




2022

Protocols and Architecture for Privacy-preserving Authentication and Secure Message Dissemination in Vehicular Ad Hoc Networks

Shafika Showkat Moni

University of Kentucky, shafika1403@gmail.com

Author ORCID Identifier:

 <https://orcid.org/0000-0002-7710-4217>

Digital Object Identifier: <https://doi.org/10.13023/etd.2022.055>

[Right click to open a feedback form in a new tab to let us know how this document benefits you.](#)

Recommended Citation

Moni, Shafika Showkat, "Protocols and Architecture for Privacy-preserving Authentication and Secure Message Dissemination in Vehicular Ad Hoc Networks" (2022). *Theses and Dissertations--Computer Science*. 116.

https://uknowledge.uky.edu/cs_etds/116

This Doctoral Dissertation is brought to you for free and open access by the Computer Science at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Computer Science by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Shafika Showkat Moni, Student

Dr. D. Manivannan, Major Professor

Dr. Simone Silvestri, Director of Graduate Studies

Protocols and Architecture for Privacy-preserving Authentication and Secure
Message Dissemination in Vehicular Ad Hoc Networks

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
in the College of Engineering at the
University of Kentucky

By
Shafika Showkat Moni
Lexington, Kentucky

Director: Dr. D. Manivannan, Associate Professor of Computer Science
Lexington, Kentucky
2022

Copyright© Shafika Showkat Moni 2022

ABSTRACT OF DISSERTATION

Protocols and Architecture for Privacy-preserving Authentication and Secure Message Dissemination in Vehicular Ad Hoc Networks

The rapid development in the automotive industry and wireless communication technologies have enhanced the popularity of Vehicular ad hoc networks (VANETs). Today, the automobile industry is developing sophisticated sensors that can provide a wide range of assistive features, including accident avoidance, automatic lane tracking, semi-autonomous driving, suggested lane changes, and more. VANETs can provide drivers a safer and more comfortable driving experience, as well as many other useful services by leveraging such technological advancements. Even though this networking technology enables smart and autonomous driving, it also introduces a plethora of attack vectors. However, the main issues to be sorted out and addressed for the widespread deployment/adoption of VANETs are privacy, authenticating users, and the distribution of secure messages. These issues have been addressed in this dissertation, and the contributions of this dissertation are summarized as follows:

Secure and privacy-preserving authentication and message dissemination in VANETs: Attackers can compromise the messages disseminated within VANETs by tampering with the message content or sending malicious messages. Therefore, it is crucial to ensure the legitimacy of the vehicles participating in the VANETs as well as the integrity and authenticity of the messages transmitted in VANETs. In VANET communication, the vehicle uses pseudonyms instead of its real identity to protect its privacy. However, the real identity of a vehicle must be revealed when it is determined to be malicious. This dissertation presents a distributed and scalable privacy-preserving authentication and message dissemination scheme in VANET.

Low overhead privacy-preserving authentication scheme in VANETs: The traditional pseudonym-based authentication scheme uses Certificate Revocation Lists (CRLs) to store the certificates of revoked and malicious entities in VANETs. However, the size of CRL increases significantly with the increased number of revoked entities. Therefore, the overhead involved in maintaining the revoked certificates is overwhelming in CRL-based solutions. This dissertation presents a lightweight

privacy-preserving authentication scheme that reduces the overhead associated with maintaining CRLs in VANETs. Our scheme also provides an efficient look-up operation for CRLs.

Efficient management of pseudonyms for privacy-preserving authentication in VANETs: In VANETs, vehicles change pseudonyms frequently to avoid the traceability of attackers. However, if only one vehicle out of 100 vehicles changes its pseudonym, an intruder can easily breach the privacy of the vehicle by linking the old and new pseudonym. This dissertation presents an efficient method for managing pseudonyms of vehicles. In our scheme, vehicles within the same region simultaneously change their pseudonyms to reduce the chance of linking two pseudonyms to the same vehicle.

KEYWORDS: Vehicular Ad Hoc Networks, Authentication, Security, Privacy preservation.

Shafika Showkat Moni

April 26, 2022

Protocols and Architecture for Privacy-preserving Authentication and Secure
Message Dissemination in Vehicular Ad Hoc Networks

By
Shafika Showkat Moni

Dr. D. Manivannan
Director of Dissertation

Dr. Simone Silvestri
Director of Graduate Studies

April 26, 2022
Date

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Dr. D. Manivannan for his guidance, patience, and continuous support. I have benefited greatly from his broad knowledge and constructive suggestions. It has been a great honor for me to work under his supervision, and his faith in me throughout the years is greatly appreciated.

I am thankful to Dr. Zongming Fei, Dr. Hana Khamfroush, Dr. Sherali Zeadally, and Dr. Nelson K. Akafuah for serving on my doctoral committee and their valuable suggestions. I would also like to thank Dr. Mirek Truszczyński, Dr. Simone Silvestri, and Dr. Yi Pike for their support and encouragement.

I am grateful to my beloved parents for their endless support, care, and love. I would like to thank all my family members and friends for inspiring me and always reminding me of the end goal.

TABLE OF CONTENTS

Acknowledgments	iii
List of Tables	vii
List of Figures	viii
Chapter 1 Introduction	1
1.1 Background	2
1.1.1 VANET	2
1.1.2 Security in VANETs	3
1.1.3 Privacy in VANETs	4
1.1.4 Authentication in VANETs	4
1.1.5 Privacy-preserving, Authentication, and Secure Message Dis- semination in VANETs	5
1.2 Motivation for the Dissertation	21
1.3 Organization of the Dissertation	22
Chapter 2 Secure and Privacy-preserving Authentication and Message Dis- semination in VANETs	24
2.1 Related works	24
2.2 Background, System Model, and Design Goals	25
2.2.1 Merkle Hash Tree	26
2.2.2 Modified Merkle Patricia Trie	26
2.2.3 System Model	29
2.2.4 Design Goals	30
2.2.5 Assumptions	31
2.3 Proposed Scheme	31
2.3.1 System Initialization	32
2.3.2 Construction of MMPT for Storing Public Keys of Revoked Vehicles	33
2.3.3 Construction of MHT for Storing Public Keys of Registered RSUs	33
2.3.4 When a Newly Registered Vehicle V Enters the Region of an RSU under its Home RTA	34
2.3.5 When a Vehicle V Needs to Send a Message M to the Nearby RSU	36
2.3.6 When an RSU Receives a Message from a Vehicle V	36
2.3.7 When a vehicle V Moves from one RTA's Region to Another RTA's Region	38
2.4 Analysis and Proof of Correctness of our Protocol	39
2.4.1 Security Analysis	39
2.4.2 Scalability of the Proposed Protocol	40

2.4.3	Proof of Correctness of our protocol using BAN Logic	41
2.4.4	Proof of Correctness using Simulation Tools	46
2.5	Performance Evaluation	47
2.5.1	Vehicle Authentication Overhead	47
2.5.2	RSU Authentication Overhead	49
2.5.3	Communication Overhead	49
2.6	Summary	50
Chapter 3	Low Overhead Privacy-preserving Authentication Scheme in VANETs	52
3.1	Background	52
3.1.1	Cuckoo Filter	52
3.2	System Model	53
3.2.1	System Model	54
3.2.2	Design Goals	54
3.2.3	Assumptions	55
3.3	Our Proposed Scheme	56
3.3.1	Construction of Merkle Hash Tree	56
3.3.2	Construction of Cuckoo Filters for Vehicle Authentication . .	56
3.3.3	Construction of Cuckoo Filters for RSU Authentication	58
3.4	V2I Authentication Phase	58
3.5	Performance Analysis and Comparison	59
3.5.1	Security analysis	60
3.5.2	Verification using SPAN and AVISPA tools	61
3.5.3	V2I Authentication Overheads Analysis	62
3.6	Related Works	64
3.7	Summary	66
Chapter 4	Efficient Management of Pseudonyms for Privacy-preserving Au- thentication in VANETs	67
4.1	Related Works	67
4.2	System Model	69
4.2.1	System Model	69
4.2.2	Basic Idea	70
4.2.3	Assumptions	71
4.2.4	Construction of Modified Merkle Patricia Trie for Efficient Man- agement of Vehicle's Pseudonyms	71
4.3	Detailed Description of CREASE	74
4.3.1	Updating the Status of Pseudonym of a vehicle by RSU	76
4.4	Performance Evaluation	78
4.4.1	Security Analysis	80
4.4.2	Formal proof of correctness of CREASE based on BAN logic .	82
4.4.3	Automated verification of CREASE based on SPAN ad AVISPA tools	87
4.4.4	Comparison with other related protocols	88
4.5	Summary	91

Chapter 5 Conclusion and Future Works	93
5.1 Summary of the Dissertation	93
5.2 Future Works	94
Appendix	95
Appendix A: Inference Rules for BAN Logic	95
Bibliography	97
Vita	109

LIST OF TABLES

1.1	Brief summary of the protocols that use ID-based signatures and/or group signatures for authentication of messages	8
1.2	Brief summary of the protocols that use ID-based signatures and/or group signatures for authentication of messages continued	9
1.3	Summary of the protocols that use RSUs for authentication and/or key distribution	11
1.4	Summary of protocols that make use of smart-cards and tamper-proof devices	13
1.5	Summary of the protocols that address the overhead involved in PKI based protocols	15
1.6	Summary of the protocols that address the overhead involved in PKI based protocols continued	16
1.7	Brief summary of protocols that support message aggregation, cooperative message authentication, and/or batch verification	19
1.8	Brief summary of protocols that support message aggregation, cooperative message authentication, and/or batch verification continued	20
2.1	Information of Malicious Vehicles stored in MMPT	28
2.2	Description of Notations used in this Chapter 2	32
2.3	Authentication Information	34
2.4	BAN logic notation	42
3.1	Notation and Description of abbreviation used in this Chapter 3	55
3.2	Authentication Information	57
3.3	Possible Cases and Results with Cuckoo Filter	60
3.4	V2I Authentication Overheads	63
4.1	Notation and Description of abbreviations used in this Chapter 4	70
4.2	Pseudonyms and Current Status	72
4.3	BAN logic notation	82
4.4	Comparison of Security Features	89

