

## Article

# LEPA: Low-Overhead and Efficient Privacy-Preserving Authentication Scheme in VANETs <sup>†</sup>

Shafika S. Moni <sup>1</sup>  and Dakshnamoorthy Manivannan <sup>2,\*</sup>

<sup>1</sup> Department of Electrical Engineering and Computer Science, Embry Riddle Aeronautical University, Daytona Beach, FL 32114, USA; monis@erau.edu

<sup>2</sup> Department of Computer Science, University of Kentucky, Lexington, KY 40508, USA

\* Correspondence: manivann@cs.uky.edu; Tel.: +1-(859)-257-9234

<sup>†</sup> This paper is a revised and extended version of our paper entitled “A lightweight Privacy-Preserving V2I Mutual Authentication Scheme using Cuckoo Filter in VANETs” which was presented in 2022 IEEE 19th Annual Consumer Communications and Networking Conference (CCNC), 8–11 January 2022, Las Vegas, NV, USA.

## Abstract

The dynamic nature of Vehicular Ad-hoc Networks (VANETs) necessitates robust authentication mechanisms to prevent adversaries from compromising vehicle privacy. To address privacy concerns, many existing approaches employ pseudonyms in place of real vehicle identities. However, the use of a single pseudonym is insufficient, as vehicle trajectories can still enable tracking. Consequently, vehicles must frequently change pseudonyms, typically selecting them from a pre-assigned pool, to ensure unlinkability and preserve privacy. In most existing schemes, a central authority issues certificates corresponding to each pseudonym, which vehicles present for authentication. While effective, this approach incurs significant computation, storage, and communication overhead, particularly in managing certificate revocation lists (CRLs), since each vehicle may possess a large number of pseudonyms. To address these challenges, we propose a Low-overhead and Efficient Privacy-preserving Authentication (LEPA) scheme for VANETs, leveraging Merkle Hash Trees (MHTs) and Cuckoo Filters (CFs) to efficiently manage pseudonym sets and revocation. We analyze the security of the proposed scheme against various attacks and demonstrate, through performance evaluation, that LEPA significantly reduces authentication and revocation overhead while maintaining strong privacy and security guarantees.

**Keywords:** VANETs; privacy-preserving; Merkle Hash Tree (MHT); Cuckoo Filter; authentication in VANETs



Academic Editor: Pingyi Fan

Received: 12 March 2026

Revised: 16 April 2026

Accepted: 6 May 2026

Published: 9 May 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

With the rapid advancement of Vehicular Ad-hoc Networks (VANETs), the development of Intelligent Transportation Systems (ITS) has gained significant momentum. VANETs aim to enhance road safety, improve traffic efficiency, and provide infotainment services through seamless communication among vehicles and Road Side Units (RSUs) via Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) links. Each vehicle is equipped with an On-Board Unit (OBU) to enable the exchange of safety-critical and non-critical information, such as accident alerts and traffic conditions.

However, the open and wireless nature of VANET communications makes them highly vulnerable to various attacks, including message modification, replay, and impersonation. As a result, secure and efficient authentication mechanisms are essential to ensure message

integrity, verify the legitimacy of communicating entities, and prevent unauthorized access. At the same time, preserving user privacy is equally critical, as direct use of vehicle identities can enable adversaries to track vehicle movements and compromise sensitive information.

To address these concerns, privacy-preserving authentication schemes based on pseudonyms have been widely adopted. In such approaches, a Trusted Authority (TA) assigns each vehicle a pool of pseudonyms and corresponding certificates, which are used in place of real identities during communication. Vehicles periodically change pseudonyms to achieve unlinkability, while conditional privacy ensures that only authorized entities can reveal the true identity of misbehaving vehicles.

Despite their effectiveness, these schemes introduce significant overhead in certificate management and revocation. In particular, when a vehicle is identified as malicious, all certificates associated with its pseudonyms must be revoked and included in a Certificate Revocation List (CRL). As the number of pseudonyms and revoked vehicles increases, the CRL grows substantially, leading to increased storage, communication, and verification overhead. Moreover, each receiver must perform CRL checks for every incoming message, which degrades authentication efficiency and limits scalability.

To overcome these limitations, we propose a *Low-overhead and Efficient Privacy-preserving Authentication (LEPA)* scheme for VANETs. The proposed approach leverages Merkle Hash Trees (MHTs) and Cuckoo Filters (CFs) to efficiently manage pseudonym sets and revocation, significantly reducing storage and verification overhead. We further analyze the security of LEPA against various attacks and demonstrate, through performance evaluation, that it achieves improved efficiency while preserving strong privacy and security guarantees.

#### *Contribution of This Paper*

The LEPA scheme proposed in this paper overcomes the scalability, storage, and communication limitations of conventional CRL-based approaches. In LEPA, each registered vehicle is provisioned with a set of pseudonyms by its Local Trusted Authority (LTA), enabling conditional privacy while supporting accountability. To efficiently manage these pseudonyms, the scheme employs a Merkle Hash Tree (MHT) [1], where each vehicle's pseudonym set is compactly represented by a single MHT, thereby reducing storage and verification complexity. Unlike some of the approaches proposed in the literature, instead of maintaining large CRLs, the LTA utilizes positive and negative Cuckoo Filters (CFs) [2] to manage valid and revoked entities in a lightweight manner. Specifically, the MHT root corresponding to a legitimate vehicle is inserted into the positive filter, while, upon detection of a malicious vehicle, only the corresponding root (rather than all individual pseudonyms assigned to it) is inserted into the negative filter. A similar dual-filter mechanism is adopted for managing the trusted and compromised in terms of Road Side Units (RSUs). During authentication, both vehicles and RSUs only need to retrieve and query the latest Cuckoo Filters from the LTA, enabling fast membership verification with minimal communication overhead. By leveraging the compactness and constant-time operations of Cuckoo Filters, LEPA significantly reduces storage, computation, and communication costs while improving revocation efficiency and scalability compared to traditional CRL-based schemes. The notations used in this paper are summarized in Table 1.

The remainder of the paper is organized as follows: In Section 2, we present a brief description of the Merkle Hash Tree (MHT) and Cuckoo Filter (CF). Section 3 introduces our system model. In Section 4, we describe our proposed low-overhead and efficient privacy-preserving authentication (LEPA) scheme. Section 5 presents the performance analysis and comparison of our scheme with other related schemes. We discuss some of the related works in Section 6. Finally, Section 8 concludes the paper.

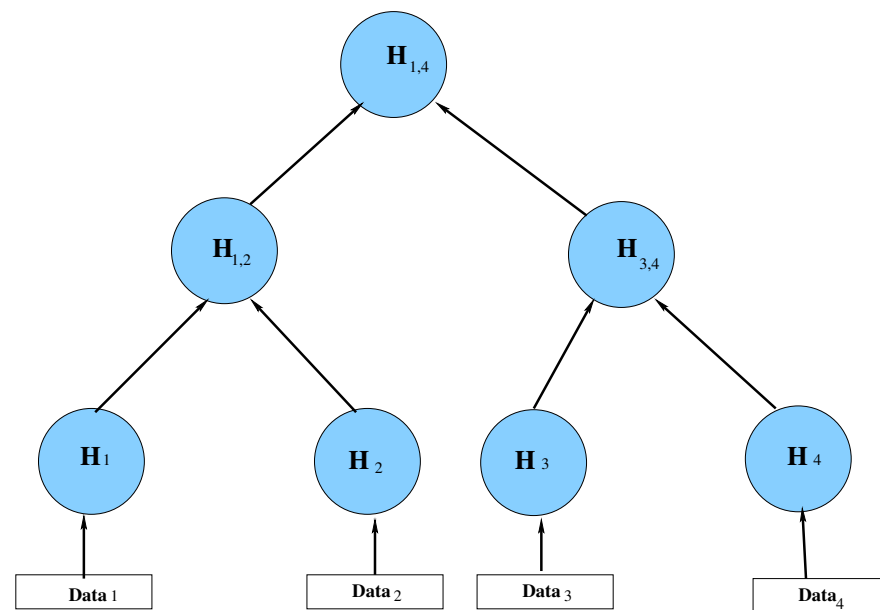
**Table 1.** Notation and Description. Reprinted with permission from [3]. Copyright Year:2022; Copyright Owner's Name: IEEE.

| Notation             | Description                                                                      |
|----------------------|----------------------------------------------------------------------------------|
| TA                   | Trusted Authority                                                                |
| LTA                  | Local TA                                                                         |
| RSU                  | Road Side Unit                                                                   |
| OBU                  | On Board Unit                                                                    |
| $PU_{TA}, PR_{TA}$   | Public and Private Keys of the TA                                                |
| $PU_{LTA}, PR_{LTA}$ | Public and Private Keys of LTA                                                   |
| $PU_{RSU}, PR_{RSU}$ | Public and Private Keys of RSU                                                   |
| $PU_V, PR_V$         | Public and Private Keys of vehicle V                                             |
| $PID_V$              | Pseudonym or Pseudo ID of a vehicle V                                            |
| $t_s$                | Message generation timestamp                                                     |
| $E$                  | RSA-1024 bit Encryption algorithm                                                |
| $H$                  | SHA-256 hash function                                                            |
| $F$                  | Fingerprint                                                                      |
| CF                   | Cuckoo Filter                                                                    |
| $PCF_{RSU}$          | Positive Cuckoo Filter of valid RSUs                                             |
| $NCF_{RSU}$          | Negative Cuckoo Filter of revoked RSUs                                           |
| $PCF_V$              | Positive Cuckoo Filter of valid Vehicles                                         |
| $NCF_V$              | Negative Cuckoo Filter of revoked Vehicles                                       |
| $B_{ndx}$            | Bucket index in positive Cuckoo Filter $PCF_V$                                   |
| MHT                  | Merkle Hash Tree                                                                 |
| MFVs                 | Missing Fingerprint Values of MHT for the corresponding pseudonym of a Vehicle V |
| $MHT_{root}$         | MHT root corresponding to a set of pseudonyms of a Vehicle V                     |
| $Exp_{MHT_{root}}$   | MHT root expired time                                                            |

## 2. Preliminaries

### 2.1. Merkle Hash Tree

A Merkle Hash Tree (MHT) [1] is a hash-based [4] tree used for data storage and verification in distributed systems. Each leaf node in MHT stores the data, whereas each non-leaf node contains the hash of its children. An MHT with four leaf nodes is shown in Figure 1. We can verify the integrity of  $Data_4$  using the corresponding Missing Hash Values (MHVs) instead of the entire tree structure. The MHVs for  $Data_4$  are  $H_{1,2}$  and  $H_3$ . We can recompute the root hash value from MHVs, first computing  $H_{3,4} = H(H(Data_4), H_3)$  and then  $H'_{1,4} = H(H_{1,2}, H_{3,4})$ .  $Data_4$  is considered as part of the MHT tree if both the calculated and actual MHT root are the same (i.e., if  $H'_{1,4} = H(H_{1,2}, H_{3,4}) = H_{3,4} = H(H(Data_4), H_3)$ ).



