and 2 generating a 128-bit response of PUF operations that need to be performed. Furthermore, we compare the verification time of the proposed scheme with schemes in [19–22]. The results in Table 3 and Figure 6 show that our proposed scheme computation overhead is better than others.

Table 3. The computation overhead in the authentication phase.

Scheme	Communicate with a RSU	Communicate with n RSUs
[19]	$4T_m + T_p + 10T_h$	$n(4T_m + T_p + 10T_h)$
[20]	$8T_a + 2T_p + 14T_h$	$n(4T_a + 2T_p + 10T_h) + 4(T_a + T_h)$
[21]	$3T_m + 2T_h$	$n(T_m + 2T_h) + 2T_m$
[22]	$5T_m + 2T_h$	$n(5T_m + 2T_h)$
Ours	$3T_a + 2T_p + 2T_{rep} + 6T_h$	$n(3T_a + 2T_p + 2T_{rep} + 6T_h)$

RSU: road-side unit

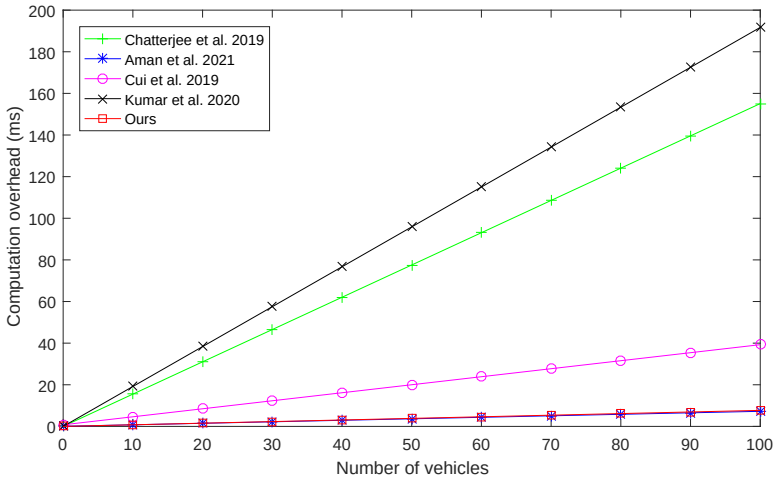

Figure 6. The computation overhead in authentication. Chatterjee et al. proposed the scheme [19], Aman et al. proposed the scheme [20], Cui et al. proposed the scheme [21], and Kumar et al. proposed the scheme [22].

6.2. Communication Overhead Comparison

Additionally, to compare the proposed protocol with the existing authentication schemes according to communication overhead, we need to assume the parameter sizes. Since p' and p are prime numbers of 64 bytes (512 bits) and 20 bytes (160 bits), respectively, the length of the elements in G_1 and G_2 are 128 bytes and 40 bytes separately. We assume the length of a one-way hash function's output is 20 bytes, the length of a timestamp is 4 bytes, the length of an identity is 20 bytes, and the length of the symmetric key encryption or decryption (AES-512) function's output is 64 bytes. Finally, we present the communication costs of our scheme and other schemes in Table 4. In the proposed scheme, Figure 5 shows that the communication messages between one RUS and a vehicle in the authentication phase are $M_2 : \{E_{K_v}(K_u), t_2, \phi_2 = H(E_{K_v}||H(R_v)||t_2)\}$ and $H(rn+1)$. Hence, the total communication costs of our scheme are $(64+4+20)+20 = 108$ bytes and $108n$ bytes for n RUSs. The cost of communication of the others can be calculated in the same way. And the results in Table 4 and Figure 7 show the proposed scheme communication overhead is less than others.