LIST OF FIGURES

1.1	VANET Scenarios	2
1.2	Securing VANETs using a central Trusted Authority (TA).	4
2.1	Merkle Hash Tree	26
2.2	Prefix Tree	27
2.3	Modified Merkle Patricia Trie storing the contents of Table 2.1	28
2.4	Proposed VANET architecture for efficient authentication of RSUs and vehicles	30
2.5	Merkle Hash Tree of public keys of RSUs	34
2.6	Mutual Authentication between an RSU and a newly registered vehicle V	36
2.7	Mutual Authentication between an RSU and a vehicle when the vehicle moves from the region of RSU_i to the region of RSU_j where RSU_i and RSU_j are under the vehicle's home RTA.	37
2.8	Message sequence chart of the proposed scheme using SPAN and AVISPA tools	46
2.9	Message sequence chart of our scheme in the presence of an intruder	46
2.10	Vehicle Authentication Overhead Comparison	48
2.11	RSU Authentication Overhead comparison	48
2.12	Revocation message format that RTA sends to RSUs	49
2.13	Comparison of communication overhead on RSU side	50
3.1	(a) A cuckoo filter (b) Insertion operation.	52
3.2	Proposed system model for V2I authentication in VANET.	53
3.3	A sample Merkle Hash Tree of pseudonyms of a vehicle.	56
3.4	Message sequence chart of our scheme using SPAN and AVISPA tools.	61
3.5	Message sequence chart of our scheme in the presence of an intruder.	61
3.6	Computation overhead on RSU for authentication.	63
3.7	Computation overhead on vehicle for authentication.	64
4.1	Proposed VANET architecture for CREASE	69
4.2	Modified Merkle Patricia Trie for storing the contents of Table 4.2	72
4.3	MMPT combined with MHT for storing pseudonyms of vehicles	73
4.4	Message sequence chart of our scheme generated by SPAN and AVISPA tool	87
4.5	Message sequence chart of our scheme in the presence of intruder generated by SPAN and AVISPA tool	87
4.6	RSU Authentication Overhead Comparison	90
4.7	Signature Verification Overhead Comparison	91

Chapter 1 Introduction

People's desire for using connected devices in day to day life has resulted in enormous growth in Internet of Things (IoT). It is estimated that the number of IoT devices will exceed 60 billion by 2025. Internet of Vehicles (IoVs) is likely to make key contribution to such rapid growth. With the support of roadside infrastructure, IoVs help drivers, pedestrians and other vehicles in using the data created by vehicular Ad Hoc Networks (VANETs). Moreover, VANETs have brought a wide range of applications in Intelligent Transportation Systems (ITS) through vehicle to vehicle (V2V), vehicle to infrastructure (V2I), and vehicle to everything (V2X) networks. Although including these network systems provide "smart" driving features and autonomy, they also introduce a plethora of attack vectors to otherwise secure vehicles. It happens because of the opening up of many communication ports to communicate with the outside world, such as sensing tire pressure, sensing front collision, receiving emergency safety messages or traffic-related messages, etc. Different vulnerabilities arise due to intruders capability in identifying and targeting high-value vehicles through openly shared credentials such as public cryptographic keys or RF identifiers such as the frequencies of individual vehicles or tire pressure monitoring sensors (TPMS).

To prevent such vulnerabilities, efficient authentication schemes for entities participating in the VANET need to be designed and implemented. Without an efficient authentication framework, attackers could easily connect to VANETs to endanger other drivers. For example, malicious vehicles may broadcast false data about an accident to avert other drivers from using any specific road, thereby creating traffic jams around. It may also pretend to be a Roadside Unit (RSU) or electronic toll booth to steal other driver's sensitive information. While the existing standard IEEE 1609.2 uses Elliptic Curve Digital Signature Algorithm (ECDSA) for signature verification, a significant limitation still lies in the verification of a single ECDSA signature requiring 7 ms of computation time for on-board unit (OBU) hardware [87]. This imbalance between the time needed to process and the time needed to receive gives rise to Denial of Service (DoS) attacks. Therefore, designing an efficient authentication scheme is of great importance in the VANET environment.

The traditional approach to secure VANETs is to deploy Public Key Infrastructure (PKI), which usually use certificate revocation lists (CRLs) to manage revoked certificates. In the PKI based schemes, the Trusted Authority (TA) assigns certificates of public keys for every registered entity in the network. A CRL is a list of all revoked certificates, usually signed and issued by the TA. In a PKI based scheme, when a vehicle receives a message from an unknown entity, it first checks its revocation status against CRL to verify the sender's certificate for message authentication. However, the size of CRLs increases with the number of revoked vehicles which will increase the overhead in the authentication process. Moreover, significant communication overhead is expected to occur to broadcast CRLs to all entities. At the same time, enough storage space is needed for vehicles to store CRLs. Besides, each vehicle broadcasts a safety-related message every 100-300 ms, according to DSRC [49]. In

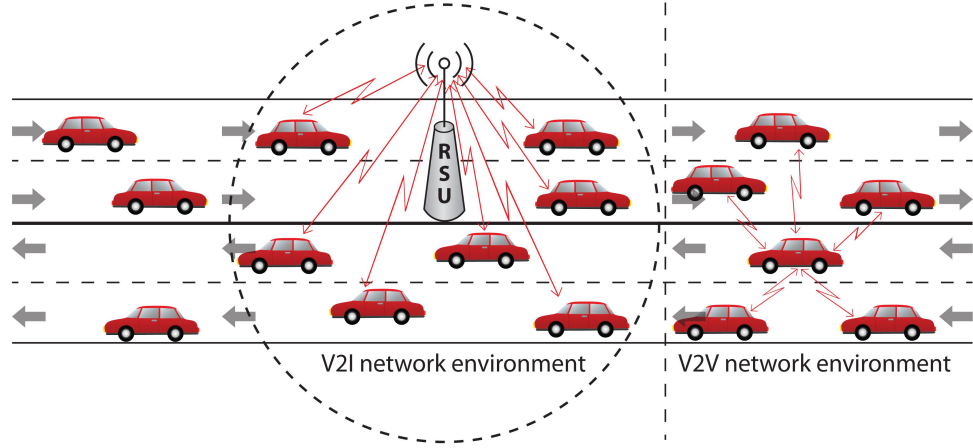


Figure 1.1: VANET Scenarios

such a scenario, each vehicle may receive a large number of messages every 100-300 ms, and it has to check the revocation status against CRL to authenticate the received messages. Therefore, it is necessary to design and implement efficient protocols to ensure privacy, authenticity, and secure message dissemination in VANETs. Next, we discuss the necessary background, motivation, and findings of our dissertation.

1.1 Background

This section introduces the VANET system and presents some of the security and privacy issues in VANET. We also describe different state-of-the-art solutions on the security, privacy, and authentication in VANET.

1.1.1 VANET

A vehicular Ad Hoc Network (VANET) is basically a type of Mobile Ad Hoc Network (MANET) which consists of vehicles as nodes. General model of VANETs proposed in the literature consists of two major components: On Board Unit (OBUs) installed on vehicles and Road Side Units (RSUs) installed on road side to support the infrastructure needed for the deployment of VANETs. Typically OBU utilizes a service that is provided by an application hosted at RSUs. Each vehicle is assumed to be equipped with a set of sensors to collect phenomena surrounding the vehicle and the OBU processes the information collected by the sensors and sends/receives them to/from other relevant vehicles directly or through nearby RSUs [3]. The RSUs may also connect to the Internet to provide the necessary services to vehicles.

A broad range of applications can be enabled by mainly two types of communications: (i) infrastructure-based communications (Vehicle to Infrastructure (V2I) communication) and (ii) direct communications between vehicles (Vehicle to Vehicle

(V2V) communication) [128] (see Fig 1.1). Major efforts for standardizing VANETs have been carried out by the IEEE 802.11 Task Group by defining enhancements to IEEE 802.11 required to support Intelligent Transportation Systems (ITS) applications. This amendment is currently known as IEEE 802.11p. The wireless communication capability between moving vehicles is achieved using dedicated short range communication (DSRC). It is anticipated that DSRC will be used for both vehicle-to-vehicle communications and vehicle-to-infrastructure communications. The spectrum is seen as particularly useful for V2X communications because it can support very low-latency, secure transmissions, fast network acquisition and in general, the ability to handle rapid and frequent hand-overs that are inherent in a vehicular network, as well as being highly robust in adverse weather conditions [105].

1.1.2 Security in VANETs

Security is crucial in VANETs because vehicles in the network can easily be attacked by malicious vehicles due to the dynamic nature of network formation. Some of these attacks may be carried out by nodes inside the network (i.e., nodes that have been already authorized to be a member of a VANET); other attacks may be carried out by vehicles that do not belong to the VANET. Among the existing types of attacks in VANETs, the most common security attacks are :

- Sybil attack: Sometimes a false message by a single malicious vehicle is not convincing. A Sybil attack occurs when a malicious vehicle uses multiple identities in parallel to impersonate a number of vehicles and generate multiple messages. Vehicles use transmitted messages to make decisions, so it is necessary to resist a Sybil attack.
- Message modification attack: An attacker alters the existing messages in this type of attack. They can modify the content of the transmitted messages or broadcast messages for their own benefit.
- Message linking attack: In message linking attacks, an attacker links two different messages sent by the same sender vehicle and tracks the path traversed by the vehicle. They can breach the privacy (e.g., travel history, places of travel) of the vehicle users.
- Denial of Service (DoS) attack: The attacker injects dummy messages into the networks to jam the communication channel. It can affect the VANET's performance as well as traffic safety. An attacker could generate a large number of traffic messages and cause an approaching vehicle not to get the actual warning messages. Therefore, the discrepancy of processing times and receiving times leads to a Denial of Service (DoS) attack.
- Replay attack: In case of a replay attack, attacker eavesdrops a transmitted message and re-transmits it several times to create confusion.

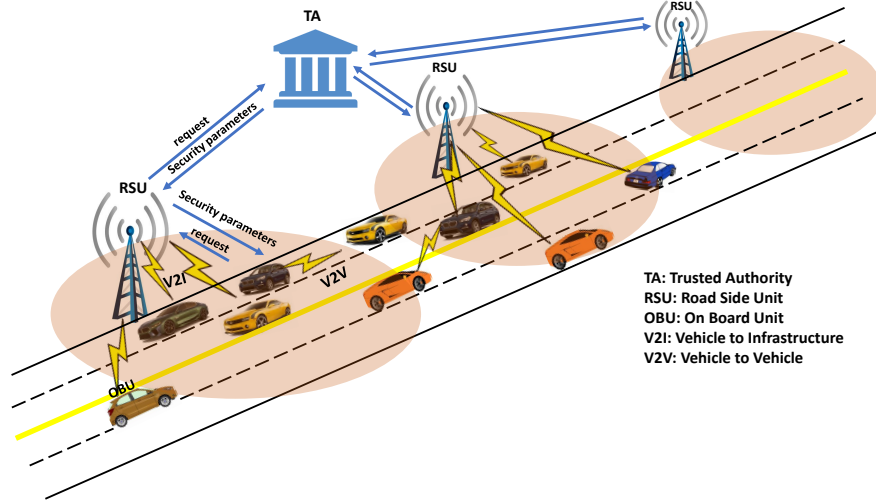


Figure 1.2: Securing VANETs using a central Trusted Authority (TA).