**Figure 1.** A Merkle Hash Tree with four leaf nodes.

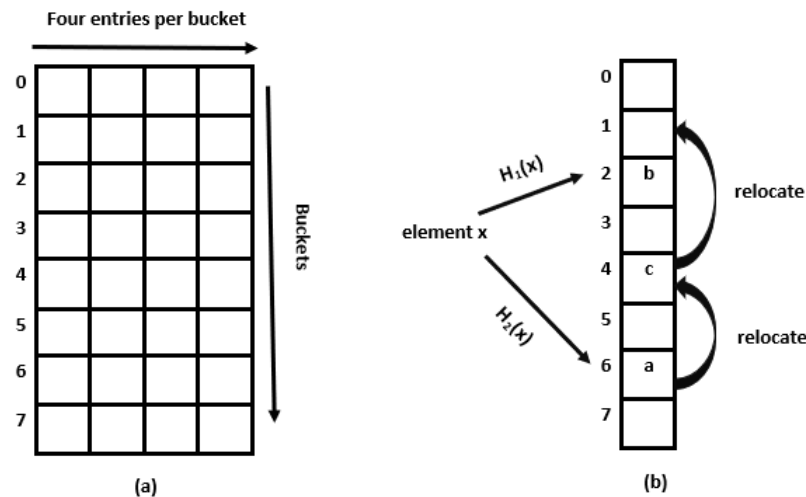
## 2.2. Cuckoo Filter

The Cuckoo Filter (CF) [2] performs set-membership tests faster using a cuckoo hash table [3]. To optimize space efficiency, the CF stores a fingerprint  $F(x)$  of an element  $x$  instead of storing the original element itself. A Cuckoo Filter comprises of a set of  $m$  buckets where each bucket has  $n$  number of entries. Each element  $x$  has two candidate buckets  $i$  and  $j$  determined by two hash functions as follows:

$$i = H_1(x) = \text{hash}(x) \bmod m$$

$$j = H_2(x) = (H_1(x) \oplus \text{hash}(F(x))) \bmod m$$

Figure 2a displays a CF with eight buckets ( $m = 8$ ), where each bucket has four entries ( $n = 4$ ). The Cuckoo Filter uses two hash functions ( $H_1(x)$ ,  $H_2(x)$ ) to find the candidate buckets for an element  $x$  in the filter. It places  $F(x)$  in any of the two empty candidate buckets. The CF randomly selects one of the buckets if both candidates are occupied and replaces the existing fingerprint  $F(y)$  of the element  $y$  in the bucket with  $F(x)$ . It then inserts the  $F(y)$  in its alternate candidate bucket. The alternate bucket location can be found by XORing the hash of the fingerprint of the element and the current bucket location. If none of the buckets are empty, a fingerprint in one of the buckets is displaced. This process is repeated until all displaced fingerprints are re-inserted in the Cuckoo Filter. Figure 2b shows that when inserting an element  $x$ , it finds that both the candidate buckets (bucket numbers 2 and 6) are occupied. Then selects bucket number 6 randomly and replaces the existing element ("a") in the bucket with  $F(x)$ . By displacing the existing item ("c") in bucket number 4, the Cuckoo Filter relocates the kicked out element ("a") in its alternate candidate bucket number 4. It then re-inserts the kicked-out item ("c") into its alternate candidate bucket 1. CF computes the fingerprint of the element  $F(x)$  to look up an element  $x$ . Then, finds the candidate buckets using  $H_1(x)$  and  $H_2(x)$ , and checks the fingerprint  $F(x)$  against the fingerprint stored in these buckets. The Cuckoo Filter returns a positive result whenever a fingerprint matches  $F(x)$ . If none match, it returns a negative one. An element  $x$ 's fingerprint is first looked up in the Cuckoo Filter using the lookup operation to delete it. Next, it deletes  $F(x)$  when it is found in one of the buckets. Both lookup and deletion operations are of time complexity  $O(1)$  for the cuckoo filter.

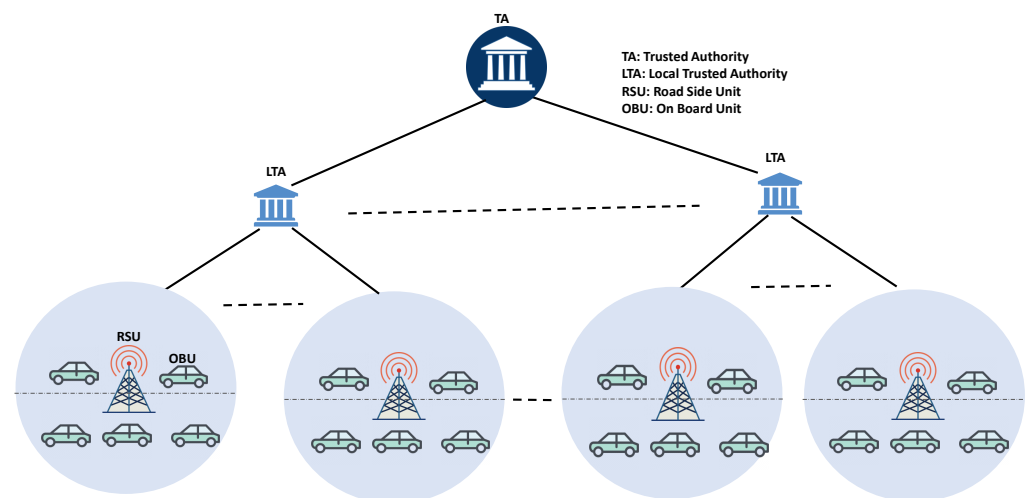


**Figure 2.** (a) A Cuckoo Filter. (b) Insertion operation. Reprinted with permission from [3]. Copyright Year:2022; Copyright Owner's Name: IEEE.

### 3. LEPA System Model

#### 3.1. System Model

The proposed LEPA framework is designed for secure and privacy-preserving authentication in Vehicular Ad-hoc Networks (VANETs). As illustrated in Figure 3, the system consists of four principal entities: the Trusted Authority (TA), multiple Local Trusted Authorities (LTAs), Road Side Units (RSUs), and Vehicles. These entities collaboratively enable scalable authentication, efficient revocation, and strong privacy guarantees.



**Figure 3.** Proposed system model for low-overhead and efficient privacy-preserving authentication in VANET.

- Trusted Authority (TA):** The Trusted Authority (TA) acts as the root of trust in the entire system and is assumed to be fully trusted and tamper-proof. It is responsible for system initialization, global parameter generation, and hierarchical key management. The TA generates its own public–private key pair, denoted by  $(PU_{TA}, PR_{TA})$ . The public key  $PU_{TA}$  is distributed to all system entities, while the private key  $PR_{TA}$  is securely maintained and never disclosed. In addition, the TA is responsible for registering all LTAs and generating a unique public–private key pair  $(PU_{LTA}, PR_{LTA})$  for each LTA. It also maintains a global registry of legitimate LTAs and may assist in dispute resolution, misbehavior tracing, and system-wide revocation when necessary.

- **Local Trusted Authority (LTA):** Each Local Trusted Authority (LTA) operates under the supervision of the TA and manages a specific geographical region. LTAs are responsible for the registration, authentication, and credential management of vehicles and RSUs within their jurisdiction. Upon registration, the LTA generates and assigns a unique public–private key pair to each RSU and vehicle. For vehicles, the LTA additionally issues a pool of pseudonyms along with corresponding credentials. The LTA maintains local authentication records and plays a critical role in revocation management by updating positive and negative membership structures (e.g., Cuckoo Filters in LEPA). Furthermore, LTAs are assumed to form a trusted federation, enabling secure inter-LTA communication. This coordination supports seamless vehicle mobility across regions, consistent authentication, and synchronized revocation information.
- **Road Side Unit (RSU):** RSUs are fixed infrastructure components deployed along roads and intersections, equipped with greater computational and communication capabilities compared to vehicles. Each RSU is registered with its LTA and possesses a unique cryptographic identity. RSUs serve as intermediaries between vehicles and the infrastructure. They are responsible for receiving safety and service messages from vehicles, verifying their authenticity, and broadcasting relevant information to nearby vehicles. RSUs may also relay messages to neighboring RSUs or backend infrastructure to support wide-area dissemination. Due to their strategic placement and extended communication range, RSUs play a key role in enabling low-latency authentication and efficient dissemination of revocation information. Vehicles always communicate through RSUs. If an RSU is not within the range of an RSU, it has to wait until it finds an RSU within its communication range.
- **Vehicle:** Each vehicle is equipped with an On-Board Unit (OBU) that enables communication with other vehicles and RSUs. During initial registration, a vehicle authenticates itself to its home LTA using its real identity. The LTA then provisions the OBU with a set of pseudonyms and associated cryptographic credentials. These pseudonyms act as temporary identifiers, allowing the vehicle to communicate anonymously while preserving unlinkability. To mitigate tracking and correlation attacks, vehicles periodically change their pseudonyms using a secure pseudonym-changing strategy. Vehicles are highly mobile and may traverse multiple LTA regions. When a vehicle enters a new region, it performs a re-registration or handover procedure with the corresponding LTA through local RSU. Upon successful authentication, the new LTA issues a fresh set of pseudonyms, ensuring continuous privacy protection and preventing cross-region tracking. Secure communication among LTAs ensures that vehicle mobility is supported without compromising authentication integrity or privacy. This coordinated pseudonym management mechanism enables scalable, region-aware, and privacy-preserving vehicular communication.

### 3.2. Design Goals

The proposed LEPA scheme is designed to achieve the following objectives:

- **Privacy Preservation:** Privacy is a fundamental requirement in VANETs. Even in scenarios where RSUs are compromised, the real identity of vehicles must remain protected. In the proposed LEPA scheme, only the Local Trusted Authority (LTA) associated with a vehicle has access to its true identity. In the event of malicious behavior, the LTA can disclose the identity exclusively to authorized entities, thereby ensuring accountability while preserving user privacy under normal operation.
- **Low Authentication Overhead:** The scheme enables mutual authentication between vehicles and RSUs while minimizing computational, storage, and communication overhead. To address the inefficiencies of traditional Certificate Revocation Lists (CRLs),

LEPA employs Cuckoo Filters, providing a lightweight and efficient mechanism for revocation checking during authentication.

- **Integrity and Authenticity of Messages:** The scheme ensures that all transmitted messages originate from legitimate entities and remain unaltered during transmission, thereby preserving message integrity and authenticity.
- **Resilience to Replay Attacks:** To prevent replay attacks, each message includes a generation timestamp, allowing vehicles and RSUs to verify message freshness and reject duplicated or outdated transmissions.
- **Unforgeability:** The scheme guarantees that adversaries cannot generate valid authentication messages or credentials on behalf of legitimate vehicles or RSUs without possessing the corresponding secret keys.
- **Unlinkability:** The use of dynamically changing pseudonyms ensures that an adversary cannot link multiple messages to the same vehicle, thereby preventing tracking based on communication patterns.
- **Conditional Anonymity:** While preserving anonymity during normal operation, the scheme supports traceability by enabling the LTA to reveal the true identity of a vehicle in the case of disputes or malicious behavior.
- **Resistance to Impersonation Attacks:** The scheme prevents adversaries from masquerading as legitimate vehicles or RSUs by enforcing strong mutual authentication and secure credential verification.

### 3.3. Assumptions

The proposed scheme is based on the following assumptions:

- **Hierarchical Resource Model:** The Trusted Authority (TA) possesses higher computational, communication, and storage capabilities than Local Trusted Authorities (LTAs). Similarly, LTAs are more resource-rich than Road Side Units (RSUs), and RSUs have greater capacity than vehicle On-Board Units (OBUs).
- **Pseudonym Management:** Each LTA maintains a separate Merkle Hash Tree (MHT) for every registered vehicle, representing the set of pseudonyms assigned to that vehicle within its jurisdiction.
- **Time Synchronization:** The clocks of the TA, LTAs, RSUs, and vehicles are assumed to be loosely synchronized, for example, using GPS-based time synchronization.

## 4. LEPA Scheme

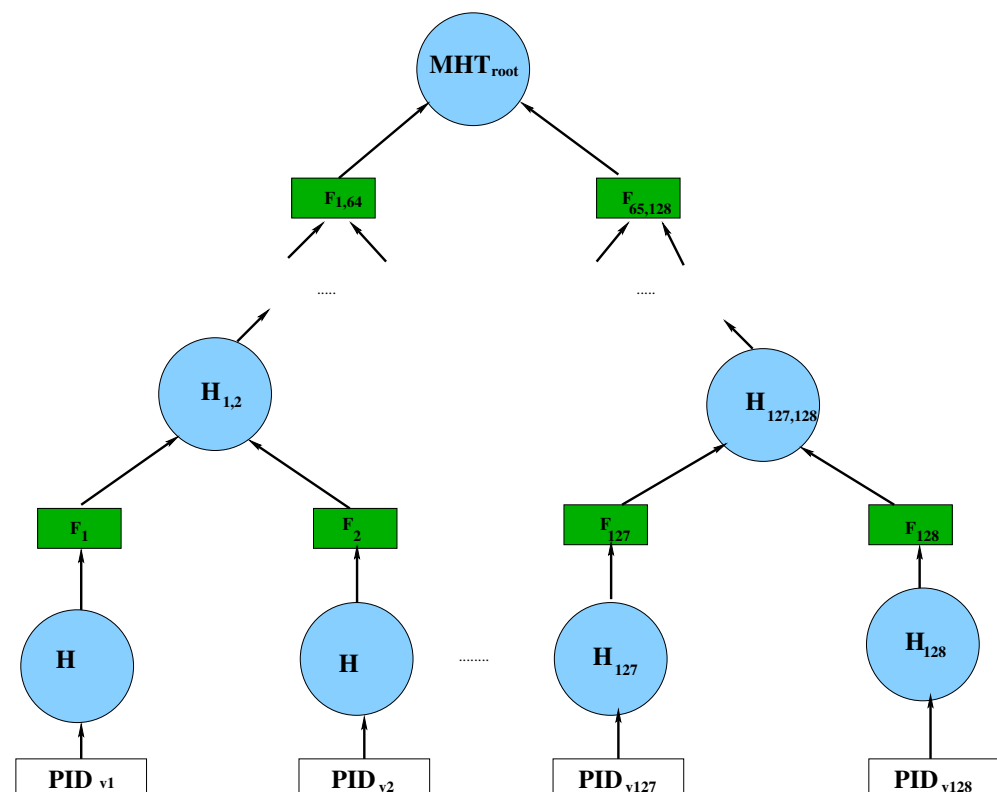
To participate in VANET communications, each vehicle must register with its home LTA. During the registration phase, the vehicle's OBU is provisioned with a pool of pseudonyms, enabling secure and privacy-preserving communication. Frequent pseudonym updates ensure unlinkability and protect vehicles from tracking.