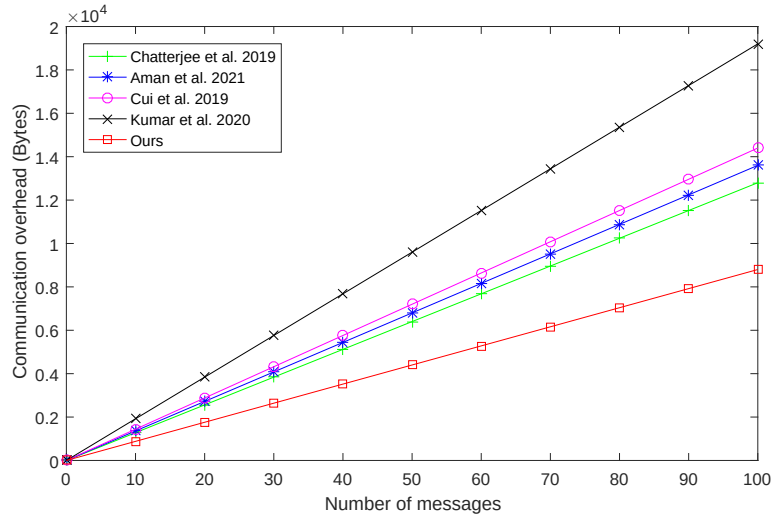

Figure 7. The communication overhead in authentication. Chatterjee et al. proposed the scheme [19], Aman et al. proposed the scheme [20], Cui et al. proposed the scheme [21], and Kumar et al. proposed the scheme [22].

Table 4. The communication overhead.

Scheme	Single Message (Bytes)	n Messages (Bytes)
[19]	128	128n
[20]	136	136n
[21]	144	144n
[22]	192	192n
Ours	108	108n

6.3. Packet Loss Rate Evaluation

In order to analyze the network stability of each scheme in the authentication phase, we conducted simulation experiments on the data packet loss rate (PLR). We utilize OMNeT++ 5.6.2, combined with SUMO 1.8.0, inet 4.2.5, and Veins 5.2, on a Windows 11 operating system. SUMO (Simulation of Urban MObility) is an open source, highly portable, microscopic and continuous multi-modal traffic simulation package designed to handle large networks. The main parameters of the simulation environment are shown in Table 5.

In our simulation experiments, the map and all road configurations adopt the default settings provided by SUMO. Figure 8 shows the simulation process of our scheme. The yellow rsu[0] in the figure represents the RSU, the green nodes represent vehicles that have completed mutual authentication with the RSU, and the red node represents a vehicle in the process of mutual authentication. The blue dotted lines indicate the process of information transmission.