1.1.3 Privacy in VANETs

Privacy refers to the privacy of the vehicles (drivers) and the location of the vehicles. When a vehicle sends a message, no one (except relevant authorities) should be able to determine the identity or location of the vehicle from the messages a vehicle sent. Ensuring privacy of vehicles (drivers) is one of the many challenging issues for which an efficient solution needs to be found because an adversary could otherwise trace a vehicle's traveling routes by capturing and analyzing its messages [71] and identify the vehicle (driver) which may have drastic consequences for the drivers. To address this issue, many researchers have proposed protocols wherein vehicles use pseudonyms instead of their real ids in communication while at the same time enabling authorities to extract the real ids from pseudonyms to trace and punish misbehaving vehicles. Such protocols are called conditional privacy-preserving protocols. Assigning pseudonyms to vehicles and changing them frequently is one of the strategies used to ensure the privacy of vehicles. Although the frequency of such change remains debatable, vehicles must change pseudonyms more frequently to maximize privacy. Factors such as availability and storage size play an important role in determining the rate at which pseudonyms should be changed.

1.1.4 Authentication in VANETs

Although the potential benefit of VANETs is increasing, malicious attackers can intercept, modify, and replay transmitted messages due to the wireless nature of V2V and V2I communication. Therefore, a vehicle user needs to check the authenticity and integrity of a received message as well as the legitimacy of the sender vehicle to accept the message. Authentication needs to be achieved at two levels – first at the node level, referred to as node authentication, and second at the message level, referred to as message authentication. All messages sent by a vehicle should be authenticated before being processed. The basic principle of message authentication can be simplified

as signing a message by the sender and then verifying the authenticity and integrity of the message at the receiver's end. Certain authentication requirements such as low computational overhead, robust and scalable authentication, efficient certificate revocation must be addressed and solved to ensure secure communication in VANET.

A vast majority of the papers in the literature addressing security, authentication, and privacy use a TA for obtaining and loading OBUs and RSUs with security parameters such as keys, certificates, and pseudonyms (illustrated in Fig 1.2).

1.1.5 Privacy-preserving, Authentication, and Secure Message Dissemination in VANETs

In this section, we group the protocols [71] addressing privacy, authentication, and secure message dissemination in VANETs into different classes and discuss the benefits and drawbacks of the protocols in each class.

Protocols that use ID-based signatures and group signatures for authentication of messages:

Generally, the TA is responsible for issuing security parameters, such as keys, certificates, and pseudonyms to vehicles. When the TA detects (or is informed by an RSU) a malicious vehicle, it revokes the vehicle's certificates (generally, one certificate for each pseudonym) and informs all other vehicles about it. This is a centralized approach which does not scale well. Moreover, as the CRL grows, the message authentication overhead increases. In this subsection, we discuss some solutions proposed for solving these problems using ID-based signatures and ID-based cryptography [8, 90, 17].

Zhang et al. [136] introduced an on-the-fly group creation approach in which the RSUs create and maintain groups. This allows vehicles to join the group maintained by the nearby RSU and also anonymously broadcast authenticated messages to vehicles within its group. However, authenticated message dissemination among vehicles in different groups is not addressed. Their approach is conditional privacy-preserving. In the decentralized group authentication protocol presented by Zhang et al. [135], RSUs are responsible for maintaining and managing the group of vehicles within its transmission range for supporting secure communication between them. The basic idea behind their scheme is as follows: the TA uses bilinear pairing for generating keys and issuing certificates to vehicles and RSUs. The TA also maintains the CRL. A Tracing Manager (TM) is responsible for tracing malicious vehicles. When a vehicle passes a nearby RSU, it uses signcryption [139] to send an encrypted request to the RSU for a group key. After receiving the group key, it uses the group signature scheme [38] to sign and send messages to members in its group. However, authenticated message dissemination between vehicles in different groups is not addressed. Xiong et al. [123] propose a scheme for managing communication among a group of vehicles effectively and spontaneously. Their scheme is based on revocable ring signatures proposed by Liu et al. [65]. This scheme allows only valid ring members to generate a ring signature for a message. In addition, trusted authorities are responsible for tracing and revoking the real signer. However, message verification overhead

increases when the number of vehicles in the group grows. Biswas et al. [14] present a scheme for authenticating safety messages broadcasted by RSUs. Their scheme is also based on ID-based signatures [8, 90] and uses proxy signatures based on Elliptic Curve Digital Signature Algorithm (ECDSA), the digital signature algorithm specified in IEEE 1609.2 standard [1] for message authentication. They compare the overhead incurred by their algorithm in signing and verification with that of a few other existing algorithms. Among the five algorithms compared, their algorithm is the only one which uses both ID-based and proxy-based signature schemes and yields comparable performance.

Chim et al. [27] propose a software based Secure and Privacy Enhancing Communication Scheme (SPECS) which relies on ID-Based Cryptography (IBC) with bilinear pairing. In this scheme, after an initial handshaking with the nearby RSU, vehicles belonging to the same group can communicate securely without the aid of the RSU. They make use of two Bloom filters [15], namely, positive and negative filters to reduce the message overhead and false positives during message authentication. Positive filter stores the authentic vehicle's hash value of pseudonym and messages, and the negative filter stores the hash value of pseudonym and messages of vehicles that have not been authenticated. It has low communication overhead and it also has an effective batch verification success rate. However, it can be vulnerable to impersonation attack.

Hsiao et al. [48] present two broadcast authentication schemes (FastAuth and SelAuth) to deal with the signature flooding problem (i.e., reduce the computation overhead involved in verifying a large number of signatures in a short amount of time). The FastAuth protocol is based on chained Huffman hash trees (a data structure designed by them) for securing periodic single-hop beacon messages. This scheme supports a one-time signature scheme whose signature verification is claimed to be 50 times faster and signature generation is claimed to be 20 times faster than using Elliptic Curve Digital Signature Algorithm (ECDSA), the Digital Signature Algorithm specified in IEEE 1609.2 standard [1] for authentication. The other protocol, namely, the SelAuth protocol, helps in isolating malicious nodes faster by selecting messages that need to be verified before forwarding. They use a selection algorithm to distinguish benign neighbors from malicious neighbors which helps in restricting the spread of messages with invalid signatures to a small area. They also show that SelAuth incurs 10% - 35% additional computational overhead compared to other closely related schemes while containing 99% of invalid signatures to one hop. They only focus on broadcast authentication and not point-to-point message authentication.

Wasef and Shen [117, 118] try to reduce the time involved in checking the CRLs during message authentication; they use the keyed Hash Message Authentication Code (HMAC), wherein the key used to calculate the HMAC is shared only between non-revoked OBUs. However, vehicles must still verify the validity of certificate and signature because it still uses a TA for generating and distributing secret keys and certificates to all OBUs. Certificate revocation is triggered by the TA which involves revoking the current secret key and securely distributing a new secret key to all non-revoked OBUs.

The dual authentication and key management technique presented by Vijayaku-

mar et al. [107] is based on Chinese Remainder Theorem (CRT) where both hash code and fingerprints of each participating vehicle are used for dual authentication. In their approach, the TA divides the users into two groups, namely Primary and Secondary, and then generates two different group keys for these two different groups of users. It provides service to vehicles' users on the basis of a Service Level Agreement (SLA). The shared group keys are refreshed when a new user joins the group or an existing group member leaves the group, thus making this scheme resistant to forward secrecy and backward secrecy attack. It is shown that this scheme is computationally more efficient compared to some of the other existing schemes, such as Chinese Remainder Group Key (CRGK) [138] and Key-tree Chinese Remainder Theorem (KCRT) [140]. However, they do not address the privacy of users in their work.

Zhang et al. [133] present a conditional privacy-preserving authentication protocol based on ID-based aggregate signatures and bilinear pairing based cryptography. Their approach allows hierarchical aggregation of signatures and batch verification. Their hierarchical aggregation technique allows re-aggregation which reduces transmission and storage overhead. Moreover, it has lower waiting time for aggregation compared to some of the other approaches presented in the literature.

Lai et al. [58] discuss the security challenges, requirements and benefits of group communication in Software Defined Network (SDN) based 5G-VANETs. They propose a Secure Group Mobility management Framework (SGMF) for group-oriented vehicular communication based on modified IPsec packet and an addressing method described in [56]. Their scheme performs better compared to some of the existing mobility management schemes with respect to hand over signaling overhead and latency. However, the hand over signaling cost may increase as the density and mobility of vehicles increase.

Cui et al. [29] propose a Secure Privacy-preserving Authentication scheme using Cuckoo Filter (SPACF). Their goal is to achieve higher success rate than some of the previously proposed schemes in the batch verification phase. Cuckoo filter and binary search are used to accomplish their goal. SPACF is shown to be more efficient than some of the previous schemes because it is pairing free and does not use map-to-point hash functions. However, this ID-based scheme still suffers from inherent key escrow problem despite eliminating much of the limitations of Public Key Infrastructure (PKI) and ID-based Batch Verification (IBV).

Table 1.1 and 1.2 summarize the strengths and weaknesses of the protocols discussed in this section.

Protocols that use RSUs for Authentication and/or Key Distribution

Some protocols presented in the literature, offload some work (such as message authentication, packet forwarding) from vehicles to RSUs and/or some work (such as key management and CRL distribution, detecting and reporting suspicious vehicles) from TA to RSUs. In this subsection we discuss protocols belonging to this category.