Each LTA operates under the authority of the TA and functions as a regional administrative entity. It is responsible for registering vehicles and RSUs within its domain and generating corresponding public/private key pairs for secure communication. Communication between the LTA and RSUs is direct, as illustrated in Figure 3.

The proposed LEPA scheme leverages Merkle Hash Trees (MHTs) to efficiently manage pseudonym sets associated with vehicles. In addition, Cuckoo Filters (CFs) are employed to enable fast and lightweight membership verification, thereby reducing authentication and revocation overhead. Cuckoo Filters are also utilized to streamline the authentication of RSUs, further enhancing the efficiency of the overall system.

#### 4.1. Construction of the Merkle Hash Tree (MHT)

In our LEPA scheme, each vehicle constructs an MHT of its set of pseudonyms. Figure 4 shows a sample MHT of 128 pseudonyms assigned to a vehicle. In this MHT, every leaf node stores a pseudonym, and all non-leaf nodes are associated with a fingerprint of cryptographic hash that is formed from the last thirteen bits of the hash of the child node. We use the method in [5] to retrieve the Missing Fingerprint Values (MFVs) associated with a pseudonym of the vehicle  $PID_V$  to recalculate the  $MHT_{root}$  for verification. Table 2 presents the MFVs corresponding to each pseudonym of the vehicle in Figure 4. Each LTA also maintains a separate MHT for each vehicle's set of pseudonyms in its region. LTA puts the expiry time  $Exp_{MHT_{root}}$  for the set of pseudonyms allocated to a vehicle in its region and the corresponding  $MHT_{root}$  in a priority queue.



**Figure 4.** A sample Merkle Hash Tree of pseudonyms of a vehicle. Reprinted with permission from [3]. Copyright Year:2022; Copyright Owner's Name: IEEE.

**Table 2.** Authentication Information.

| $PID_i$     | MFVs                                                                              |
|-------------|-----------------------------------------------------------------------------------|
| $PID_1$     | $F_2, F_{3,4}, F_{5,8}, F_{9,16}, F_{17,32}, F_{33,64}, F_{65,128}$               |
| $PID_2$     | $F_1, F_{3,4}, F_{5,8}, F_{9,16}, F_{17,32}, F_{33,64}, F_{65,128}$               |
| $PID_3$     | $F_4, F_{1,2}, F_{5,8}, F_{9,16}, F_{17,32}, F_{33,64}, F_{65,128}$               |
| $PID_4$     | $F_3, F_{1,2}, F_{5,8}, F_{9,16}, F_{17,32}, F_{33,64}, F_{65,128}$               |
| .....       | .....                                                                             |
| $PID_{64}$  | $F_{63}, F_{61,62}, F_{57,60}, F_{49,56}, F_{33,48}, F_{1,32}, F_{65,128}$        |
| .....       | .....                                                                             |
| $PID_{128}$ | $F_{127}, F_{123,126}, F_{121,124}, F_{113,120}, F_{97,112}, F_{65,96}, F_{1,64}$ |

#### 4.2. Construction of Cuckoo Filters (CFs)

Each LTA maintains two separate Cuckoo Filters (CFs): one for the vehicles registered under its jurisdiction and another for the RSUs within its region. The positive CF stores entries for valid entities, while the negative CF contains entries for malicious or revoked

entities. This dual use of positive and negative CFs enables efficient authentication of both vehicles and RSUs during V2I communication.

#### 4.2.1. Construction of CFs for Vehicles

Each LTA maintains two Cuckoo Filter structures, namely a Positive Cuckoo Filter ( $PCF_V$ ) and a Negative Cuckoo Filter ( $NCF_V$ ), to enable efficient authentication and revocation management of vehicles within its jurisdiction. The  $PCF_V$  stores the membership information of legitimate vehicles, whereas the  $NCF_V$  maintains entries corresponding to revoked or malicious vehicles. During the registration phase, each vehicle is assigned a bucket index  $B_{ndx}$  associated with the root value of its pseudonym set in the  $PCF_V$ . This index is securely communicated to the vehicle and is used during subsequent authentication processes. The insert, lookup, and delete operations on  $PCF_V$  and  $NCF_V$  are performed under the following conditions.

**When a vehicle registers with its home LTA:** Upon registration, each vehicle authenticates itself to its home LTA using its real identity. The LTA provisions the vehicle's On-Board Unit (OBU) with a set of pseudonyms and corresponding credentials. These pseudonyms act as temporary identifiers that enable anonymous and unlinkable communication. To mitigate tracking and correlation attacks, the vehicle periodically updates its pseudonym using a secure pseudonym-changing mechanism. Vehicles are not confined to their home LTA's region and may freely traverse across regions managed by different LTAs. When a vehicle enters a new region, it performs a registration (handover) procedure with the local LTA via a nearby RSU within its communication range. Upon successful authentication, the new LTA issues a fresh set of pseudonyms, thereby ensuring continuous privacy protection and preventing cross-region tracking. LTAs are assumed to form a trusted network and can securely communicate with each other to support coordinated authentication and pseudonym management. For each registered vehicle, the LTA constructs a Merkle Hash Tree (MHT) over the vehicle's pseudonym set and computes the corresponding root value. The LTA then derives a fingerprint of this root value and inserts it into the Positive Cuckoo Filter ( $PCF_V$ ) to represent the vehicle as a valid member in the system.

**When the  $Exp_{MHT_{root}}$  expires:** The LTA puts the expiry time  $Exp_{MHT_{root}}$  for the set of pseudonyms allocated to a vehicle in its region and the corresponding  $MHT_{root}$  in a priority queue. The priority queue is sorted by the  $Exp_{MHT_{root}}$ . The LTA periodically checks the priority queue to determine if there is an expired set of pseudonyms associated with a vehicle's  $MHT_{root}$  in its region. It removes the expired  $MHT_{root}$  from the priority queue. The LTA then searches for  $F(MHT_{root})$  and deletes it from both  $PCF_V$  and  $NCF_V$ .

**When an LTA finds a malicious vehicle in its region:** An LTA receives a report when a misbehaving vehicle is found in its region. In the report for the misbehaving vehicle, the sender's pseudo ID ( $PID_{V_{sender}}$ ) and the malicious vehicle's pseudo ID ( $PID_{V_{malicious}}$ ) are included. Upon receiving the misbehaving report, the LTA first verifies the validity of  $PID_{V_{sender}}$ . As soon as the LTA determines a vehicle is malicious using an underlying method (malicious node detection has been extensively studied in the literature; any of those methods can be used for determining malicious nodes), it deletes corresponding  $F(MHT_{root})$  from the  $PCF_V$ . After that, LTA inserts the  $F(MHT_{root})$  of the malicious vehicle into the  $NCF_V$ .

**When a vehicle leaves an LTA's region:** Before leaving an LTA's region, a vehicle sends a message that includes its current pseudo ID  $PID_v$ ,  $MHT_{root}$ , and  $B_{ndx}$  for  $F(MHT_{root})$  in  $PCF_V$ . In response to the vehicle's message, the LTA verifies the credentials of the vehicle and deletes the vehicle's associated  $F(MHT_{root})$  from  $PCF_V$ .

**When a vehicle enters a new LTA's region:** When a vehicle enters a region covered by a new LTA, it requests a new set of pseudonyms from the LTA through its local RSU. The request message contains the current pseudo ID of the vehicle  $PID_v$ , the MHT root of

its pseudonyms  $MHT_{root}$ , and the ID of its previous LTA. The new LTA communicates with the previous LTA through a secure connection channel to verify the vehicle's credentials. The previous LTA can look up its log table and Cuckoo Filters for pseudonym verification. The new LTA generates a new set of pseudonyms for the vehicle after successful verification. Next, the LTA constructs an MHT of pseudonyms associated with the vehicle and inserts the fingerprint of the MHT root  $F(MHT_{root})$  in the  $PCF_V$ .

#### 4.2.2. Construction of CFs for RSUs

In our scheme, LTA maintains a Positive Cuckoo Filter  $PCF_{RSU}$  of all valid RSUs registered with it. When an RSU is compromised, LTA deletes the fingerprint of the public key of the RSU from  $PCF_{RSU}$ . After that, the LTA inserts the fingerprint of the public key of the RSU into the  $NCF_{RSU}$ .

#### 4.3. V2I Authentication Phase

In our LEPA scheme, the LTA stores the fingerprint of the MHT root  $F(MHT_{root})$  corresponding to every valid vehicle in its region in a  $PCF_V$ . Similarly, the LTA constructs a  $NCF_V$ , which stores the fingerprint of the MHT root  $F(MHT_{root})$  for revoked or malicious vehicles in its region. LTA also initializes a  $PCF_{RSU}$  of valid RSUs and an  $NCF_{RSU}$  of compromised RSU in its region for efficient authentication. LTA signs both the positive and negative CFs  $Sig_{LTA}(PCF_{RSU}, NCF_{RSU})$  and  $Sig_{LTA}(PCF_V, NCF_V)$  and periodically broadcasts them. The latest CFs are used to authenticate both vehicles and RSUs in V2I communications.

Each RSU periodically broadcasts beacon messages that includes its ID  $ID_{RSU_i}$ , public key  $PU_{RSU_i}$ , and message generation timestamp  $t_s$ . Upon receiving the beacon message, a vehicle  $V_i$  first checks  $t_s$  for freshness. After that,  $V_i$  computes the fingerprint of  $PU_{RSU_i}$  and looks into both the  $PCF_{RSU}$  and  $NCF_{RSU}$ . If the  $V_i$  finds the fingerprint of  $PU_{RSU_i}$  in  $PCF_{RSU}$  and not in  $NCF_{RSU}$ , the RSU is then considered valid. Algorithm 1 presents how a vehicle authenticates an RSU using both  $PCF_{RSU}$  and  $NCF_{RSU}$ .

---

**Algorithm 1:** Authentication of  $PU_{RSU_i}$  by  $V_i$ . Reprinted with permission from [3]. Copyright Year:2022; Copyright Owner's Name: IEEE

---

```

1 Check the freshness of the received message using  $t_s$ ;
2 if message is fresh then
3    $\zeta_{RSU_i} = \text{fingerprint}(PU_{RSU_i})$ ;
4   if  $\zeta_{RSU_i} \in PCF_{RSU}$  then
5     if  $\zeta_{RSU_i} \notin NCF_{RSU}$  then
6        $PU_{RSU_i}$  is considered valid;
7     else
8        $PU_{RSU_i}$  sends message to LTA for verification ;
9     end if
10  end if
11  if  $\zeta_{RSU_i} \notin PCF_{RSU}$  then
12    if  $\zeta_{RSU_i} \in NCF_{RSU}$  then
13       $PU_{RSU_i}$  is considered malicious;
14    else
15       $V_i$  waits for updated CFs from the LTA;
16    end if
17  end if
18 else
19   Drops the beacon message;
20 end if

```

---

$V_i$  sends a message to the  $RSU_i$  after successfully authenticating the  $PU_{RSU_i}$  for mutual authentication. The message contains the pseudo-ID of sender vehicle  $PID_{V_i}$ , missing fingerprint values  $MFVs$  to calculate the root value corresponding to  $V_i$ 's set of pseudonyms, index of the bucket  $B_{ndx}$  in positive Cuckoo Filter  $PCF_V$ , ID of the  $RSU_i$   $ID_{RSU_i}$ , and message generation timestamp  $t_s$ .  $V_i$  encrypts the message using the public key  $PU_{RSU_i}$  of the  $RSU_i$ . In order to read a message sent by  $V_i$ ,  $RSU_i$  first decrypts it with its private key,  $PR_{RSU_i}$ . Then, it checks the  $t_s$  for the freshness of the message. If the  $t_s$  is valid, the  $RSU_i$  recalculates the MHT root value corresponding to the  $V_i$  using the received  $MFVs$  and  $PID_{V_i}$ . After that,  $RSU_i$  checks if the  $F(MHT_{root})$  exists in the  $PCF_V$  with bucket index  $B_{ndx}$ .  $RSU_i$  also checks negative  $NCF_V$  for the  $F(MHT_{root})$ . If  $RSU_i$  finds the  $F(MHT_{root})$  in  $PCF_V$  with  $B_{ndx}$  and not in  $NCF_V$ , the vehicle  $V_i$  is considered as valid. A detailed description of the authentication process is presented in Algorithm 2.