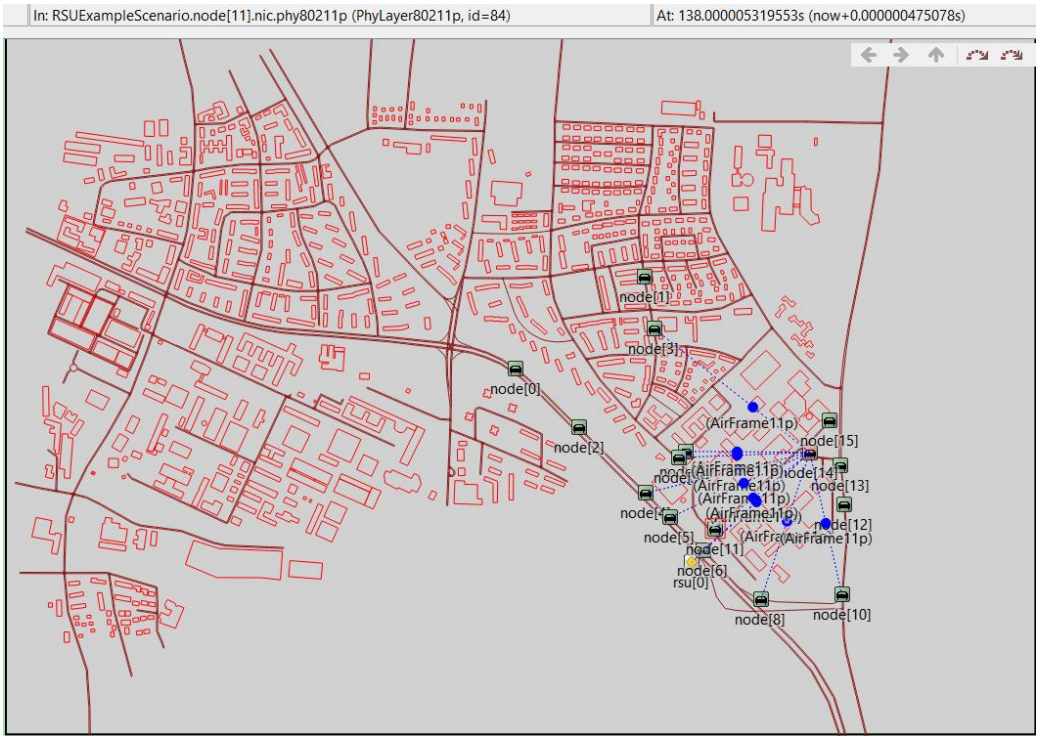

Figure 8. Simulation process of packet loss rate evaluation in the authentication phase.

Table 5. Environment configuration for simulation.

Parameters	Values
Area	3000 m × 2500 m × 50 m
MAC Layer	802.11p
Data Rate	5 Mb/s
Broadcast Interval	1000 m
Number of RSUs	1
Number of Vehicles	20–100
Vehicle Speed	5–30 m/s
Simulation Time	450 s

We define the packet loss rate PLR as shown in Equation (3) [24]. N_l represents the total number of RSU lost packets, and N_r represents the total number of RSU accepted packets.

$$PLR = \left(\frac{N_l}{N_l + N_r}\right) * 100\% \tag{3}$$

For each scheme, when the number of vehicles in the simulation scene was 20, 40, 60, 80 and 100, we conducted a set of experiments, respectively. Each set of experiments consisted of ten trials, during which we counted the packet loss rate for each trial, and then calculated the average packet loss rate for the ten trials. The simulation results of each scheme with four different vehicle numbers were obtained, as shown in Figure 9. From the simulation results, compared with other algorithms, our scheme exhibits obvious advantages in terms of packet loss rate. The main reason is that our scheme has the lowest communication and calculation costs during the authentication stage. Meanwhile, as the number of vehicles in the simulation scenario increases, the packet loss rate also rises accordingly. This is because when the number of vehicles increases, the communication load in the entire scene rises sharply, and network congestion and mutual interference between pieces of information will also increase.

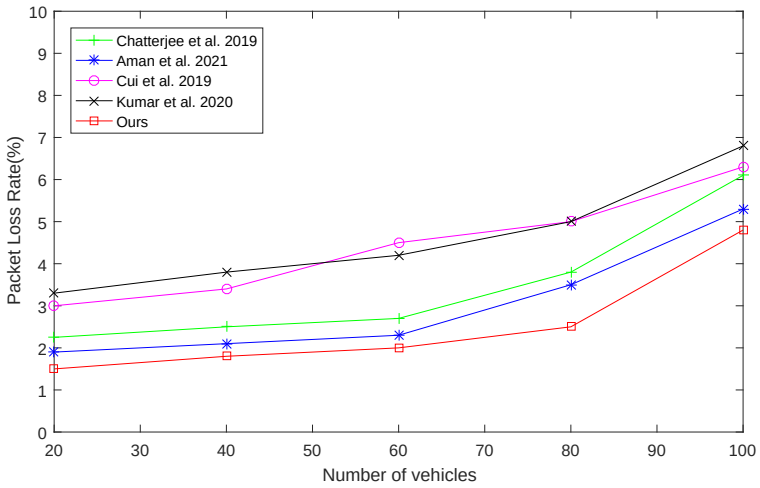

Figure 9. Simulation result of packet loss rate. Chatterjee et al. proposed the scheme [19], Aman et al. proposed the scheme [20], Cui et al. proposed the scheme [21], and Kumar et al. proposed the scheme [22].