The RSU-aided message authentication scheme, called RAISE, proposed by Zhang et al. [129] offloads the overhead involved in message authentication to RSUs. This requires dense deployment of RSUs. Vehicles establish a shared key with the RSU

Table 1.1: Brief summary of the protocols that use ID-based signatures and/or group signatures for authentication of messages

Paper	Focus area(s)	Method(s) used	Strengths and Weaknesses
Zhang et al. [135]	Authentication, privacy, traceability, and confidentiality	Bilinear pairing, group signature [38], and signcryption [139]	RSUs are responsible for maintaining groups, so decentralized in some sense; no scalable mechanism to support broadcast throughout the network; group-signatures generally have high signature verification and revocation costs.
Xiong et al. [123]	Secure V2V communication	Bilinear pairing and Revocable ring signatures [65]	Does not require ubiquitous deployment of RSUs; message verification cost may increase as the number of vehicles grows.
Biswas et al. [14]	Authentication	ID and Proxy-based signature scheme	Lower overhead compared to some compared algorithms; addresses only authentication of RSU messages.
Chim et al. [27]	Authentication, security, and privacy	Identity Based Cryptography (IBC) with bilinear pairing	Low overhead and authenticates messages effectively; can be vulnerable to impersonation attack.
Hsiao et al. [48]	Broadcast authentication	Chained Huffman hash trees (based on Merkle hash tree and Huffman tree)	More efficient than ECDSA specified in IEEE 1609.2 standard; the protocol for authenticating beacons will not work correctly if beacons are missed.
Vijayakumar et al. [107]	Secure data transmission in VANETs	Vehicular Public Key Infrastructure (VPKI) and dual authentication and key management techniques	Provides resistance against forward secrecy and backward secrecy attacks; takes single broadcast message to get the updated group key; does not address location privacy.

Table 1.2: Brief summary of the protocols that use ID-based signatures and/or group signatures for authentication of messages continued ...

Zhang et al. [134]	Privacy, security, and authentication	Bilinear pairing based cryptography; multiple trusted authorities; ID-based aggregate signature technique for authentication	Certificate distribution is not centralized; Bilinear pairing based cryptography generally has high computational overhead.
Wasef and Shen [117, 118]	Fast message authentication	Bilinear pairing	Claims to make the CRL checking process faster; High overheads involved in distributing a secret key to all non-revoked OBUs.
Lai et al. [58]	Secure group communication in SDN based 5G-VANETs	PKI; secure group management and group handover	Provides better group hand over authentication in terms of hand over signaling overhead and latency; cost may increase with increase in density and mobility of vehicles.
Cui et al. [29]	Privacy, security, and Authentication	Cuckoo filter and binary search methods	It is pairing free and does not use map-to-point hash functions; suffers from inherent key escrow problem.

using Diffie-Hellman algorithm. They also take the k-anonymity [102] approach to prevent an adversary from associating a message with a particular vehicle to ensure the privacy of the vehicles.

The message authentication scheme proposed by Zhang et al. [130] is an extension of the scheme presented in [129]; this extension includes a method for vehicles to cooperatively authenticate messages in the absence of an RSU. Hao et al. [46, 45] present a distributed key management framework and also a method for cooperative message authentication for speeding up message authentication. Sun et al. [97] also present a group signature and identity-based signature scheme for secure and authenticated message dissemination. Papadimitratos et al. [81] also present a distributed method for distributing CRLs using RSUs to reduce the overhead involved in CRL distribution.

Lu et al. [68] propose a Social-based PRivacy-preserving packet forwardING protocol which prevents packet analysis attack, packet tracing attack, black hole attack and grey hole attack in vehicular Delay Tolerant Networks (DTNs). This protocol relies on placing RSUs at high social intersections and using group signatures to prevent the disclosure of identity of senders, target vehicles and relaying vehicles. The

RSUs help in forwarding packets between vehicles which helps in reducing packet loss.

Shim's [94] Conditional Privacy-preserving Authentication Scheme (CPAS), is a secure conditional privacy-preserving scheme for V2I communications. It uses bilinear pairing based cryptography to generate and store key parameters and ID-based signatures for authentication. Their scheme requires RSUs to verify messages sent by vehicles in batches to speed up the message authentication process. They do not address V2V communications.

The Logical Key Hierarchy (LKH) based schemes [47, 93, 22] and Topology Matching Key Management (TMKM) based schemes [99, 100, 101] for Group Key Management (GKM) wherein all the key management functionalities are handled by the Key Distribution Center (KDC) have re-keying overhead. Park et al. [82] address this problem and propose a Group Key Management (GKM) scheme, called RSU-based Decentralized Key Management (RDKM). RDKM is based on versakey framework [23] for secure vehicular multicast communication. In this scheme, part of the GKM functions are offloaded to RSUs in a distributed manner. For efficient operation of this protocol, the authors suggest placing RSUs at the intersection of streets. For forming groups, the authors suggest placing vehicles within the region of an RSU in the same group. This helps an RSU manage the group keys efficiently. Their performance evaluation shows that this approach results in approximately 60% to 80% reduction in communication overhead compared to some of the existing GKM-based schemes. They also propose a new performance measure namely, Group Key Management Overhead (GKMO), and observe a rapid increase in GKMO for both LKH and TMKM schemes compared to the RDKM scheme. However, RDKM requires more storage space to store information about keys at each vehicle compared to the LKH and TMKM schemes.

In a Sybil attack, a malicious node can use multiple identities and inject false messages into the network. Zhou et al. [141] propose a protocol, called Privacy Preserving Detection of Abuses of Pseudonyms (P^2 DAP), to detect Sybil attacks. In their scheme, the Department of Motor Vehicles is used as the TA to provide a pool of pseudonyms to each vehicle and releases part of its workload to RSUs as follows. Two-level hashing of every pseudonym is generated where the key of the first-level hash is known to the RSUs to identify whether the pseudonyms belong to the same group of vehicles. The second-level hash key is known only to the TA to map each pseudonym to an individual vehicle. Each time an RSU finds suspicious pseudonyms, it reports this incident to the TA for verification. But the generation and management of a large number of pseudonyms can be costly.

The authentication and key establishment scheme for V2V and V2I communications, presented by Li et al. [59], is also based on ID-based public-key cryptography, blind signatures [70, 30], and one-way hash chain. The blind signature scheme used in their scheme allows vehicles to communicate with the RSUs to access the services provided by them without revealing their real identities, location, and so on. They use TA for populating the OBUs with the necessary secret key, group key and pseudo id offline or through a secure secret channel. The methods used are not simple and moreover using a centralized TA is not scalable.

Table 1.3: Summary of the protocols that use RSUs for authentication and/or key distribution

Paper	Focus area(s)	Method(s) used	Strengths and Weaknesses
Li et al. [59]	Location privacy and authentication	ID-based public-key cryptography, blind signature, and one-way hash chain	Solves location privacy, anonymity problem; uses a central trusted third party, which is not scalable.
Zhang et al. [129, 130]	Privacy, security, and authentication	RSU-aided message authentication, cooperative message authentication, Diffie-Hellman algorithm and k-anonymity	Offloads the overhead involved in message authentication to RSUs; low communication overhead. Diffie-Hellman protocol is prone to man-in-the middle attack; vehicles still need to be pre-loaded with public keys.
Lin et al. [64]	Privacy, security and authentication	Uses TA to get (public, private) keys; TESLA [84] hash chains for message authentication	Aims to reduce the overhead involved in certificate generation and distribution.
Lu et al. [68]	Secure packet forwarding in vehicular DTNs and privacy	RSU assisted packet forwarding; Bilinear pairing	Provides high packet delivery ratio, preserves conditional privacy and resists packet tracing attack, packet analysis attack, and black (grey) hole attacks; ignores mobility of vehicles and fluctuations in traffic.
Park et al. [82]	Distributed key management	PKI and RSU-based key management	Reduces re-keying overhead; can have high storage overhead to store a large number of keys.
Zhou et al. [141]	Privacy and security; Sybil Attack Detection	Distributed passive overhearing by RSUs; PKI based pseudonym assignment	Detects Sybil attacks with low overhead and delay.

The secure privacy-preserving protocol presented by Lin et al. [64] aims to reduce the overhead related to signing and verifying packets based on public key cryptography. They propose attaching a short message authentication code tag with each packet instead of a signature. As in the TESLA protocol [84], each vehicle generates a hash chain h_1, h_2, \dots, h_n from a random seed S ; here, $h_n = S$, and $h_i = H^{j-i}(h_j)$ for $i < j$, where H is a hash function. Each element in the hash chain is used as key to generate MAC codes for several packets and the keys are released after a short delay δ (as in [84]) for the receiver to authenticate the packet.

Table 1.3 summarizes the strengths and weaknesses of the protocols discussed in this section.

Protocols based on Smart Cards and Tamper-proof Devices

Conventional PKI [88] based schemes require each vehicle to verify the signatures of each of the other vehicles sending messages to it; this results in computational overhead for the OBUs of the vehicles. To overcome this drawback of PKI based approach, Zhang et al. [131] proposed an ID-based Batch Verification (IBV) scheme. Under IBV scheme, an RSU can verify the signatures of multiple messages all at once; so signature verification is more efficient under this approach. The authors use ID-based cryptography for generating private keys associated with pseudo-identities. However, the IBV [131] scheme depends on the availability of a tamper-proof hardware device on each vehicle to securely store the system-wide secret key. Since the system wide secret key is stored on tamper-proof hardware of each vehicle, if one of these devices is compromised, the whole system is compromised. Moreover, this does not ensure privacy of vehicles because real ID of a vehicle could be traced by other vehicles.

The authentication protocol presented by Ying and Nayak [126, 125] uses dynamic login IDs to preserve privacy. The user gets a smart card loaded with the vehicle's pseudonym and password. Smart card inserted into the vehicle's OBU, authenticates its owner by asking for the real ID and password and generates dynamic login identity for the user and sends it to the nearby RSU. Upon receiving this message, the RSU verifies if it is valid, computes its own dynamic login id and sends its dynamic login id and the dynamic login id of the vehicle to the TA. The TA computes the anonymous keys and the corresponding certificates for the vehicle and sends them to the RSU securely. The RSU then broadcasts the keys and certificates securely to the vehicles in the region and the corresponding vehicles receive them and use them for communication. Their privacy-preserving anonymous authentication scheme not only authenticates received messages but also verifies the legitimacy of the senders of the messages (i.e., it checks if the sender is a malicious node which forged the ID of some legitimate node). In addition, to reduce the computational complexity, they do not use bilinear pairing based cryptography. It allows the user's password to be changed dynamically. So, this scheme can resist smart card loss attack, impersonation attack, and password guessing attack.