---

**Algorithm 2:**  $RSU_i$  Authenticates  $V_i$ 


---

```

1 if The message is Fresh then
2   Calculate the  $MHT_{root}$  value using  $MFVs$  and  $PID_{V_i}$ ;
3    $\zeta_{MHT_{root}} = fingerprint(MHT_{root})$ ;
4   Look up into the  $PCF_V$  with  $B_{ndx}$  and  $NCF_V$  for  $\zeta_{MHT_{root}}$  corresponding to  $V_i$ ;
5   if  $\zeta_{MHT_{root}} \in PCF_V$  with  $B_{ndx}$  then
6     if  $\zeta_{MHT_{root}} \notin NCF_V$  then
7        $V_i$  is considered valid;
8     else
9        $RSU_i$  sends message to LTA for verification;
10    end if
11  end if
12  if  $\zeta_{MHT_{root}} \notin PCF_V$  with  $B_{ndx}$  then
13    if  $\zeta_{MHT_{root}} \in NCF_V$  then
14       $V_i$  is considered malicious;
15    else
16       $RSU_i$  waits for updated CFs from LTA;
17    end if
18  end if
19 else
20   Drops the received message;
21 end if

```

---

Table 3 presents the four possible outcomes of query results from the positive and negative Cuckoo Filters (CFs). Based on these results, the receiver determines whether the sender is valid or not using the results of both positive and negative Cuckoo Filters in Table 3.

The query result in Case 1 confirms the sender's validity, while Case 2 identifies the sender as malicious. In such instances, the receiver reports the malicious sender to the LTA. Case 3 indicates that the LTA has not yet updated the Cuckoo CFs. If this occurs, the receiver waits for the updated CFs from the LTA. Case 4 arises due to a false positive from the CFs. When this happens, the receiver forwards the received message to its LTA for further verification. Upon receiving the message, the LTA checks its previous CFs and log tables to verify the sender's authenticity. If the sender is found to be legitimate, the LTA removes the  $F(MHT_{root})$  corresponding to  $V_i$  or  $F(PU_{RSU_i})$  from the negative CF. Conversely, if the sender is determined to be illegitimate, the LTA removes the corresponding fingerprint

from the positive CF. The LTA then broadcasts the updated positive and negative CFs to all entities within its region, ensuring the system's authentication integrity is maintained.

**Table 3.** Possible Cases and Results with Cuckoo Filter. Reprinted with permission from [3]. Copyright Year: 2022; Copyright Owner's Name: IEEE.

| Case | Positive CF | Negative CF | Conclusion                                   |
|------|-------------|-------------|----------------------------------------------|
| 1    | True        | False       | Valid                                        |
| 2    | False       | True        | Malicious                                    |
| 3    | False       | False       | Filters have not been updated                |
| 4    | True        | True        | Forwards to the LTA for further verification |

## 5. Performance Analysis and Comparison

In this section, we first conduct a detailed analysis of the security properties of our proposed LEPA scheme. We systematically evaluate its robustness against potential vulnerabilities to ensure its reliability. Subsequently, we verify the security of the LEPA scheme against common attack vectors, including replay attacks, man-in-the-middle attacks, and impersonation attacks. To achieve this, we utilize two well-established tools: SPAN (Security Protocol ANimator) [6] and AVISPA (Automated Validation of Internet Security Protocols and Applications) [7], a formal tool for the automated validation of cryptographic protocols. These tools enable rigorous testing and validation of our scheme's resistance to these threats.

Following the security analysis, we perform a comparative evaluation of the authentication overheads of the LEPA scheme with those of three notable existing schemes: LIAP [8], NERA [9], and HDMA [10]. This comparison highlights the efficiency of our proposed approach, demonstrating its advantages in terms of reduced computational and communication costs.

### 5.1. Security Analysis

In this section, we analyze the security of our LEPA scheme.

#### 5.1.1. Mutual Authentication

The LEPA scheme leverages both positive and negative CFs  $< (PCF_V, NCF_V), (PCF_{RSU}, NCF_{RSU}) >$  to authenticate vehicles and RSUs. LTA signs both the positive CFs and negative CFs  $Sig_{LTA(PCF_{RSU}, NCF_{RSU})}$  and  $Sig_{LTA(PCF_V, NCF_V)}$ , and periodically broadcasts them. Vehicles and RSUs authenticate each other using the results of both the latest positive and negative CFs stated in Table 3. Therefore, VANET communication is more secure due to V2I mutual authentication.

#### 5.1.2. Identity Anonymity

When a vehicle registers with its home LTA using its real identity, the vehicle's OBU is loaded with a set of pseudonyms. In our LEPA scheme, vehicles use pseudonyms for communication. The real identity of a vehicle is never used in communication. Only the home LTA knows a vehicle's real identity. The real identity of a vehicle remains secure even if all the RSUs are compromised.

#### 5.1.3. Resistance to Message Modification Attack

Under the LEPA scheme, the RSUs periodically broadcast beacon messages containing its  $ID_{RSU}$ ,  $PU_{RSU}$ , and  $t_s$ . Vehicles authenticate the  $PU_{RSU}$  received in the beacon message. Then, vehicles encrypt messages using the  $PU_{RSU}$  for mutual authentication. Since the attackers do not know the  $PR_{RSU}$ , only the intended RSUs can read/modify the messages.

#### 5.1.4. Resistance to Replay Attack

In the LEPA scheme, both vehicles and RSUs append message generation timestamps  $t_s$  with the messages. Upon receiving the messages, the receivers first check the  $t_s$  to resist replay attack. The clocks of RSUs and vehicles are assumed to be loosely synchronized (this can be done using GPS).

#### 5.2. Verification Using SPAN and AVISPA Tools

SPAN (Security Protocol ANimator) [6] and AVISPA (Automated Validation of Internet Security Protocols and Applications) [7] tools are widely used in literature [11–14] to verify the security of protocols. We also verify the security of the LEPA protocol using SPAN and AVISPA tools.

Our model consists of three basic roles : *rsu*, *localTA*, and *vehicle*, and are denoted by  $a$ ,  $b$ , and  $c$ , respectively. In our model, we use  $ka$ ,  $kb$ , and  $kc$  to represent the public keys of the *rsu*, *localTA*, and *vehicle* respectively, and  $h$  represents the cryptographic hash function. Here, *localTA* first activates the start signal. *vehicle* gets a set of pseudonyms and the bucket index *nonce-5* in the positive Cuckoo Filter ( $PCF_v$ ) from the *localTA*. *vehicle* sends its current pseudonym *nonce-1*, Missing Fingerprint Values, and bucket index *nonce-5* for V2I authentication. Figure 5 presents the message sequence chart of our proposed scheme generated by SPAN and AVISPA tools.

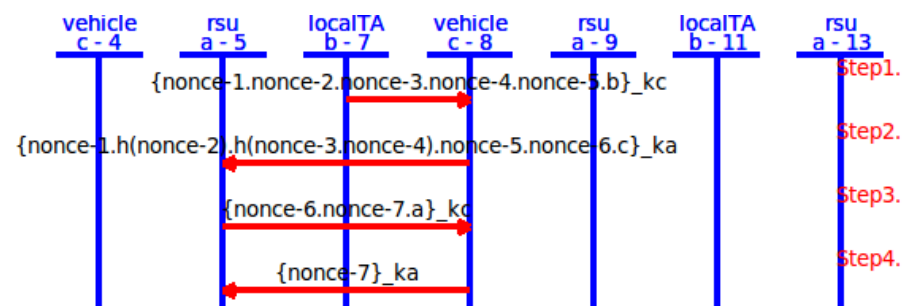


Figure 5. Message sequence chart of our scheme using SPAN and AVISPA tools.

Figure 6 shows the message sequence chart in the presence of an intruder. It is evident from the sequence chart that the intruder cannot read or modify the messages. The intruder only listens and forwards the messages. We describe the sequence of exchanged messages in the presence of an intruder  $i$  as follows:

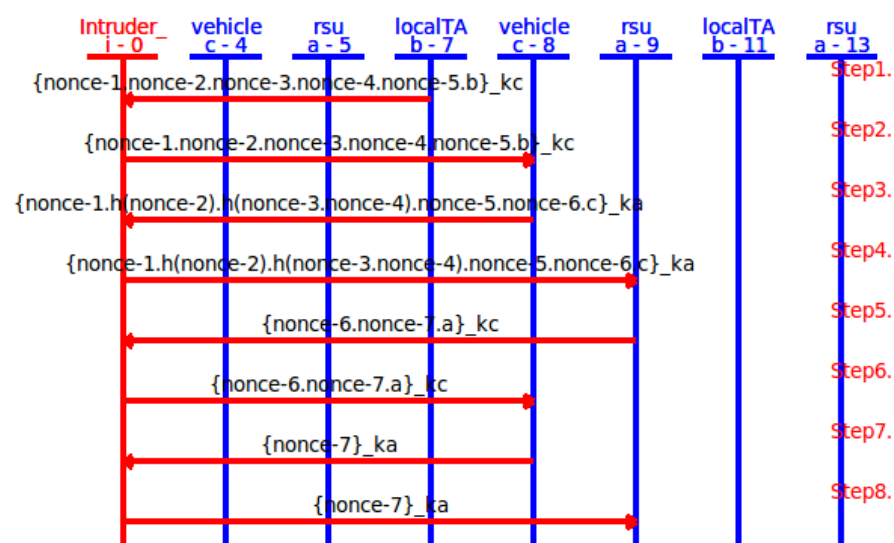


Figure 6. Message sequence chart of our scheme in the presence of an intruder.

- Step 1:* The *local TA* initiates the session and sends a message containing the *vehicle's* set of pseudonyms and the bucket index *nonce-5* in  $PCF_v$  to the *vehicle*.
- Step 2:* Since the intruder does not know the private key of the *vehicle*, the intruder only listens to the message.
- Step 3:* The *vehicle* sends the received *nonce-5*, its current pseudonym *nonce-1*, and *MFVs* along with a nonce to the *rsu*.
- Step 4:* The intruder is unable to read and/or modify the message as it is encrypted using public key *ka* of *rsu*. The intruder only views the message and passes it to the *rsu*.
- Step 5:* *rsu* decrypts the received message using its private key  $inv(ka)$ . Next, it recalculates the  $MHT_{root}$  value using the received *MFVs*. After that, it looks into  $PCF_v$  with *nonce-5* and also searches  $NCF_v$  for  $MHT_{root}$ . *rsu* sends a new nonce to the *vehicle*, along with the received nonce after verification.
- Step 6:* The intruder only listens to the message and passes it to the *vehicle*.
- Step 7:* The *vehicle* retrieves the nonces by using its private key *kc* and sends back the received nonce to the *rsu*.
- Step 8:* The intruder is only able to view the message but unable to read or modify it. He/she passes the message to the *rsu*.

### 5.3. Proof of Correctness Using BAN Logic

Borrows, Abadi, and Needham (BAN) logic [15] has been extensively used as a formal framework for verifying the correctness and security of authentication protocols. Its structured approach to reasoning about trust and message exchanges in cryptographic systems makes it a powerful tool for analyzing protocol behavior and validating security properties. In this subsection, we perform the formal verification of our LEPA scheme using both the traditional BAN logic and the PKI-based extended BAN logic [16]. The extended BAN logic incorporates support for Public Key Infrastructure (PKI), enabling a more comprehensive analysis of protocols that rely on public key cryptography.

We begin by providing a concise overview of the BAN logic, including its core principles, key notations, and inference rules. This introduction serves as a foundation for understanding the formal verification process. Next, we present an idealized representation of the LEPA scheme, which abstracts the protocol's message exchanges into a formal model suitable for logical reasoning. Following this, we define the initial assumptions underlying the LEPA scheme, such as trust relationships and shared knowledge between entities. We also outline the security goals that the protocol is designed to achieve, including mutual authentication, message integrity, and resistance to attacks such as replay and impersonation. Finally, we apply the inference rules of BAN logic and its PKI-based extension to derive conclusions about the LEPA scheme's ability to achieve the defined goals. Through logical derivation, we demonstrate how the LEPA scheme satisfies these objectives, ensuring its robustness and reliability in secure communications.

We use the following BAN logic inference rules (see Table 4 for BAN logic notation) to perform the formal verification of our LEPA scheme.