7. Conclusions

Many authentication schemes overlook the potential vulnerability of vehicles to physical attacks, which could compromise the confidentiality of the secret key. Indeed, side-channel attacks such as power consumption, electromagnetic radiation, and timing analysis remain viable avenues for accessing the secret key stored in TPDs. To mitigate the risk of physical theft of the secret key, we propose a physically preserving authentication approach

based on PUFs for VANETs. In our proposed scheme, PUFs are utilized to generate the secret key only when necessary for V2I authentication, eliminating the need for storing the secret key in the permanent storage of TPDs. This ensures the physical security of the system. Moreover, we integrate route planning into the protocol to enhance the efficiency of V2I authentication. Comparative analysis with recent schemes reveals that our proposed approach exhibits lower computational and communication overhead, while still satisfying fundamental security requirements such as authentication, integrity, anonymity, and traceability.

8. Future Works

Although this paper has conducted a comprehensive theoretical and simulation analysis of our proposed scheme from both security and performance perspectives, due to time constraints and realistic limitations, our scheme has yet to be deployed and tested in real-world scenarios. Addressing potential issues that may arise during real-world deployment is one of our future research directions. In addition, our proposed scheme, in order to obtain the authentication key of all routes in the vehicle driving path at one time and ensure the privacy of the path, The computational cost is slightly higher. How to reduce the calculation cost without compromising the existing performance is also a research direction of ours for the future.

Author Contributions: Methodology, L.L., Z.Z. and S.-L.P.; Writing—original draft, L.L. and H.D.; Writing—review and editing, L.L.; Visualization, L.L. and H.D.; Resources, H.D.; Validation, L.L.; Supervision, Z.Z. and S.-L.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank the editor and the anonymous reviewers for their detailed comments and suggestions regarding this article.

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

References

1. Zhang, C.; Lu, R.; Lin, X.; Ho, P.H.; Shen, X. An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks. In Proceedings of the IEEE INFOCOM 2008—The 27th Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008; pp. 246–250. [CrossRef]
2. Kenney, J.B. Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proc. IEEE* **2011**, *99*, 1162–1182. [CrossRef]
3. Jiang, D.; Delgrossi, L. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. In Proceedings of the VTC Spring 2008—IEEE Vehicular Technology Conference, Singapore, 11–14 May 2008; pp. 2036–2040. [CrossRef]
4. Chen, J.; Yan, H.; Liu, Z.; Zhang, M.; Xiong, H.; Yu, S. When Federated Learning Meets Privacy-Preserving Computation. *ACM Comput. Surv.* **2024**. [CrossRef]
5. Xia, Y.; Liu, Y.; Dong, S.; Li, M.; Guo, C. SVCA: Secure and Verifiable Chained Aggregation for Privacy-Preserving Federated Learning. *IEEE Internet Things J.* **2024**, *11*, 18351–18365. [CrossRef]
6. Wang, Y.; Wang, X.; Dai, H.N.; Zhang, X.; Imran, M. A Data Reporting Protocol With Revocable Anonymous Authentication for Edge-Assisted Intelligent Transport Systems. *IEEE Trans. Ind. Inform.* **2023**, *19*, 7835–7847. [CrossRef]
7. Chen, J.; Wang, Z.; Srivastava, G.; Alghamdi, T.A.; Khan, F.; Kumari, S.; Xiong, H. Industrial blockchain threshold signatures in federated learning for unified space-air-ground-sea model training. *J. Ind. Inf. Integr.* **2024**, *39*, 100593. [CrossRef]
8. Alladi, T.; Chamola, V.; Naren; Kumar, N. PARTH: A two-stage lightweight mutual authentication protocol for UAV surveillance networks. *Comput. Commun.* **2020**, *160*, 81–90. [CrossRef]
9. Yan, W.; Tehranipoor, F.; Chandy, J.A. PUF-Based Fuzzy Authentication without Error Correcting Codes. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2017**, *36*, 1445–1457. [CrossRef]
10. Hiraishi, H. Passenger Condition Based Route-Planning for Cognitive Vehicle System. *Int. J. Softw. Sci. Comput. Intell.* **2018**, *10*, 25–35. [CrossRef]
11. Liang, Y.; Liu, Y.; Gupta, B.B. PPRP: Preserving-Privacy Route Planning Scheme in VANETs. *ACM Trans. Internet Technol.* **2022**, *22*, 85. [CrossRef]