The protocol presented by Paruchuri and Duresi [83] also uses smart cards to generate anonymous keys on-the-fly for establishing secure V2V as well as V2I com-

Table 1.4: Summary of protocols that make use of smart-cards and tamper-proof devices

Paper	Focus area(s)	Method(s) used	Strengths and weaknesses
Shim's [94]	Privacy and authenticated V2I communication	ID-based cryptography; bilinear pairing based cryptography for key generation	Fast batch verification of messages at the RSUs; vehicles need to be equipped with Tamper-Proof Devices (TPDs); TPDs could be susceptible to side-channel attacks.
Paruchuri and Durresi [83]	Authentication, privacy, and security	Smart cards to store keys and perform encryption/decryption	Requires the use of smart cards.
Ying and Nayak [126]	Privacy, security, and authentication	Login ids are generated dynamically for ensuring privacy	Smart cards are used for generating login ids dynamically; it can resist password attacks, and impersonation attacks; can tolerate smart card loss; can handle compromised RSUs.
Ying and Nayak [125]	Privacy, security and authentication	Diffie-Helman protocol; smart cards; hash functions; centralized trusted authority for loading smart-cards with login id and password	Can resist smart card loss attack; can also resist impersonation and password guessing attack.
Zhang et al. [131]	Authentication, batch verification	ID-based cryptography; Camenisch-Lysyanskaya (CL) signature; tamper-proof device	The real ID of the vehicles could be tracked; vulnerable to impersonation attack.

munication. The TA issues smart cards as well as the keys to the vehicles and certificates to RSUs. The vehicle's ID, required cryptographic keys, and driver information are stored on the smart card. To send a message to vehicles within its group, first a vehicle needs to get a session key securely from the nearby RSU. To send a message m to a vehicle within its group, it encrypts the message m and the ID of the OBU and the signature of m (hash of m encrypted with its private key V_{Pr}) encrypted with the public key $E_{RSU_{P_u}}$ of the RSU using the session key K_e as follows and sends it:

$$E_{K_e}(m, E_{RSU_{P_u}}(OBU_{ID}, E_{V_{Pr}}(H(m))))$$

A receiving vehicle can decrypt the message using the session key issued by the RSU. However, it cannot decrypt the second part because the private Key of the RSU is needed for decrypting the second part. The second part is used by the RSU to trace misbehaving nodes, when necessary. Table 1.4 summarizes the strengths and weaknesses of protocols discussed in this section.

Next, we discuss some of the protocols that minimize the overhead involved in using the Public Key Infrastructure (PKI) for generating and assigning keys, pseudonyms, certificates, and CRLs to vehicles and/or RSUs.

Minimizing the Overhead involved in Public Key Infrastructure (PKI) based Protocols

The anonymous authentication protocol presented by Wang et al. [110] uses the TA to assign each vehicle and each RSU a long term certificate during registration. Each RSU is responsible for assigning a master key to each vehicle entering its region after authenticating the vehicle based on its long term certificate. Then the vehicle uses the master key to generate pseudonyms locally and uses them to sign messages to preserve anonymity. This approach has lower signature verification overhead compared to the protocols presented in [118] and [53]. Moreover, it supports both single and batch authentication of messages.

The Secure and Authenticated Key Management Protocol (SA-KMP) presented by Hengchuan et al. [103], combines the idea of the Public Key Regime (PKR) (which delegates the distribution of public keys to the RSUs, eliminating the need to distribute digital certificates) proposed by Shen et al. [92] and the idea of 3-D matrix key distribution scheme (which generates the keys dynamically instead of preloading the keys), proposed by Hamid et al. [42, 43]. Wasef et al. [116] propose a mechanism based on PKI which supports not only location privacy but also authentication; it also uses a distributed approach for certificate revocation. They use the Message Authentication Acceleration (MAAC) [117] protocol to make the revocation checking process faster without checking the CRLs. However, their solution can only preserve the location privacy of vehicles within its group. They also propose a method for mitigating Denial of Service (DoS) attacks.

Biswas and Misisic [12],[11],[13] use proxy signatures for privacy-preserving authentication. One drawback of this solution is that it requires larger keys for generating and verifying signatures. As a result, it incurs higher computational cost compared

Table 1.5: Summary of the protocols that address the overhead involved in PKI based protocols

Paper	Focus area(s)	Method(s) used	Strengths and weaknesses
Wasef et al. [116]	Location privacy, authentication, and certificate revocation	PKI and Message Authentication Acceleration (MAAC) protocol	Fast revocation check process and mitigates DoS attacks; location privacy may not be ensured against outsider attacks.
Biswas and Misic [12, 11, 13]	Self authentication and anonymous message delivery	PKI, proxy signatures	Preserves message integrity and anonymity; RSU assisted proxy signatures.
Dong et al. [34]	Privacy-preserving data forwarding; specifically designed for service oriented VANETs	Lite-TA-based public key cryptography; on-path onion encryption scheme	Efficient, robust, and ensures higher trust level; high computational overhead.
Haas et al. [41]	Distribution of CRLs	PKI and Bloom filters [15]	Does not require ubiquitous deployment of RSUs; false positives can be prevented; computational overhead is somewhat low.
Shen et al. [91]	Secure communication	Chameleon hash signature [54]	Ensures unlinkability, traceability, and defense against replay attack; V2V authentication is not addressed.
Whyte et al. [119]	Security credential management system; V2V communication	Public Key Infrastructure	The authors try to achieve a tradeoff between privacy, security, and efficiency; decentralized certificate distribution; frequent certificate changes could cause high overhead.
Alshaer [5]	Securing VANET connectivity with the support of RSUs	Vehicular Public Key Infrastructure (VPKI)	Can predict uplink and downlink connectivity probabilities in VANETs; assumes RSUs are trustworthy.

Table 1.6: Summary of the protocols that address the overhead involved in PKI based protocols continued ...

Hengchuan et al. [103]	Authenticated key management	Public key regime (PKR) [92]; the 3-D matrix key distribution scheme [42, 43]	This approach is more scalable than PKI based approaches; key generation takes less time compared to Elliptic Curve Diffie-Hellman and Diffie-Hellman protocols.
Wang et al. [110]	Privacy, security, and authentication	Does not use PKI for generating pseudonyms and the related certificates to vehicles; vehicles generate their own pseudonyms	This approach has less signature verification overhead compared to the protocols presented in [118] and [53]; it supports both single and batch authentication.

to other competitive schemes such as Elliptic Curve Cryptosystems (ECC) to provide similar security strength.

Dong et al. [34] propose a privacy-preserving data forwarding scheme for service oriented VANETs based on Lite-TA-based public key cryptography and on-path onion encryption scheme. This scheme has lower encryption cost and public key management complexity compared to conventional public key encryption schemes. However, since this approach requires relaying nodes to encrypt the message before forwarding to prevent adversaries from tracing message flows, it incurs higher computation overhead for forwarding packets.

Haas et al. [41] propose a method for quick, organized and efficient distribution of CRLs through V2V communication [55]. This scheme ensures backward privacy of revoked vehicles prior to their revocation. They use the probabilistic data structure Bloom filters [15] for quickly checking CRLs. However, false positives may occur. But they claim that, false positives can be avoided by discarding the certificate of the vehicles that may trigger a false positive. The use of Bloom filters reduces the overhead incurred for checking CRLs. It is observed that the distribution of CRLs through V2V communication is more efficient and cost effective than the RSU-based distribution scheme because it does not require widespread deployment of RSUs.

Many of the research works based on chameleon hash signature [54] using fixed public keys for authentication do not guarantee message unlinkability. Shen et al. [91] address this problem and propose a light weight privacy-preserving protocol that relies on Elliptic Curve based chameleon hash signature and dynamic public keys. They consider the registration phase and the mutual authentication phase between OBUs and RSUs in their protocol. They also considered the TA tracking phase to ensure authenticity and traceability. Whenever any suspicious event occurs, the TA can

recover the real identity of the OBU that created the event by executing the TA tracking phase. The use of chameleon hash-based signature for messages helps in preventing replay attacks and impersonation attacks. However, V2V authentication is not addressed in this protocol.

The Security Credential Management System (SCMS) proposed by Whyte et al. [119] is based on PKI; it was developed under a cooperative agreement with the United States Department of Transportation (USDOT), the leading candidate for V2V security backend design in the United States. This scheme adds some additional features such as the number of vehicles it supports and tries to achieve a tradeoff between security, privacy and efficiency of traditional PKI based approaches. Additionally, they propose (i) a frequent certificate changing (e.g., every 5 minutes) scheme to enhance protection against attackers outside of the SCMS and (ii) organizational separation of operations of SCMS to protect against attackers inside the SCMS.

Alshaer [5] proposed a secure connection model based on the Vehicular Public Key Infrastructure (VPKI) that utilizes trusted RSUs to establish secure connections and distribute secret keys to vehicles within their transmission range. The probability of the number of reachable neighboring vehicles that a Communication Enabled Vehicle (CEV) can reach has been derived using Exponential distribution of time and space headways with a Robustness Factor (EwRF). They claim that suitable statistical distribution (e.g., exponential distribution, Generalized Extreme Value (GEV) distribution) that characterizes inter-vehicle spacing can accurately contribute to secure connectivity. This approach requires the widespread deployment of RSUs and RSUs are assumed to be reliable. Table 1.5 and Table 1.6 summarize the strengths and weaknesses of the protocols discussed in this section.

Message Aggregation and Cooperative Message Authentication

Multiple vehicles could observe the same phenomena on the road and try to disseminate it to other vehicles which wastes bandwidth. To address this problem, message aggregation (as has been proposed for sensor networks earlier) has been proposed. Moreover, to reduce the overhead involved in message authentication, cooperative message authentication wherein vehicles share the overhead due to message authentication, has been proposed. We discuss the protocols in these categories in this subsection.

Protocols using Message Aggregation

The Aggregated Emergency Message Authentication (AEMA) scheme proposed by Zhu et al. [40] is based on bilinear pairing. Under AEMA, each vehicle registers with the TA (they call it an Offline Security Manager (OSM)) and obtains its public key certificate. Then, when a vehicle needs to send an emergency message, it uses the following format (*Type*, *Loc*, *ID*, *Time*, *Sig*, *Cert*) to send it. Here, *Type* indicates the type of the event, *Loc* is the location where the event occurred, *ID* is the pseudo ID of the vehicle, *Time* is the time when the event occurred, *Sig* denotes the signature of the

message, and $Cert$ is the certificate. The receiver verifies the validity of the certificate $Cert$ and the signature Sig and accepts the message. The authors assume that each event is uniquely determined by $Type$, Loc , and $Time$. Hence, an intermediate node receiving the message can eliminate duplicates and aggregate the messages. The overhead involved in computing the signature based on bilinear pairing is of some concern. In addition, the algorithm depends on the central OSM for issuing certificates. The authors assume that each observed event has a unique type. This scheme does not ensure location privacy of vehicles because each message carries the location information of vehicles.