**Table 4.** BAN logic notation.

| Notation                           | Description                                |
|------------------------------------|--------------------------------------------|
| $P \models X$                      | P believes X                               |
| $P \triangleleft X$                | P sees X                                   |
| $P \mid \sim X$                    | P once said X                              |
| $P \Rightarrow X$                  | P controls X                               |
| $\#(X)$                            | X is fresh                                 |
| $\{X\}_k$                          | X is encrypted under the key k             |
| $(X)_k$                            | X is hashed with the key k                 |
| $\kappa(P, K_P)$                   | P has public key $K_P$                     |
| $\Pi(K_P^{-1})$                    | P has private key $K_P^{-1}$               |
| $\sigma(X, K_P^{-1})$              | X signed with private key $K_P^{-1}$       |
| $P \rightarrow Q : \Re(X, (X, Q))$ | P sends X to the intended receiver Q       |
| $\sigma(\Re(X, Q), K_P^{-1})$      | X signed with private key $K_P^{-1}$ for Q |

### 5.3.1. Inference Rules for BAN Logic

We use the following BAN logic rules to perform the formal verification of our LEPA scheme:

- *R1: Message meaning rule*

$$\frac{P \models \kappa(Q, K_Q), P \models \Pi(K_Q^{-1}), P \triangleleft \sigma(X, K_Q^{-1})}{P \models Q \mid \sim X} \quad (1)$$

In this context, P and Q are communicating entities, and Q possesses its (public,private) key pair  $(K_Q, K_Q^{-1})$ . The notation  $\sigma(X, K_Q^{-1})$  signifies that Q has signed the message X with its private key  $K_Q^{-1}$ , ensuring non-repudiation. This signature serves as evidence to verify the authenticity of the sender, which, in this case, is Q.

Now, if P trusts the validity of the Q's public key  $K_Q$  and private key  $K_Q^{-1}$  of Q, and P observes X is signed with Q's private key  $K_Q^{-1}$ , then P can conclude that X was indeed generated by Q.

$$\frac{P \models \kappa(Q, K_Q), P \models \Pi(K_Q^{-1}), P \triangleleft \sigma(\Re(X, P), K_Q^{-1})}{P \models Q \mid \sim X} \quad (2)$$

Similarly, the notation  $\sigma(\Re(X, P), K_Q^{-1})$  indicates that the message X has been signed with Q's private key  $K_Q^{-1}$  and is intended for P ensuring the non-repudiation.

In this scenario, if P trusts that Q possesses a valid public key  $K_Q$  and private key  $K_Q^{-1}$ , and P observes X is signed with Q's private key  $K_Q^{-1}$  for which P is the intended recipient, then P believes that Q previously stated X.

- *R2: Nonce Verification rule*

$$\frac{P \models \#(X), P \models Q \mid \sim X}{P \models Q \models X} \quad (3)$$

The above rule deals with the freshness of the message. If P believes that X is fresh, and Q said X once, then P believes Q believes X is fresh.

- *R3: Jurisdiction rule*

$$\frac{P \models Q \Rightarrow X, P \models Q \models X}{P \models X} \quad (4)$$

If P believes that Q controls X and P believes that Q believes X, then P believes X.

- *R4: Freshness rule*

$$\frac{P \models \#(X)}{P \models \#(X, Y)} \quad (5)$$

Here,  $X$  and  $Y$  are part of the same message and  $\#(X)$  denotes freshness of  $X$ . The attackers replay the same messages several times to create confusion in a replay attack. The freshness of a message can resist the replay attack.

The above rule states that if part of a message is fresh, then the entire message is fresh. The receiver  $P$  can check the freshness of the message to defend replay attack.

- *R5: Sees rule*

$$\frac{P \triangleleft (X, Y)}{P \triangleleft (X)} \quad (6)$$

Here,  $X$  and  $Y$  are part of the same message. If  $P$  sees the message, it can also see the individual part.

$$\frac{P \triangleleft \sigma(\mathcal{R}(X, all), K_Q^{-1})}{P \triangleleft \sigma(\mathcal{R}(X, P), K_Q^{-1})} \quad (7)$$

In this scenario,  $P$  trusts that  $Q$  possesses a valid public key  $K_Q$  and private key  $K_Q^{-1}$ . If  $P$  observes  $X$  is signed with  $Q$ 's private key  $K_Q^{-1}$  and broadcasted to all the communicating entities, then  $P$  can also see the message  $X$ .

$$\frac{P \models \kappa(Q, K_Q), P \models \Pi(K_Q^{-1}), P \triangleleft \{X\}_{K_Q^{-1}}}{P \triangleleft X} \quad (8)$$

Similarly, If  $P$  sees a message  $X$  encrypted with the private key  $K_Q^{-1}$  of  $Q$ , then  $P$  can decrypt and see the message contents using the public key  $K_Q$  of  $Q$ .

$$\frac{P \models \kappa(Q, K_Q), P \models \Pi(K_Q^{-1}), P \triangleleft \sigma(X, K_Q^{-1})}{P \triangleleft X} \quad (9)$$

The above rule states that If  $P$  sees a message  $X$  signed with the private Key  $K_Q^{-1}$  of  $Q$ , then  $P$  can extract the message contents using the public key  $K_Q$  of  $Q$ .

$$\frac{P \models \kappa(P, K_P), P \models \Pi(K_P^{-1}), P \triangleleft \{X\}_{K_P^{-1}}}{P \triangleleft X} \quad (10)$$

This means that if  $P$  sees a message, then  $P$  can see the components of the message as long as it knows the required key.

### 5.3.2. Protocol Idealization

The protocol messages are idealized as follows:

- $LTA \rightarrow V_i$   
M1:  $\langle \sigma(\mathcal{R}(PCF_{RSU}, NCF_{RSU}, t_{LTA}), all), K_{LTA}^{-1} \rangle$
- $RSU_i \rightarrow V_i$   
M2:  $\langle (ID_{RSU_i}, K_{RSU_i}, t_{RSU_i}), all \rangle$
- $LTA \rightarrow RSU_i$   
M3:  $\langle \sigma(\mathcal{R}(PCF_V, NCF_V, t_{LTA}), all), K_{LTA}^{-1} \rangle$
- $V_i \rightarrow RSU_i$   
M4:  $\langle (PID_{V_i}, MFVs, B_{ndx}, K_{V_i}, ID_{RSU_i}, t_{V_i}), K_{RSU_i}^{-1} \rangle$

### 5.3.3. Initial Assumptions and Goal of the Protocol

In the LEPA scheme, the TA distributes its public key ( $K_{TA}$ ) to all LTAs, RSUs, and vehicles, and the private key ( $K_{TA}^{-1}$ ) is kept secret. Each LTA is responsible for generating the public and private key pairs for each RSU ( $K_{RSU}, K_{RSU}^{-1}$ ) and vehicle ( $K_V, K_V^{-1}$ ) registered under it. LTA also maintains positive and negative Cuckoo Filters for both vehicles ( $PCF_V, NCF_V$ ) and RSUs ( $PCF_{RSU}, NCF_{RSU}$ ) to facilitate the V2I authentication process. Here is an overview of the protocol's initial assumptions:

- A1:  $LTA \models (\kappa(TA, K_{TA}), \Pi(K_{TA}^{-1}))$
- A2:  $RSU_i \models (\kappa(TA, K_{TA}), \Pi(K_{TA}^{-1}))$
- A3:  $V_i \models (\kappa(TA, K_{TA}), \Pi(K_{TA}^{-1}))$
- A4:  $RSU_i \models (\kappa(LTA, K_{TA}), \Pi(K_{LTA}^{-1}))$
- A5:  $V_i \models (\kappa(LTA, K_{LTA}), \Pi(K_{LTA}^{-1}))$
- A6:  $RSU_i \models LTA \Rightarrow (PCF_V, NCF_V)$
- A7:  $V_i \models LTA \Rightarrow (PCF_{RSU}, NCF_{RSU})$
- A8:  $RSU_i \models \#(t_{LTA})$
- A9:  $V_i \models \#(t_{LTA})$
- A10:  $RSU_i \models \#(t_{V_i})$
- A11:  $V_i \models \#(t_{RSU_i})$
- A12:  $RSU_i \models LTA \Rightarrow B_{ndx}$

Our protocol aims to establish secure communication through mutual authentication between RSUs and Vehicles. The main goals of our protocol are as follows:

- G1:  $V_i \models LTA \models (PCF_{RSU}, NCF_{RSU})$
- G2:  $V_i \models PU_{RSU_i}$
- G3:  $RSU_i \models LTA \models (PCF_V, NCF_V)$
- G2:  $RSU_i \models PID_{V_i}$

### 5.3.4. Logic Derivation

On the basis of logic postulates and initial assumptions, we provide a formal proof of our proposed LEPA scheme in this subsection.

G1 can be deduced from M1 as follows:

- D1:  $V_i \triangleleft \sigma((PCF_{RSU}, NCF_{RSU}, t_{LTA}), K_{LTA}^{-1})$  (From A5 and R5 (7))
- D2:  $V_i \models LTA \sim (PCF_{RSU}, NCF_{RSU})$  (From A5 and R1 (1))
- D3:  $V_i \models \#(PCF_{RSU}, NCF_{RSU})$  (From A9 and R4 (5))
- D4:  $V_i \models LTA \models (PCF_{RSU}, NCF_{RSU})$  (From D2, D3, and R2 (3)) ..... (G1)

G2 can be deduced from M2 as follows:

- D5:  $V_i \triangleleft (ID_{RSU_i}, K_{RSU_i}, t_{RSU_i})$  (From R5 (7))
- D6:  $V_i \models \#K_{RSU_i}$  (From A11 and R4 (5))
- D7:  $V_i \models (PCF_{RSU}, NCF_{RSU})$  (From A7, D4, and R3 (4))
- D8:  $V_i \models K_{RSU_i}$  (From D6, D7, and R3 (4)) ..... (G2)

From M3 we deduce the following:

- D9:  $RSU_i \triangleleft \sigma((PCF_V, NCF_V, t_{LTA}), K_{LTA}^{-1})$  (From A4 and R5 (7))
- D10:  $RSU_i \models LTA \sim (PCF_V, NCF_V)$  (From A4 and R1 (1))
- D11:  $RSU_i \models \#(PCF_V, NCF_V)$  (From A8 and R4 (5))
- D12:  $RSU_i \models LTA \models (PCF_V, NCF_V)$  (From D10, D11, and R2 (3)) ..... (G3)

G4 can be deduced from M4 as follows:

- D13:  $RSU_i \triangleleft (PID_{V_i}, MFVs, B_{ndx}, K_{V_i}, t_{V_i})$  (From R5 (7))
- D14:  $RSU_i \models \#(PID_{V_i}, MFVs, B_{ndx}, K_{V_i})$  (From A10 and R4 (5))
- D15:  $RSU_i \models (PCF_V, NCF_V)$  (From A6, D12, and R3 (4))

- D16:  $RSU_i | \equiv PID_{V_i}$  (From A12, D14, D15, and R3 (4)) ..... (G4)

The above analysis shows our protocol satisfies all the goals (G1–G4) and allows both vehicles and RSUs to authenticate each other in VANET communication.

#### 5.4. V2I Authentication Overheads Analysis

In the LEPA scheme, the RSUs periodically broadcast beacon messages containing its ID  $ID_{RSU}$ , public key  $PU_{RSU}$ , and message generation timestamp  $t_s$ . When a vehicle enters an RSU's region, it checks the  $PCF_{RSU}$  and  $NCF_{RSU}$  to verify the  $PU_{RSU}$  received in RSU's beacon message. After successful verification, the vehicle sends a message to the RSU to authenticate itself. This message contains the pseudo ID of the vehicle  $PID_V$ , the Missing fingerprint of MHT for the corresponding pseudo ID  $MFVs$ , and the bucket index in the positive Cuckoo Filter for vehicles  $B_{ndx}$ . When the RSU receives the vehicle's message, it first recalculates the  $MHT_{root}$  using the  $MFVs$ . Next, the RSU looks up into the  $PCF_V$  with  $B_{ndx}$  and  $NCF_V$ .

In pseudonym-based scheme [17], TA/CA stores all the unexpired pseudonyms of revoked or malicious vehicles in the CRL. A CRL generally consists of a header, a current date, a date of the last update, the date of the next update, and a list of revoked certificates signed by the certificate issuer [17]. Under US-based SAE J273 standard [18], a vehicle requires 262,800 pseudonyms a year if it changes its pseudonym every 120 s or after a 1 km distance. The CRL is expected to contain a fixed length header of 51 bytes and a variable length of pseudonyms, which is 16 bytes per pseudonym. The size of CRL increases significantly with the increased number of revoked vehicles and the CRL is broadcasted periodically in VANET. Therefore, it introduces additional overhead for authentication. Further, when RSUs try to download CRLs synchronously, congestion and overload may occur in the network. Our LEPA scheme leverages CFs for efficient authentication of both vehicles and RSUs. On the other hand, CA revokes the certificate of malicious vehicles and RSUs and inserts them into the Certificate Revocation List of Vehicles (VCRLs) and RSUs (RCRLs) in LIAP [8]. Both Vehicles and RSUs check RCRLs and VCRLs to authenticate each other. In NERA [9], RSUs authenticate vehicles using the CRL containing the real ID of the vehicle. In the HDMA scheme [10], an RSU checks if a pseudonym associated with a vehicle is included in the CRL for V2I authentication. A Cuckoo Filter with  $m = 1$  M buckets of  $n = 4$  cells per bucket and fingerprints of  $f = 13$  bits can store the fingerprints of the MHT root value of 4 M revoked vehicles. In this case, the size of the Cuckoo Filter is approximately 50 MB, whereas, to store the thousands of unexpired pseudonyms of 4 M of revoked vehicles (only 2 K pseudonyms per vehicle), the size of CRL is approximately 525 GB. Both LIAP [8] and NERA [9] use bilinear pairing and Map-To-Point operations, which are costly.