12. Zhu, L.; Li, M.; Zhang, Z.; Xu, C.; Zhang, R.; Du, X.; Guizani, N. Privacy-Preserving Authentication and Data Aggregation for Fog-Based Smart Grid. *IEEE Commun. Mag.* **2019**, *57*, 80–85. [CrossRef]
13. Zhang, J.; Zhong, H.; Cui, J.; Tian, M.; Xu, Y.; Liu, L. Edge Computing-Based Privacy-Preserving Authentication Framework and Protocol for 5G-Enabled Vehicular Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 7940–7954. [CrossRef]
14. Cui, J.; Wei, L.; Zhang, J.; Xu, Y.; Zhong, H. An Efficient Message-Authentication Scheme Based on Edge Computing for Vehicular Ad Hoc Networks. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1621–1632. [CrossRef]
15. Hathal, W.; Cruickshank, H.; Sun, Z.; Maple, C. Certificateless and Lightweight Authentication Scheme for Vehicular Communication Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 16110–16125. [CrossRef]
16. Cui, J.; Zhang, X.; Zhong, H.; Ying, Z.; Liu, L. RSMA: Reputation System-Based Lightweight Message Authentication Framework and Protocol for 5G-Enabled Vehicular Networks. *IEEE Internet Things J.* **2019**, *6*, 6417–6428. [CrossRef]
17. Badar, H.M.S.; Qadri, S.; Shamshad, S.; Ayub, M.F.; Mahmood, K.; Kumar, N. An Identity Based Authentication Protocol for Smart Grid Environment Using Physical Uncloneable Function. *IEEE Trans. Smart Grid* **2021**, *12*, 4426–4434. [CrossRef]
18. Yevgeniy, D.; Leonid, R.; Adam, S. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In Proceedings of the Advances in Cryptology—EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 523–540.
19. Chatterjee, U.; Govindan, V.; Sadhukhan, R.; Mukhopadhyay, D.; Chakraborty, R.S.; Mahata, D.; Prabhu, M.M. Building PUF Based Authentication and Key Exchange Protocol for IoT without Explicit CRPs in Verifier Database. *IEEE Trans. Dependable Secur. Comput.* **2019**, *16*, 424–437. [CrossRef]
20. Aman, M.N.; Javaid, U.; Sikdar, B. A Privacy-Preserving and Scalable Authentication Protocol for the Internet of Vehicles. *IEEE Internet Things J.* **2021**, *8*, 1123–1139. [CrossRef]
21. Cui, J.; Wu, D.; Zhang, J.; Xu, Y.; Zhong, H. An Efficient Authentication Scheme Based on Semi-Trusted Authority in VANETs. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2972–2986. [CrossRef]
22. Kumar, V.; Ahmad, M.; Mishra, D.; Kumari, S.; Khan, M.K. RSEAP: RFID based secure and efficient authentication protocol for vehicular cloud computing. *Veh. Commun.* **2020**, *22*, 100213. [CrossRef]
23. Liang, Y.; Liu, Y.; Zhang, X.; Liu, G. Physically Secure and Privacy-Preserving Charging Authentication Framework with Data Aggregation in Vehicle-to-Grid Networks. *IEEE Trans. Intell. Transp. Syst.* **2024**. [CrossRef]
24. Su, H.; Dong, S.; Zhang, W.T. An efficient privacy-preserving authentication scheme that mitigates TA dependency in VANETs. *Veh. Commun.* **2024**, *45*, 100727. [CrossRef]