Dietzel et al. [32] proposed selective attestation and trust fusion to detect attacks as well as mitigate their effects for semantic aggregation in VANETs. Their approach is based on a generic data aggregation model, which makes it extensible and suitable for the existing data aggregation schemes. In the trust fusion mechanism, multiple warnings of the same event are linked to alleviate the need for a Global Unique Identifier system (GUID) by using a fuzzy logic technique. However, the bandwidth needed for selective attestation could slow down the message dissemination process. Many of the existing message aggregation techniques require roads to be segmented into small fixed-size regions for aggregating messages originating from these regions. However, messages originating across regions cannot be aggregated using these approaches. Van der Heijden et al. [106] address this problem and present a scheme that allows more dynamic aggregation of messages.

Next, we discuss protocols in which vehicles cooperate to authenticate messages in order to reduce the overhead involved in message authentication.

Protocols that use Cooperative Message Authentication or Batch Verification

Hao et al.'s [44] distributed key management and Co-operative Message Authentication Protocol (CMAP) based on short group signature [16] can detect compromised RSUs and the malicious vehicles colluding with them. Vehicles getting keys from the same RSU form a group. To ensure reliable key distribution, messages are encrypted using Elliptic Curve Integrated Encryption Scheme (ECIES) and are transmitted using the Transmission Control Protocol (TCP). This scheme allows the cooperative verifiers to cooperatively authenticate messages. Cooperative verifiers are selected dynamically and distributively based on their own geographic locations relative to the sender of the message. However, a malicious vehicle can pretend to be a cooperative verifier by creating many Sybil nodes within its transmission range, which makes this scheme vulnerable to Sybil attack.

Most of the research work on secure incentive schemes focus only on cooperative packet forwarding; but due to the high mobility of vehicles, packets could be lost. To address this problem, Lai et al. [57] propose a Secure Incentive scheme for Reliable Co-operative downloading in highway VANETs (SIRC) that uses two phases, namely, cooperative downloading and cooperative forwarding which encourage vehicles to cooperate through an incentive scheme; SIRC utilizes aggregated Camenisch-Lysyanskaya (CL) signature [21] to cooperate with others in securely downloading-and-forwarding

packets. In this scheme, a reputation system is implemented to reward the cooperating vehicles and punish the malicious vehicles. In addition, a partial prepayment strategy is used to minimize the payment risk to client vehicles. This scheme can resist various attacks such as free riding attack, DoS attacks and packet injection/removing attack. The performance evaluation of SIRC shows that it has high download success rate, low download delay, and moderate computation and communication overhead. A disadvantage of this approach is that the reputation information about vehicles which have high variability in their spatial distribution need to be calculated and stored.

Table 1.7: Brief summary of protocols that support message aggregation, cooperative message authentication, and/or batch verification

Paper	Focus area(s)	Method(s) used	Strengths and weaknesses
Zhu et al. [40]	Emergency message authentication	Bilinear pairing; message aggregation	Does not ensure location privacy; useful for propagation of short emergency messages only.
Jiang et al. [51]	Privacy, security, and authentication	Pseudonyms and ID-based signature, hash message authentication code	Supports batch authentication of requests; Tamper-proof devices (TPDs) are needed to store pseudonyms; TPDs could be susceptible to side-channel attacks.
Lin and Li [63]	Privacy, security, and authentication	Cooperative message authentication; uses large number of pseudonyms for ensuring anonymity	Due to the large number of pseudonyms issued to vehicles, CRLs could grow.
Lai et al. [57]	Reliable cooperative downloading	PKI; incentive scheme based on reputation	Can resist different types of attacks including DoS attacks; can be difficult to calculate and store reputation information correctly.
Wang and Liu [108]	Secure cooperative communication in heterogeneous vehicular networks	PKI, stochastic geometry theory and optimization	Flexible; allows to switch between DSRC, D2D-V, and cellular networks modes; Requires OBUs with high computation power.

Wang and Liu [108] proposed a scheme that satisfies the security requirements in

Table 1.8: Brief summary of protocols that support message aggregation, cooperative message authentication, and/or batch verification continued ...

Dietzel et al. [32]	Secure data aggregation	Generic aggregation model and Fuzzy logic methodology	Extensible and alleviates the need of a Global Unique Identifier system (GUID); bandwidth overhead could decrease dissemination speed.
Hao et al. [44]	Authentication, security, and privacy	Bilinear pairing; short group signatures [16]	Cooperative message authentication to speed up authentication; does not ensure location privacy of vehicles; vehicles in different regions cannot securely exchange messages; group-signatures generally have high signature verification and revocation costs; RSUs are assumed to be trustworthy; cooperative authentication would work only if the density of vehicles is high; susceptible to location modification because messages are selected for verification based on location information.

Vehicular Heterogeneous Networks (VHNs) wherein support for cooperative communication among various types of networks such as networks based on DSRC-based on IEEE 802.11p, Device to Device (D2D) communication and cellular communication needs to be provided. A mode selection algorithm that allows the OBUs to check the remaining lifetime of a packet and switch between three different modes (DSRC, D2D-V and cellular networks) is also presented. They found that sufficient power and vehicle density are the main factors for the successful transmission of messages securely in such networks.

Lin and Li [63] presented a cooperative message authentication scheme to reduce not only the overhead involved in message authentication but also the authentication delay. This scheme tries to minimize the authentication overhead on the same message by different vehicles when vehicles are allowed to cooperatively authenticate messages. To encourage vehicles to cooperate in message authentication, vehicles are issued evidence tokens. An evidence token issued to a vehicle reflects its contribution to authentication in the past; this encourages vehicles to participate in the message authentication process, which helps in distributing the authentication load among many vehicles. Evidence tokens are obtained from the TA via the RSU in its current region. It also uses a large number of pseudonyms, which could result in long CRLs. Jiang et al. [51] also propose an authentication scheme under which requests from

multiple vehicles can be authenticated in batches rather than one by one. Cheon and Yi [26] proposed a method for batch verification of multiple signatures generated by different signers as well as a single signer. They showed how this technique can be applied to the modified DSA and ECDSA based signatures. They also show that their batch verification approach is seven times faster than individual verification. Wasef et al. [115, 114] proposed a flexible certificate distribution scheme and an efficient way for vehicles to update their certificates. To decrease the message authentication overhead, they also proposed a method for verifying certificate-based signatures of messages in batches. Zhang and Zhang [137] developed a method for aggregating signatures in a certificate-less public key setting.

Table 1.7 and Table 1.8 summarize the strengths and weaknesses of the protocols discussed in this section.

1.2 Motivation for the Dissertation

Although the potential benefit of VANETs is increasing, widespread deployment of VANETs face some serious challenges. Due to the wireless nature of VANET communication, it is vulnerable to a large number of attack vectors. Malicious attackers can intercept, modify, and replay transmitted messages due to the wireless nature of V2V and V2I communication. Therefore, a vehicle user needs to check the authenticity and integrity of a received message as well as the legitimacy of the sender vehicle to accept the message. Authentication plays an important role in secure message dissemination. Without an effective authentication framework, attackers could compromise other drivers on VANETs easily. For example, malicious vehicles spreading false information about an accident might block the road, leading to a traffic jam. It may also spoof an RSU or electronic toll booth to steal other drivers' sensitive data. Moreover, vehicle users may refuse to participate in VANET due to a lack of privacy and security. We address this problem and present a distributed, scalable, low overhead, and privacy-preserving authentication scheme in this dissertation. Our scheme exploits Merkle Hash Tree and Modified Merkle Patricia Trie (MMPT) to overcome the performance limitations of the conventional approach of authentication such as ECDSA algorithm for efficient authentication of RSUs [74] and Vehicles [75] in VANETs.

Most of the solutions for ensuring security and privacy in VANETs use a pseudonym-based approach to protect vehicles' privacy. In a pseudonym-based approach, legitimate vehicles get a set of pseudonyms from a Certificate Authority (CA) or Trusted Authority (TA) before participating in VANETs. Vehicles use pseudonyms instead of real identities in V2V and V2I communication. In order to avoid traceability, vehicles change their pseudonyms frequently. When vehicles are about to run out of pseudonyms, they request a new set of pseudonyms. Pseudonyms ensure conditional privacy in the sense that TA can still retrieve the real ID of vehicles. So, the TA can revoke the credentials and pseudonyms of malicious vehicles. In this approach, Certificate Revocation List (CRL) is used to store all the unexpired pseudonyms of revoked or malicious vehicles. The CA or TA also stores the credentials of revoked or compromised RSUs in a CRL. TA distributes the CRLs periodically. Both vehi-

cles and RSUs store and check the CRL to authenticate each other. However, the size of CRL increases with the number of revoked vehicles and RSUs which leads to significant storage and computation overhead. It also incurs a higher delay to update and broadcast the CRL periodically. We present a low overhead and efficient privacy-preserving pseudonym-based authentication scheme [77] to address the above challenges. Our scheme leverages cuckoo filters to reduce the storage, computation, and communication overhead associated with the CRL. Vehicles and RSUs only need to store and check cuckoo filters for mutual authentication.

In a pseudonym based approach, vehicles are required to change their pseudonym frequently to avoid traceability. However, periodically changing the pseudonym of a vehicle is not effective to prevent pseudonym linking attacks. For example, suppose out of 100 vehicles, only one vehicle changes pseudonym. In that case, an intruder can easily link the old and the new pseudonyms used by the vehicle by linking two messages to the same vehicle and track the path traversed by the vehicle. In addition to that, more research needs to be done in devising an efficient method for managing pseudonyms of vehicles.

Sometimes a single malicious message is not powerful enough to affect the security of the VANETs system. However, a serious threat arises when a malicious vehicle uses multiple pseudonyms in parallel to impersonate a number of vehicles and generate multiple false messages. Since vehicles use the information in received messages to make decisions or take action specially in a critical situation, such as a road accident or traffic jam etc., vehicles unable to detect this type of attack become vulnerable. Therefore, the number of pseudonyms and their validity period that a vehicle can use should be limited.

We address the above issues and propose a distributed and decentralized certificateless authentication scheme [76] to efficiently store vehicles' pseudonyms and their corresponding 'current status' values. Only one pseudonym of a vehicle is valid in our scheme due to the concatenation of pseudonym expiration time to resist Sybil attacks. In our scheme, RSUs assist vehicles within their transmission range to change their pseudonym by associating an expiration time with the registered pseudonym. Once this expiration time elapses, the vehicle will again communicate with an RSU to activate a new pseudonym from a pool of pseudonyms received from its home regional TA during initial registration. We assume that a vehicle will always have a sufficient number of pseudonyms, so that it will not need to reuse a pseudonym within a year. Expiration time associated with a pseudonym helps vehicles within the same RSU's region to change their pseudonym simultaneously and frequently to reduce the chance of linkability between the same vehicle's two pseudonyms.