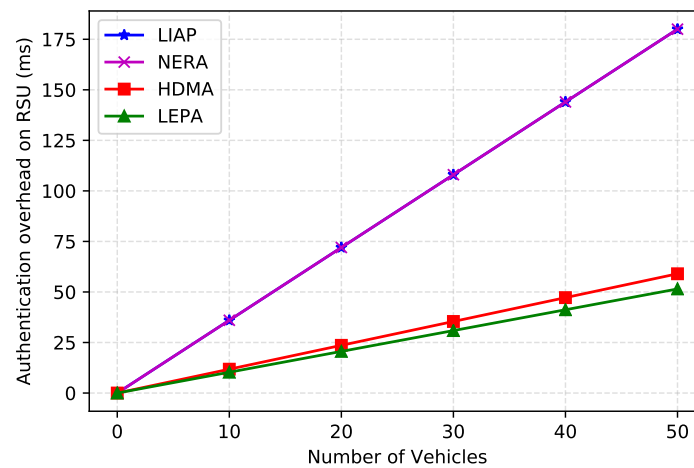
Table 5 presents the comparison of the authentication overheads of our LEPA scheme with LIAP, NERA, and HDMA schemes.

**Table 5.** V2I Authentication Overheads.

| Schemes | Authentication Overheads on Vehicle | Authentication Overheads on RSU        |
|---------|-------------------------------------|----------------------------------------|
| NERA    | $T_{MUL} + T_{MTP} + 3T_{PAR}$      | $T_{MUL} + T_{MTP} + 3T_{PAR}$         |
| HDMA    | $T_{RSA\_V} + T_{RSA\_D}$           | $T_{RSA\_V} + T_{RSA\_D} + T_{RSA\_E}$ |
| LIAP    | $T_{MUL} + T_{MTP} + 3T_{PAR}$      | $T_{MUL} + T_{MTP} + 3T_{PAR}$         |
| LEPA    | $T_{RSA\_E}$                        | $(\log n + 1)H + T_{RSA\_D}$           |

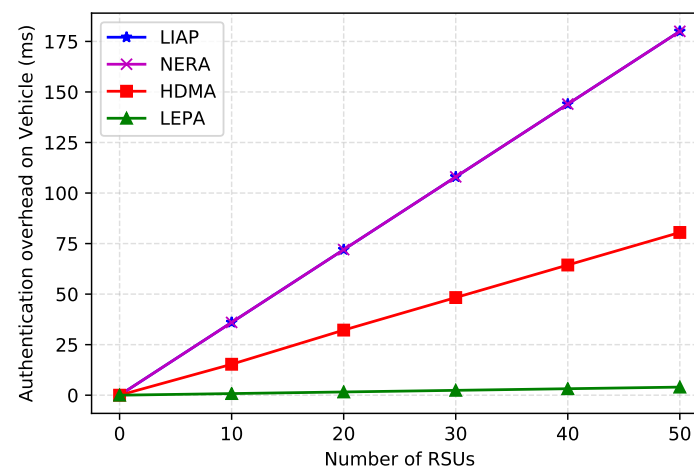
In Table 5,  $T_{RSA\_E}$  denotes the time for RSA-1024-bit encryption operation, while  $T_{RSA\_D}$  denotes the time of the RSA-1024-bit decryption operation. Similarly,  $T_{RSA\_V}$  refers to Verification time for the RSA 1024 algorithm. The time for an SHA-256 hash operation time is

represented by  $H$ . The variable  $n$  denotes the number of pseudonyms assigned to a vehicle, and  $T_{MUL}$  denotes the time needed to perform one point multiplication over  $\mathbb{G}_1$ ,  $T_{MTP}$  denotes the time needed to perform a MapToPoint hash operation, and  $T_{PAR}$  denotes the time required for executing a pairing operation. For this analysis, we select a base field size of 160-bit ( $q = 160$  bits) for the bilinear map [19], achieving security equivalent to 1024-bit RSA. In addition,  $\mathbb{G}_1$  is represented by 161 bits to ensure compatibility with the base field size. Both LIAP [8] and NERA [9] use bilinear pairing and Map-To-Point operations and the computation overhead is  $T_{MUL} + T_{MTP} + 3T_{PAR}$  (where  $T_{MUL} = 39$  ms,  $T_{MTP} = 0.09$  ms and  $T_{PAR} = 3.21$  ms). The required computations time for  $T_{RSA\_E} = 0.08$  ms,  $H = 111$  MiB/s,  $T_{RSA\_D} = 1.46$  ms, and  $T_{RSA\_V} = 0.07$  ms using a Crypto++ 5.6.0 [20] that runs on an Intel Core 2 1.83 GHz processor using Microsoft Visual Studio 2017. The comparison of authentication overhead of vehicles on RSU in the LEPA scheme with that of LIAP [8], NERA [9], and HDMA [10] is illustrated in Figure 7. As the number of vehicles grows in LIAP and NERA, computation costs increase significantly. In contrast, vehicle authentication costs under HDMA are low and very low under the LEPA scheme.



**Figure 7.** Computation overhead on RSU for authentication. Reprinted with permission from [3]. Copyright Year:2022; Copyright Owner's Name: IEEE.

Figure 8 shows that our LEPA scheme has significantly lower vehicle authentication overhead compared to LIAP, NERA, and HDMA. When there are 20 RSUs, for instance, the computation cost is approximately 75 ms for both LIAP [8] and NERA [9] and 30 ms for HDMA [10]. In contrast, the computation cost for our LEPA scheme is only 1.2 ms.



**Figure 8.** Computation overhead on vehicle for authentication. Reprinted with permission from [3]. Copyright Year:2022; Copyright Owner's Name: IEEE.

### 5.5. Communication Overhead

In LEPA, the LTA periodically broadcasts the signed Cuckoo Filters  $Sig_{LTA(PCF_V, NCF_V)}$  and  $Sig_{LTA(PCF_{RSU}, NCF_{RSU})}$  to all entities within its region. Each Cuckoo Filter is a compact probabilistic data structure whose size depends on the number of enrolled entries and the target false positive rate, not on the number of revoked vehicles. For a deployment of 4 M vehicles with a false positive rate of  $\epsilon = 10^{-3}$  (a Cuckoo Filter requires approximately 14–16 bits per entry) and a total filter size of approximately 50 MB, a reduction of four orders of magnitude compared to the 525 GB CRL broadcast in NERA and LIAP. LEPA eliminates per-vehicle broadcast entirely. Rather than flooding the network with a full CRL on every cycle, the LTA distributes a single compact filter update to RSUs, which stores it locally. Authentication itself proceeds through a simple unicast message exchange:  $V_i$  sends a single encrypted authentication request to  $RSU_j$ , and  $RSU_j$  responds with a single unicast reply. An RSU performs all revocation lookups locally against its cached filters in  $O(1)$  time, with no communication required beyond that single message exchange. Consequently, the per-session communication cost in LEPA does not depend on the number of active vehicles or the number of revoked vehicles, making it more efficient than CRL-based schemes. This contrasts sharply with NERA and LIAP, where an RSU's downlink is burdened by  $O(N \times |CRL|)$  broadcast message overhead per cycle, and simultaneous CRL downloads from the CA create synchronization congestion across the infrastructure. HDMA reduces this cost through hierarchical multicast distribution but does not eliminate it. CRL segments must still be disseminated across the network, and overhead remains proportional to revocation list size. LEPA avoids both the broadcast overhead and the synchronization bottleneck: revocation updates are incremental  $O(1)$  time filter insertions propagated by the LTA, requiring no coordinated mass download and imposing no measurable channel load on the vehicle-RSU link. Table 6 compares the communication overhead of NERA, HDMA, LIAP, and LEPA.

**Table 6.** Communication Overhead Comparison.

| Schemes | Revocation Distributed        | Data | Messages per Session | Message Type | Volume of Message per Vehicle |
|---------|-------------------------------|------|----------------------|--------------|-------------------------------|
| NERA    | Full CRL(pseudonyms)          |      | $O(N)$               | Broadcast    | $\approx 525$ GB              |
| HDMA    | Partial CRL (hierarchical)    |      | $O(\log(N))$         | Multicast    | High                          |
| LIAP    | VCRL+ RCRL (bilinear pairing) |      | $O(N)$               | Broadcast    | $\approx 525$ GB              |
| LEPA    | Cuckoo Filter                 |      | $O(1)$               | Unicast      | $\approx 50$ MB               |

## 6. Related Works

To ensure secure communication and enhance transportation efficiency, it is crucial to remove compromised or misbehaving entities from the VANET. Numerous privacy-preserving authentication schemes [21–24] in VANET rely on Certificate Revocation Lists (CRLs) to store information about revoked entities. However, as the number of revoked entities increases, the size of the CRL grows substantially, leading to significant computation and communication overhead. For instance, in a pseudonym-based approach [18], where a vehicle changes its pseudonym every 120 s, it would require approximately 262,800 pseudonyms annually (or 21,900 pseudonyms per month). To maintain communication security, all unexpired pseudonyms of a malicious vehicle must be revoked. This

requirement exacerbates the growth of the CRL. Consequently, managing such a large and continuously growing CRL becomes a challenging and resource-intensive task.

To address the computational and storage overhead associated with CRLs, Zhu et al. [24] proposed a privacy-preserving authentication scheme based on group signatures. In this scheme, the region is divided into multiple domains, with each RSU responsible for authenticating vehicles within its domain. Upon successful authentication, the RSU generates and distributes a group key to authenticated vehicles, enabling the use of hash-based message authentication codes (HMACs) for secure message authentication. This scheme assumes that RSUs are inherently trustworthy. However, this assumption introduces a critical vulnerability, as malicious entities can impersonate RSUs to broadcast false information or steal sensitive data from vehicle owners [25] threats highlight the need for enhanced mechanisms to ensure the integrity and trustworthiness of RSUs in VANET environments.

Wang and Yuo [8] proposed a Local Identity-based Anonymous Message Authentication Protocol (LIAP) for VANETs. In this scheme, both vehicles and RSUs obtain long-term certificates from a Certificate Authority (CA) during the registration process. The CA is responsible for revoking the certificates of malicious vehicles and RSUs, subsequently adding them to the Certificate Revocation Lists for Vehicles (VCRLs) and RSUs (RCRLs), respectively. To authenticate one another, vehicles and RSUs cross-check the VCRLs and RCRLs. However, this process introduces significant computational and storage overhead due to the increasing size of the CRLs as the number of revoked entities grows.

Wang et al. [10] proposed a Hybrid D2D Message Authentication (HDMA) scheme for VANETs with 5G capability. In this scheme, the Certificate Authority (CA) generates and distributes a set of pseudonyms and corresponding certificates for each vehicle registered under it. Vehicles authenticate themselves by sending their pseudonym certificates to the Road Side Base Station (RSBS) or Road Side Unit (RSU). Upon receiving a message from a vehicle, the RSU or RSBS verifies the pseudonym against the Certificate Revocation List (CRL) to check if it has been revoked. Additionally, a zero-knowledge proof algorithm is utilized for Vehicle-to-Infrastructure (V2I) authentication.

To reduce the computational overhead associated with CRL checking, Liu et al. [26] introduced a Lightweight V2I Authentication Protocol (LVAP). In this protocol, the Trusted Authority (TA) predicts which RSUs the On-Board Unit (OBU) will encounter along its route and forms a group comprising the identified RSUs and incoming vehicles. The TA generates and distributes group keys to streamline the authentication process. To enhance security, all RSUs receive updates from the TA regarding misbehaving vehicles. However, this scheme requires the TA to maintain a list of revoked vehicles, and it does not protect the route privacy of vehicles. To address the issue of route privacy, Lv and Liu [27] proposed a Privacy-Preserving and Lightweight V2I Authentication (PLVA) scheme using BGN homomorphic encryption. In this approach, the CA assists vehicles in obtaining information about RSUs along their route to enable fast authentication. Importantly, the CA remains unaware of which RSU's information was accessed by the vehicle. However, as the number of RSUs increases, this scheme incurs additional computational and communication overhead. Wang and Liu [28] proposed a Secure and Efficient Message Authentication (SEMA) protocol that combines pseudonym-based and group-based methods. In this scheme, the TA generates a set of pseudonyms and corresponding certificates for each registered vehicle. RSUs authenticate vehicles by checking the CRL and, for valid vehicles, generate a local group key. This group key is then used by vehicles within the RSU's region for secure Vehicle-to-Vehicle (V2V) communication, enabling efficient message exchange. However, the process of CRL checking imposes computational and storage overhead on the RSUs, which can affect the system's overall performance.

Bayat et al. [9] proposed an efficient RSU-based authentication scheme that leverages bilinear pairing and the Map-To-Point operation. In this scheme, when a vehicle enters an RSU's region to participate in VANET communication, the RSU performs mutual authentication with the vehicle and subsequently generates a set of pseudo-IDs and corresponding secret keys for it. Unlike traditional CRLs, this scheme only includes the real IDs of malicious vehicles in the CRL, significantly reducing its size and the associated computational overhead. However, a notable limitation of this approach is the absence of message generation timestamps in transmitted messages, rendering the scheme vulnerable to replay attacks.

The Cuckoo Filter (CF) has gained significant attention in recent VANET research [29–32] due to its efficient computational performance. Cui et al. [33] proposed a Cuckoo Filter-based message authentication scheme for VANETs utilizing edge computing. In this scheme, the RSU selects certain vehicles within its region to act as Edge Computing Vehicles (ECVs). These ECVs assist the RSU in verifying message signatures from other vehicles within its transmission range. The RSU stores authentication information in a Cuckoo Filter, which is then broadcast to all vehicles in its region. Vehicles use the cuckoo filter to quickly verify the authenticity of messages, significantly reducing authentication overhead.

Zhao et al. [34] proposed a Cuckoo-Filter-based pseudonym certificate revocation scheme for VANETs. In this approach, the CA computes the fingerprints of all unexpired pseudonym certificates for revoked vehicles and stores them in a Cuckoo Filter. The CA periodically broadcasts this filter to facilitate vehicle authentication. Compared to traditional CRLs, this method significantly reduces computational overhead. However, storing all unexpired pseudonyms of revoked vehicles in the Cuckoo Filter can increase both storage and computational overhead, particularly as the number of revoked pseudonyms grows.

Wang et al. [35] propose a solution to offload partial revocation tasks to network edges, streamlining the revocation process in vehicles. Their method introduces decentralized management of the Global Certificate Revocation List (GCRL) using permissioned blockchain technology, ensuring tamper-proof properties and fast synchronization of the GCRL. Additionally, the method includes mechanisms to detect illegal revocation behaviors through vehicle validation. To evaluate their approach, the authors developed a prototype using Hyperledger Fabric and compared its blockchain synchronization latency performance with that of the Proof-of-Work scheme, demonstrating its efficiency.

Highlighting the drawbacks of an existing certificate-based authentication in VANETs, Zhou et al. [36] propose a security-enhanced solution: a certificate-less conditional privacy-preserving authentication scheme. To demonstrate the feasibility and practicality of their approach, they conduct a thorough security analysis within a formal security model. Furthermore, they evaluate the performance of their proposed scheme by comparing it with existing methods, showcasing its advantages in terms of security and efficiency.

Sang et al. [37] propose PACM, a privacy-preserving authentication scheme for VANETs with on-chain certificate management. In this approach, service managers (SMs) in each domain act as blockchain nodes to create a distributed system. Using elliptic curve cryptography (ECC) and XOR operations, the scheme ensures secure and lightweight mutual authentication between vehicles and RSUs with regularly updated pseudonyms. Blockchain records certificate issuance and revocation, ensuring transparency, while the Counting Garbled Bloom Filter (CGBF) enables fast certificate queries and revocations. Additionally, a hash-chain-based billing mechanism ensures non-forgeable and non-repudiable transactions. Security analysis and experiments confirm PACM's stronger security and lower overhead.

Liang et al. [38] propose a Multilayer Coded Caching CRL scheme (MCC-CRL) for VANETs. The MCC-CRL minimizes the distribution of CRL data to meet revocation require-

ments, significantly reducing communication overhead and processing time. Additionally, it incorporates an efficient authentication mechanism to ensure conditional privacy preservation in a computationally efficient manner. Extensive simulations demonstrate that the MCC-CRL scheme is both secure and effective for vehicle revocation in VANET environments.

Jayashree et al. [39] propose a group-signature-based certificate-less verification scheme (EGSCVS) designed to support lightweight end-to-end authentication in VANETs. EGSCVS utilizes a certificate-less aggregation signature mechanism, which leverages the efficiency of Elliptic Curve Cryptography (ECC) to optimize storage requirements and minimize communication and computational overhead during the authentication process. By eliminating the need for certificate management, the EGSCVS scheme reduces the complexity associated with traditional certificate-based systems. Furthermore, the use of ECC ensures strong cryptographic security while maintaining low computational costs.

Li et al. [40] propose an innovative and efficient certificate-less message authentication scheme. The proposed scheme introduces a lightweight security authentication protocol by enhancing the RSA dynamic accumulator and integrating it with non-interactive discrete logarithm zero-knowledge proofs. A key feature of their scheme is the elimination of bilinear pairing operations, which significantly reduces computational overhead. Furthermore, the scheme supports real-time tracking and revocation of malicious vehicle identities, addressing critical security challenges in the Internet of Vehicles (IoV). The security analysis demonstrates that the scheme is provably secure under the S-RSA assumption in the random oracle model, meeting the diverse security requirements of IoV environments. Performance evaluation highlights the scheme's efficiency, showing considerable reductions in both computational and communication overheads.

Zhang et al. [41] propose a conditional privacy-preserving authentication protocol using a Shared Group Session Key (SGSK) that integrates self-healing key distribution, blockchain, and the MTI/C0 protocol. Their approach replaces time-consuming CRL checks with SGSK updates to revoke malicious vehicles. The SGSK is shared among unrevoked vehicles within and across domains, preventing revoked vehicles from accessing the system or sending false messages. The protocol improves computational efficiency by reducing bilinear pairing operations while resisting various attacks and maintaining conditional privacy. Implemented on the Hyperledger Fabric platform, the protocol revokes 180 malicious vehicles across domains within one second and verifies 600 messages per second. Performance evaluations show it outperforms existing methods in revocation cost, computational overhead, and communication overhead. To validate its practicality, Zhang et al. used SUMO and NS2 to simulate VANET scenarios, demonstrating the protocol's efficiency and effectiveness for real-world applications.

Many recent works propose a wide range of authentication and privacy-preserving mechanisms for VANETs, leveraging techniques such as lightweight cryptography, ECC, PUFs, blockchain, pseudonym systems, and AI-based models (e.g., Kaur et al. [42], Li et al. [43], Chen et al. [44], Mukathe et al. [45]). While these approaches report strong security guarantees and improved efficiency through formal analyses and simulations, many rely on idealized assumptions and fail to comprehensively address realistic adversarial conditions such as compromised infrastructure, side-channel attacks, and dynamic network behavior (as highlighted by Liu et al. [46], and Das et al. [47]). Moreover, although privacy-enhancing techniques like pseudonym updates, token-based authentication, and conditional anonymity are widely adopted (e.g., Luo et al. [48], Ming et al. [49], Tan et al. [50]), privacy and traceability remains only partially resolved. Recent research works using blockchain and reinforcement learning (e.g., Li et al. [51], Mukathe et al. [45]) also raise concerns regarding scalability, latency, and deployment feasibility in resource-constrained and high-mobility environments.

Many of the recent surveys [52–55] highlight the growing need for secure and privacy-preserving authentication in VANETs, given their vulnerability to attacks such as impersonation, eavesdropping, and insider threats. While various approaches—such as certificate-less, pseudonym-based, and ECC-based schemes—aim to balance privacy and traceability, they often suffer from limitations in security, efficiency, and scalability. Emerging techniques, including AI, reinforcement learning, blockchain, and post-quantum cryptography, show promise, but significant gaps remain, necessitating hybrid solutions, standardization, and improved interoperability.

**Our contribution:** The LEPA scheme proposed in this paper addresses the limitations of traditional CRL-based approaches. Conventional pseudonym-based authentication schemes incur significant computation and storage overhead due to the need to manage and distribute large CRLs, particularly when each pseudonym of a revoked vehicle must be individually recorded.

To overcome this limitation, the proposed LEPA scheme leverages the computational efficiency related to using Cuckoo Filters in conjunction with Merkle Hash Trees. Instead of storing all pseudonyms of a revoked vehicle, LEPA inserts only a single compact entry—the corresponding MHT root ( $MHT_{\text{root}}$ )—into the Cuckoo Filter. This design enables efficient aggregation of all pseudonyms associated with a vehicle, significantly reducing storage requirements and accelerating revocation checks during authentication.

Furthermore, LEPA optimizes lookup operations for Vehicle-to-Infrastructure (V2I) communication by enabling both vehicles and RSUs to perform fast membership queries using Cuckoo Filters. This results in low-latency authentication and improved scalability, even in large-scale VANET deployments.

By integrating privacy preservation, compact representation of information related to revocation, and efficient authentication mechanisms, the proposed LEPA scheme provides a scalable and robust solution for secure and authenticated VANET communication.

## 7. Acknowledgment

A preliminary version of this work was presented at the IEEE 19th Annual Consumer Communications & Networking Conference (CCNC) [3]. The conference paper, limited to five pages, provided an initial overview of the proposed lightweight authentication approach. In contrast, this journal version presents a substantially extended and more comprehensive treatment of the LEPA scheme.

Specifically, this paper includes a detailed system model, formal security analysis and verification, and an in-depth evaluation of communication and computational overhead. It further provides enhanced design descriptions, expanded performance analysis, and additional discussions on scalability and robustness in large-scale VANET environments.

Moreover, parts of this work are derived from the Ph.D. dissertation of the first author [56], where preliminary concepts and foundational components of the proposed scheme were initially developed. This manuscript consolidates, refines, and significantly extends those contributions into a unified and rigorous framework.

## 8. Conclusions

This paper presents a Low-overhead and Efficient Privacy-preserving Authentication (LEPA) scheme for VANETs, leveraging the capabilities of Cuckoo Filters to optimize the authentication process. In the proposed LEPA scheme, both vehicles and RSUs utilize Cuckoo Filters for mutual authentication, enabling a more efficient and scalable verification process compared to traditional CRL-based methods. By significantly reducing the computational and storage overhead, the LEPA scheme ensures fast and secure communication in vehicular networks.

The security analysis and verification of the LEPA scheme demonstrate its robustness against various attacks, including message modification, replay, and message injection attacks. These results underscore the scheme's ability to maintain the integrity and confidentiality of messages exchanged within VANET environments.

Furthermore, the performance analysis reveals that the LEPA scheme outperforms other comparable authentication protocols in terms of computational cost, making it a practical solution for real-world deployment in VANET systems. The efficient use of Cuckoo Filters enables rapid lookup and verification, ensuring that authentication operations do not become a bottleneck in high-density traffic scenarios.

As a future extension, this research aims to expand the use of Cuckoo Filters for efficient Vehicle-to-Vehicle (V2V) authentication. By adapting the scheme for V2V communication, the proposed methodology could further enhance the scalability and security of VANET systems while maintaining low computational overhead.

**Author Contributions:** Conceptualization, S.S.M. and D.M.; methodology, S.S.M. and D.M.; software, S.S.M.; validation, S.S.M. and D.M.; formal analysis, S.S.M.; investigation, S.S.M.; resources, D.M.; writing—original draft preparation, S.S.M.; writing—review and editing, D.M.; supervision, D.M.; project administration, D.M.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No data was used for this research.