1.3 Organization of the Dissertation

The remainder of this dissertation is organized as follows.

- In Chapter 2, we present a distributed, scalable, low-overhead and privacy-preserving authentication scheme for VANETs. Our scheme utilizes a Merkle Hash Tree (MHT) to authenticate Road Side Units (RSUs) and Modified Merkle

Patricia Trie (MMPT) to authenticate vehicles. Upon successful mutual authentication of RSUs and vehicles, messages are encrypted and sent to RSUs for further dissemination. Each RSU creates a group key for the legitimate vehicles within its region to disseminate messages securely.

- We present a lightweight privacy-preserving V2I authentication scheme based on cuckoo filters in Chapter 3. In comparison with Certificate Revocation Lists (CRLs), the use of cuckoo filters significantly reduces the storage, communication, and computation overheads for V2I authentication.
- In Chapter 4, we present a Certificateless and REused-pseudonym based Authentication Scheme for Enabling security and privacy (CREASE) in VANETs. Our scheme uses Merkle Hash Tree combined with Modified Merkle Patricia Trie to efficiently store and manage the set of pseudonyms assigned to a vehicle. A vehicle in our scheme picks and uses a random pseudonym from a given set of pseudonyms assigned to it as well as changes its pseudonym frequently and securely to ensure privacy. In our scheme, each RSU assists vehicles in its region to change their pseudonym simultaneously to prevent vehicles' routes being tracked; this is accomplished by assigning the same expiration time for all the pseudonyms of all vehicles in its region.
- Finally, Chapter 5 summarizes the dissertation.

Chapter 2 Secure and Privacy-preserving Authentication and Message Dissemination in VANETs

In this chapter, we propose a distributed, scalable, low-overhead, and privacy-preserving authentication scheme for VANETs. The proposed scheme uses a Merkle Hash Tree (MHT) for authenticating Road Side Units (RSUs) and Modified Merkle Patricia Trie (MMPT) for authenticating vehicles. We also present an informal analysis as well as formal correctness proof of the proposed scheme.

2.1 Related works

In this section, we discuss some of the related works and their strengths and weaknesses. Many of the existing schemes presented in the literature [71, 118, 51, 142] use CRL for authentication. However, the size of CRL grows with the increased number of revoked entities (Vehicles and RSUs), which can cause significant computation and storage overhead in large scale VANETs.

To address the computational and storage overhead caused by CRLs, Zhu *et al.* [142] proposed a privacy-preserving authentication scheme based on group signature [25]. They divided the region into domains and an RSU authenticates a vehicle whenever it receives a join request in its domain. After authentication, the vehicle gets the group key for hash message authentication code (HMAC) computation to facilitate message authentication without checking the CRL. HMAC is used to ensure message integrity and reduce the number of invalid messages. RSUs are assumed to be trusted in this scheme. However, any malicious entity can pretend to be an RSU and steal the sensitive information of vehicle users.

Liu *et al.* [67] proposed a lightweight V2I authentication protocol that employs group communication to reduce the computational overhead involved in checking the CRL. In this scheme, first the TA predicts the RSUs that the OBU will pass by. After that TA initializes a communication group containing the OBU and the RSUs and distributes the group key to them. TA deletes the information about a misbehaving vehicle and forwards this update to all RSUs. This approach is not scalable.

Wasef *et al.* [118] proposed an Expedite Message Authentication Protocol (EMAP) based on the keyed-Hash Message Authentication Code (HMAC) to reduce the computation time for authentication and revocation process. In this scheme, only legitimate vehicles get a shared secret key from the TA to calculate HMAC. Later on, vehicles broadcast messages along with the calculated HMAC to facilitate the revocation status checking process. This scheme reduces the revocation checking time caused by conventional authentication schemes that adopt CRL. TA updates the shared secret key whenever it revokes the certificate of misbehaving vehicles. However, it is difficult to update the shared secret key globally. Besides, TA broadcasts a revocation message containing certificates of revoked vehicles. Thus, this scheme still requires the distribution of CRL, which incurs communication overhead in the system.

Jiang *et al.* [52] proposed an anonymous batch authentication scheme (ABAH) based on hash message authentication code (HMAC) to replace the CRL checking process. In their scheme, a large area is divided into several domains, and RSUs are responsible for managing vehicles in a localized manner. The RSU authenticates vehicles by checking their revocation status in CRL and sends the group key to valid vehicles. If the authentication fails, the RSU does not send the group key to the revoked vehicle. After successful authentication, vehicles broadcast safety-related messages containing HMAC, which is derived from the group key. However, the revocation status checking process using CRL of vehicles still exists in their scheme.

Wang and Yuo [111] proposed a local identity based anonymous message authentication protocol (LIAP) in which vehicles and RSUs obtain long term certificate from CA (Certificate Authority) for mutual authentication. In this scheme, the certificate revocation list of RSUs (RCRL) and vehicles (VCRL) are distributed separately by the CA. Each RSU uses a linear search algorithm to check VCRL for authenticating vehicles. On the other hand, vehicles only store RCRL to check the validity of RSU. After mutual authentication, vehicles get keys from RSUs to generate pseudonym for V2V communication.

Paruchuri and Duresi [83] proposed a certificate-based scheme that provides anonymous authentication using a smart card. The smart card stores vehicle's information and required cryptographic keys. Only OBUs that have been authenticated get session keys from RSU for secure and authenticated communication with other vehicles as well as RSUs. However, each OBU stores many session keys from different RSUs in this scheme. In addition to that, if the smart card is stolen or lost, then secret information can be leaked.

Tangade *et al.* [104] proposed a decentralized and scalable privacy-preserving authentication scheme based on identity-based cryptography and symmetric hash message authentication code (HMAC). This scheme uses ID-based cryptography for authentication of vehicles by RSUs and HMAC for the vehicle to vehicle authentication. However, they did not consider the authentication of RSU by vehicles during V2I communication.

Motivated by the existing authentication schemes' above drawbacks, we propose a distributed and scalable conditional privacy-preserving authentication scheme for VANETs. Our scheme uses a distributed architecture. Moreover, in our scheme, we combine Merkle Hash Tree (MHT) root value with the latest MHT root-generation timestamp to replace CRL of RSUs for efficient RSUs authentication. We use Modified Merkle Patricia Trie (MMPT) to efficiently manage revoked vehicles to overcome the drawbacks of using CRL for authenticating vehicles. After successful mutual authentication of RSUs and vehicles, messages are encrypted and sent to RSUs for further dissemination. RSUs establish a group key shared only between legitimate vehicles within its region to disseminate messages securely.

2.2 Background, System Model, and Design Goals

In this section, we first describe Merkle Hash Tree (MHT) [72] and Modified Merkle Patricia Trie (MMPT), which are used in our scheme. After that, we briefly intro-

duce the system model. Then, we present our design goals and assumptions. We do not provide a detailed description of RSA encryption, RSA decryption, RSA signature, RSA signature verification, secure hashing algorithm (SHA), and public key certificate. We assume that the reader is familiar with these terminologies/concepts.

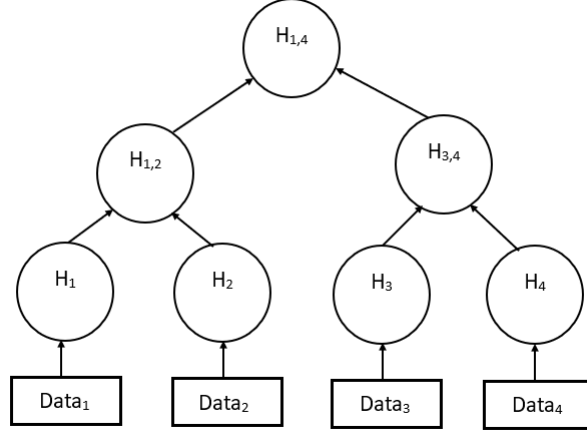


Figure 2.1: Merkle Hash Tree

2.2.1 Merkle Hash Tree

A Merkle Hash Tree (MHT) [72] is a hash-based [89] tree structure in which each leaf node stores the data and each non-leaf node contains the hash of its children. Fig. 2.1 shows an MHT with four leaf nodes where data are stored and values stored at non-leaf nodes are derived from hash of its children. MHT helps in verifying the integrity of data stored at leaf nodes efficiently. For instance, we only need corresponding Missing Hash Values (MHVs) of MHT (H_2 , $H_{3,4}$) and root value $H_{1,4}$ instead of entire tree structure to verify the integrity of $Data_1$. We can recompute the root hash value from MHVs by first computing $H_{1,2} = H(H(Data_1), H_2)$ and then $H'_{1,4} = H(H_{1,2}, H_{3,4})$. If the computed $H'_{1,4}$ is same as the original root value $H_{1,4}$, integrity of $Data_1$ is ensured.

2.2.2 Modified Merkle Patricia Trie

Modified Merkle Patricia Trie (MMPT) is based on Patricia Trie and Merkle Tree with additional optimizations to meet the requirements of Ethereum [121]. Patricia Trie (also known as prefix tree, radix tree or trie) finds common prefix data more efficiently compared to an hash table. It stores single character of the key or string at each level. However, it has a time complexity of $O(n)$ (where n is the length of the key or string) for searching and inserting.

For example, there is a long string of nodes on the left side of Fig. 2.2 wherein most of the nodes are non-leaf nodes and used to build a path to store the word "TOAST".

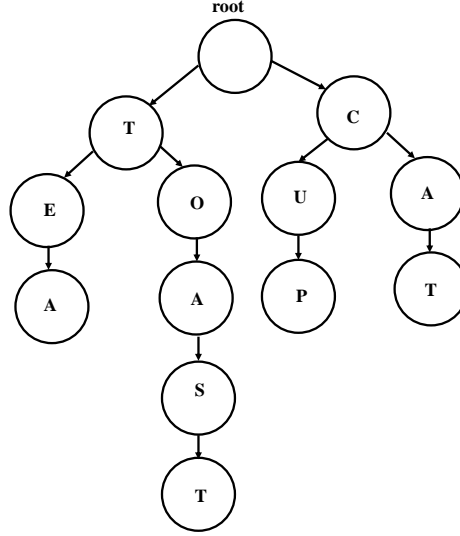


Figure 2.2: Prefix Tree

It is necessary to read each letter in the word and travel down the corresponding path to find the word. On the other hand, if there is currently no path with the same prefix as the node key during the insertion of a node, the MMPT stores the remaining keys in the key field of the leaf or extension node.