**Acknowledgments:** No support not covered by the author contribution or funding sections was received.

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

## References

1. Merkle, R.C. Protocols for public key cryptosystems. In Proceedings of the 1980 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 14–16 April 1980.
2. Fan, B.; Andersen, D.G.; Kaminsky, M.; Mitzenmacher, M.D. Cuckoo Filter: Practically better than bloom. In Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies, Sydney, Australia, 2–5 December 2014; pp. 75–88.
3. Moni, S.S.; Manivannan, D. A lightweight privacy-preserving V2I mutual authentication scheme using Cuckoo Filter in VANETs. In 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2022; pp. 815–820.
4. Rivest, R.L. RFC 1321: *The MD5 Message-Digest Algorithm*; Technical Report; Internet Activities Board. 1992. Available online: <https://www.ietf.org/rfc/rfc1321.txt> (accessed on 3 April 2026).
5. Ramabaja, L.; Avdullahu, A. Compact merkle multiproofs. *arXiv* **2020**, arXiv:2002.07648.
6. Irisa: SPAN. 2006. Available online: <http://people.irisa.fr/Thomas.Genet/span/> (accessed on 2 February 2020).
7. Avispa: AVISPA. 2002. Available online: <http://www.avispa-project.org> (accessed on 2 February 2020).
8. Wang, S.; Yao, N. LIAP: A local identity-based anonymous message authentication protocol in VANETs. *Comput. Commun.* **2017**, *112*, 154–164.
9. Bayat, M.; Pournaghi, M.; Rahimi, M.; Barmshoory, M. NERA: A new and efficient RSU based authentication scheme for VANETs. *Wirel. Netw.* **2020**, *26*, 3083–3098.
10. Wang, P.; Chen, C.-M.; Kumari, S.; Shojafar, M.; Tafazolli, R.; Liu, Y.-N. HDMA: Hybrid D2D message authentication scheme for 5G-enabled VANETs. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 5071–5080.
11. Adhikari, S.; Ray, S.; Biswas, G.P.; Obaidat, M.S. Efficient and secure business model for content centric network using elliptic curve cryptography. *Int. J. Commun. Syst.* **2019**, *32*, 3839.
12. Amin, R.; Islam, S.H.; Obaidat, M.S.; Biswas, G.; Hsiao, K.-F. An anonymous and robust multi-server authentication protocol using multiple registration servers. *Int. J. Commun. Syst.* **2017**, *30*, 3457.
13. Moon, J.; Lee, Y.; Kim, J.; Won, D. Improving an anonymous and provably secure authentication protocol for a mobile user. *Secur. Commun. Netw.* **2017**, *2017*, 1378128.

14. Benarous, L.; Kadri, B.; Bitam, S.; Mellouk, A. Privacy-preserving authentication scheme for on-road on-demand refilling of pseudonym in VANET. *Int. J. Commun. Syst.* **2020**, *33*, 4087.
15. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36.
16. Sufatrio, Yap, R.H.C. Extending BAN logic for reasoning with modern PKI-based protocols. In *Proceedings of the 2008 IFIP International Conference on Network and Parallel Computing*, Shanghai, China, 18–20 October 2008; pp. 190–197.
17. Moni, S.S.; Manivannan, D. A scalable and distributed architecture for secure and privacy-preserving authentication and message dissemination in VANETs. *Internet Things* **2020**, *13*, 100350.
18. Standard, S.V. *Dedicated Short Range Communications (DSRC) Message Set Dictionary*; SAE International: Warrendale, PA, USA, 2006. [https://doi.org/10.4271/J2735\\_201603](https://doi.org/10.4271/J2735_201603).
19. Barreto, P.; Lynn, B.; Scott, M. Efficient Implementation of Pairing-Based Cryptosystems. *Cryptology* **2004**, *17*, 321–334.
20. Dai, W. Crypto++ Library 5.6.0. 2009. Available online: <http://www.cryptopp.com/release560.html> (accessed on 5 May 2026).
21. Manivannan, D.; Moni, S.S.; Zeadally, S. Secure authentication and privacy-preserving techniques in Vehicular Ad-hoc NETWORKS (VANETs). *Veh. Commun.* **2020**, *25*, 100247.
22. Wasef, A.; Shen, X. EMAP: Expedite message authentication protocol for vehicular ad hoc networks. *IEEE Trans. Mob. Comput.* **2013**, *12*, 78–89.
23. Jiang, S.; Zhu, X.; Wang, L. A conditional privacy scheme based on anonymized batch authentication in vehicular ad hoc networks. In *Proceedings of the IEEE WCNC*, Shanghai, China, 7–10 April 2013; pp. 2375–2380.
24. Zhu, X.; Jiang, S.; Wang, L.; Li, H.; Zhang, W.; Li, Z. Privacy preserving authentication based on group signature for VANETs. In *Proceedings of the IEEE Global Telecommunications Conference Workshops Atlanta, GA, USA*, 9–13 December, 2013.
25. Moni, S.S.; Manivannan, D. An efficient RSU authentication scheme based on Merkle Hash Tree for VANETs. In *Proceedings of the 2020 IEEE International Conference on Communications (ICC)*, Virtually, 7–11 June 2020; pp. 1–7. <https://doi.org/10.1109/ICC40277.2020.9149028>.
26. Liu, Y.; Guo, W.; Zhong, Q.; Yao, G. LVAP: Lightweight V2I authentication protocol using group communication in VANETs. *Int. J. Commun. Syst.* **2017**, *30*, 3317.
27. Lv, S.; Liu, Y. Plva: Privacy-preserving and lightweight v2i authentication protocol. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 6633–6639.
28. Wang, P.; Liu, Y. SEMA: Secure and efficient message authentication protocol for VANETs. *IEEE Syst. J.* **2021**, *15*, 846–855.
29. Cui, J.; Zhang, J.; Zhong, H.; Xu, Y. SPACF: A secure privacy-preserving authentication scheme for VANET with Cuckoo Filter. *IEEE Trans. Veh. Technol.* **2017**, *66*, 10283–10295.
30. Li, K.; Lau, W.F.; Au, M.H. A secure and efficient privacy-preserving authentication scheme for vehicular networks with batch verification using Cuckoo Filter. In *Proceedings of the International Conference on Network and System Security, Sapporo, Japan, December 15–18, 2019*; Springer: Berlin/Heidelberg, Germany 2019; pp. 615–631.
31. Pan, J.; Cui, J.; Wei, L.; Xu, Y.; Zhong, H. Secure data sharing scheme for VANETs based on edge computing. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 169.
32. Xie, Z.; Ding, W.; Wang, H.; Xiao, Y.; Liu, Z. D-ary Cuckoo Filter: A space efficient data structure for set membership lookup. In *Proceedings of the 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*; IEEE: Shenzhen, China, 15–17 December 2017; pp. 190–197.
33. 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.
34. Zhang, H.; Zhang, D.; Chen, H.; Xu, J. Improving efficiency of pseudonym revocation in VANET using cuckoo filter. In *Proceedings of the 2020 IEEE 20th International Conference on Communication Technology (ICCT)*, Nanning, China, 28–31 October 2020; pp. 763–769.
35. Wang, Q.; Gao, D.; Foh, C.H.; Zhang, H.; Leung, V.C.M. Decentralized CRL management for vehicular networks with permissioned blockchain. *IEEE Trans. Veh. Technol.* **2022**, *71*, 11408–11420. <https://doi.org/10.1109/TVT.2022.3194206>.
36. Zhou, X.; Luo, M.; Vijayakumar, P.; Peng, C.; He, D. Efficient certificateless conditional privacy-preserving authentication for VANETs. *IEEE Trans. Veh. Technol.* **2022**, *71*, 7863–7875. <https://doi.org/10.1109/TVT.2022.3169948>.
37. Sang, G.; Chen, J.; Liu, Y.; Wu, H.; Zhou, Y.; Jiang, S. PACM: Privacy-preserving authentication scheme with on-chain certificate management for VANETs. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 216–228. <https://doi.org/10.1109/TNSM.2022.3201551>.
38. Liang, J.; Ma, M.; Yang, G. Efficient-lightweight CRL distribution in VANETs: A multilayer coded caching methodology. In *Proceedings of the 2023 IEEE Wireless Communications and Networking Conference (WCNC)*, Glasgow, Scotland, 26–29 March 2023; pp. 1–6. <https://doi.org/10.1109/WCNC55385.2023.10118685>.
39. Jayashree, S.; Kumar, S.V.N.S. An efficient group signature based certificate less verification scheme for Vehicular Ad-hoc Network. *Wirel. Netw.* **2024**, *30*, 3269–3298.

40. Li, Y.; Cao, L.; Zheng, G.; Men, H.; Chen, L. Improved rsa dynamic cryptographic accumulator-based anonymous batch authentication scheme for internet of vehicles. *Comput. Electr. Eng.* **2024**, *117*, 109261. <https://doi.org/10.1016/j.compeleceng.2024.109261>.
41. Zhang, R.; Zhou, W. Shared group session key-based conditional privacy-preserving authentication protocol for VANETs. *VEhicular Commun.* **2024**, *47*, 100782. <https://doi.org/10.1016/j.vehcom.2024.100782>.
42. Kaur, G.; Kakkar, D. A secure lightweight authentication model with interference aware routing and attack detection approach in VANET. *Clust. Comput.* **2025**, *28*, 109. <https://doi.org/10.1007/s10586-024-04772-1>.
43. Li, J.; Chen, J.; Liu, Y.; Liu, S.; Zhang, Y. An efficient and revocable PUF-based authentication scheme for secure V2R mutual communication in VANETs. *IEEE Internet Things J.* **2025**, *12*, 37974–37987. <https://doi.org/10.1109/JIOT.2025.3585643>.
44. Chen, X.; Zhang, X.; Zhong, S.; Liu, S. Anonymous authentication based on blockchain and zero-knowledge proof for vehicular ad hoc networks. *J. Supercomput.* **2025**, *81*, 1416.
45. Mukathe, D.; Di, W.; Ahmed, W.; Worku, T. Blockchain-powered authenticated key agreement scheme with reputation-incentive mechanism for Vehicle-to-Vehicle communication in IoV. *IEEE Internet Things J.* **2025**, *12*, 25500–25515. <https://doi.org/10.1109/JIOT.2025.3558278>.
46. Liu, S.; Wang, Z.; Chen, C.-M. Enhanced authentication and key agreement protocol for VANETs. *IEEE Internet Things J.* **2026**, *13*, 7680–7694. <https://doi.org/10.1109/JIOT.2025.3640015>.
47. Das, P.; Ray, S.; Dasgupta, M.; Kumari, S.; Chen, C.-M.; Chandra Govil, M.; Amoon, M. Mutual authentication and trust establishment (MATE) protocol in VANET using puncturable pseudorandom function (PPRF): MATE-PPRF. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 13293–13305. <https://doi.org/10.1109/TITS.2025.3589123>.
48. Luo, S.; Cao, X.; Li, X.; Ren, Z.; Wang, Y.; Miao, Y. Efficient revocable conditional anonymous authentication with verifiable self-generated pseudonyms for VANETs. *IEEE Trans. Inf. Forensics Secur.* **2026**, *21*, 2147–2161. <https://doi.org/10.1109/TIFS.2026.3657109>.
49. Ming, Y.; Zhang, X.; Liu, H.; Wang, C.; Wang, S. Communication-efficient conditional privacy-preserving authentication based on ring signature in VANETs. *IEEE Trans. Veh. Technol.* **2026**, *75*, 4870–4881. <https://doi.org/10.1109/TVT.2025.3610447>.
50. Tan, H.; Wang, M.; Shen, J.; Vijayakumar, P.; Moh, S.; Wu, Q.M.J. Blockchain-assisted conditional anonymous authentication and adaptive tree-based group key agreement for VANETs. *IEEE Trans. Dependable Secur. Comput.* **2026**, *23*, 2664–2679. <https://doi.org/10.1109/TDSC.2025.3628884>.
51. Li, R.; Cui, J.; Zhang, J.; Wei, L.; Zhong, H.; He, D. Blockchain-assisted revocable cross-domain authentication for Vehicular Ad-hoc Networks. *IEEE Trans. Dependable Secur. Comput.* **2025**, *22*, 4593–4606. <https://doi.org/10.1109/TDSC.2025.3549748>.
52. Manasrah, A.; Yaseen, Q.; Al-Aqrabi, H.; Liu, L. Identity-based authentication in VANETs: A review. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 4260–4282. <https://doi.org/10.1109/TITS.2025.3528932>.
53. Rahmawati Agustina, E.; Ramli, K.; Rahman Hakim, A.; Harwahyu, R. A systematic literature review on privacy preservation in VANETs: Trends, challenges, and future directions. *IEEE Access* **2025**, *13*, 88421–88444. <https://doi.org/10.1109/ACCESS.2025.3570491>.
54. Liu, S.; Wang, Z. Security analysis on a two-factor privacy-preserving protocol for efficient authentication in internet of vehicles networks. *IEEE Internet Things J.* **2025**, *12*, 34623–34632. <https://doi.org/10.1109/JIOT.2025.3562393>.
55. Mohammadzadeh Mianji, E.; Muntean, G.-M.; Tal, I. Enhancing vehicular network security, privacy, and trust through reinforcement learning: A comprehensive survey. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 21393–21426. <https://doi.org/10.1109/TITS.2025.3612202>.
56. Moni, S.S. Protocols and Architecture for Privacy-Preserving Authentication and Secure Message Dissemination in Vehicular Ad Hoc Networks. Ph.D. Thesis, University of Kentucky, Lexington, KY, USA, 2022.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.