MMPT is a new data structure that binds a key with a value and can be authenticated cryptographically [69]. Every node in MMPT is expressed as a key-value pair. This key-value pair is used as paths in MMPT, and nibble is the unit that composes the path. Complexity involved for insert, lookup, and deletion of a node in MMPT is $O(\log n)$ (where n is the total number of leaf nodes in MMPT). MMPT has three types of nodes, i.e., leaf node, branch node, and extension node. Next, we describe these nodes and also explain how we use MMPT to store the public keys of revoked vehicles.

- **Leaf Node:** The prefix field in a leaf node determines the type of node, and the prefix value of the leaf node in our model is 2. In our scheme, the leaf node contains the public key PU_V of a revoked vehicle V as key while value contains the ID ID_{RSU} of RSU that has reported about the vehicle.
- **Branch Node:** Branch nodes are non-leaf nodes and have more than one child node. There are up to 16 branches, from 0 to f in a branch node where each branch contains one hexadecimal digit. A branch node has a prefix value of 1 in our scheme.
- **Extension Node:** An extension node is an optimized node of a branch node. The prefix value of an extension node in our scheme is 0. An extension node's key consists of a partial path (shared nibble) that allows us to skip ahead and a pointer to the next node.

Table 2.1: Information of Malicious Vehicles stored in MMPT

PU_V	RSU ID
$a711355$	43245699
$a77d337$	43245676
$a7f9365$	43245625
$a77d397$	43247681

Operations on MMPT

Next we explain how we use MMPT to store the pair (PU_V, ID_{RSU}) , where PU_V is the public key (in hexadecimal representation) of vehicle V which has been reported to be malicious and ID_{RSU} is the identity of the RSU which reported V to be malicious. MMPT is used by an RSU for authenticating a vehicle V. Table 2.1 contains a sample list of $(PU_V, RSU ID)$ pairs of vehicles reported to be malicious. In this table, public keys of vehicles are given in hexadecimal form. Fig. 2.3 shows the MMPT storing the four key-value pairs in Table 2.1, where all keys share the same nibble of $a7$. In the MMPT, the root node is an extension node that contains the shared nibble ($a7$). The "next node" field in the root node points to the next node, which in our case is a branch node (Branch Node 0). If we look at the first key ($a711355$), we can find 1 after $a7$. This 1 allows us to skip ahead and leads us to the leaf node (Leaf Node

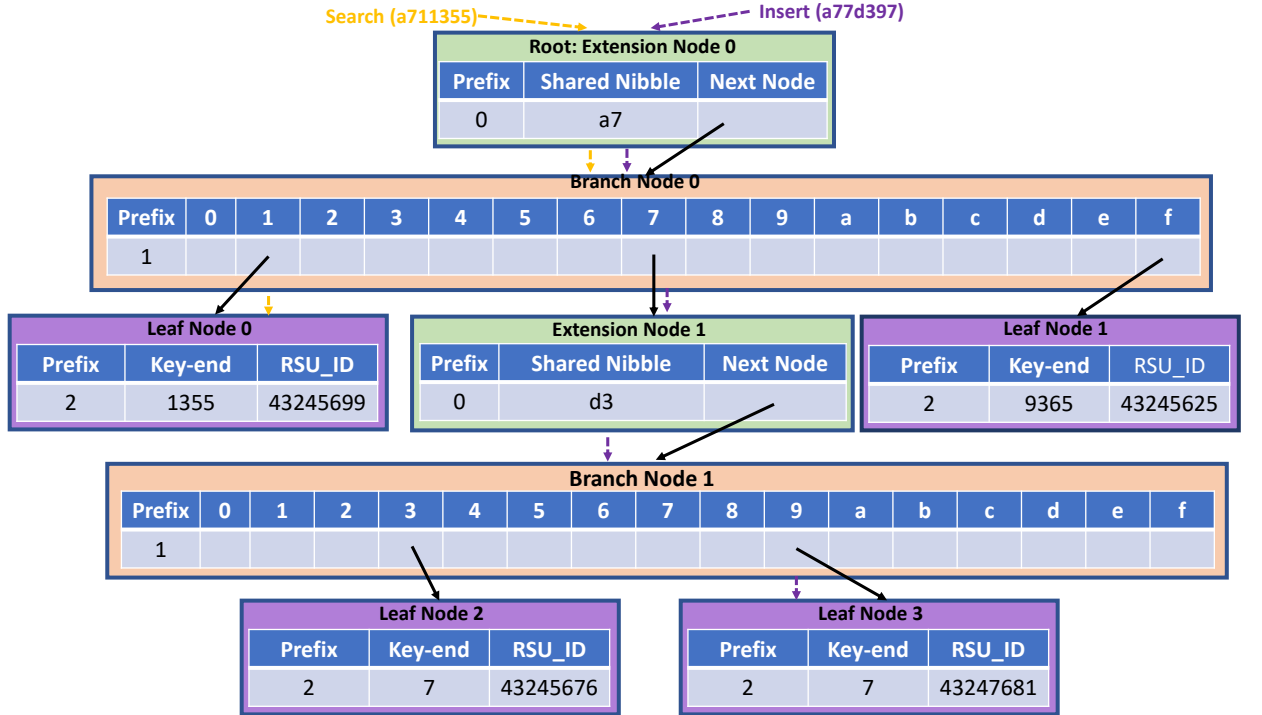


Figure 2.3: Modified Merkle Patricia Trie storing the contents of Table 2.1

0) in Fig. 2.3, where the remaining value of the key along with the RSU ID is stored. Therefore, We need to start traversing from the root node to lookup a key in MMPT.

The insertion operation inserts the public key PU_V of a malicious vehicle along with the ID ID_{RSU} of the RSU which reported the vehicle to be malicious into the MMPT. To insert a entry, we need to traverse from the root node. As we traverse, we need get the prefix of the current node and its nibbles. If the current node's prefix is 1, we check whether the slot corresponding to the next nibble of the branch node points to NULL. If so, a new leaf node or a new extension node is generated based on the remaining nibbles left in the key to be inserted. If not, then traverse down to the next node. If the prefix of the current node is 0, then find the shared nibbles and remaining nibbles left in the key. Next, generate a new leaf node, or a new branch node, or a new extension node according to the remaining nibbles in the key after sharing. For example, in Fig. 2.3, when we insert the key "a77d397", we first start from the root node and look at the prefix of the current node. In our case, the root node is Extension Node 0, and the pointer to the next node field points to the Branch Node 0. The slot corresponding to the next nibble in the Branch Node 0 is not NULL, and the remaining nibbles in the key are greater than 1. Next, we travel down to Extension Node 1, where the partial path has diverged at Branch Node 1. Hence the slot corresponding to the Branch Node 1 is NULL, and the left nibble in the key is 1. Therefore, a new leaf node (Leaf Node 3) is generated into this branch.

The branch nodes, extension nodes, and leaf nodes used in MMPT help in shortening the length of the unique path to the leaf nodes and make all the three basic operations (lookup, insert, and delete) efficient. So, to insert/delete/search the public key of a revoked vehicle that is 2028 bits long, we need to traverse at most 512 nodes in the MMPT, if we use hexadecimal representation of the key; if we use Base64 representation, we traverse at most 338 nodes in the MMPT.

2.2.3 System Model

The proposed VANET system architecture, shown in Fig. 2.4, has two layers. The upper layer consists of the Trusted Authority (TA) and Regional Trusted Authorities (RTAs), and the lower layer consists of RSUs and OBUs. TA is the root of the whole system, while each RTA acts as a lower-level local TA. In this scheme, TA may represent a whole country, or a state and RTA may represent a state or a county/district. When a vehicle moves from the region covered by one RTA to a region covered by another RTA, the RTA under which that vehicle has been registered (called home RTA) will be contacted to get the credential of the vehicle for authentication. TA and RTAs communicate with each other through a secured channel. On the other hand, RSUs and vehicles communicate with each other based on the Dedicated Short Range Communications (DSRC) protocol standard [49]. We also assume communication between the RSUs as well as between RSUs and RTAs are secure. Functions of TA, RTAs, RSUs, and vehicles are described next:

- **TA:** The TA generates its public and private keys (PU_{TA} , PR_{TA}) and distributes PU_{TA} to all RTAs, RSUs, and vehicles securely. The TA also generates

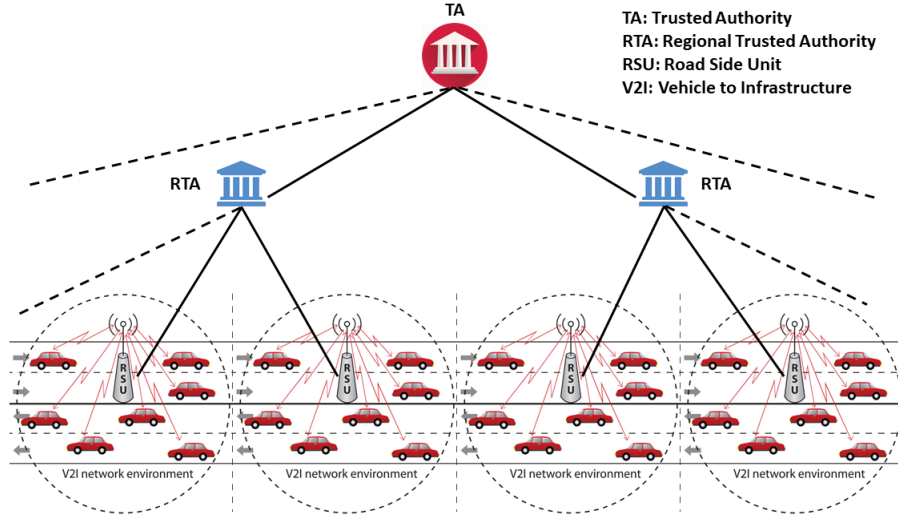


Figure 2.4: Proposed VANET architecture for efficient authentication of RSUs and vehicles

public, private key pair (PU_{RTA}, PR_{RTA}) for each RTA and distributes them to the respective RTAs securely.

- **RTA:** Each RTA is responsible for generating and distributing public and private keys for all vehicles as well as RSUs registered under it. Each RTA also generates the pseudo-ID of vehicles and maintains an MHT that is created based on the public key of all RSUs registered under it. RTA also forwards the credentials of RSUs and vehicles registered under it to the TA. RTA also distributes PU_{TA} to all RSUs in its region and all vehicles registered with it.
- **RSU:** RSUs are placed along the roadside and get their public and private key pair (PU_{RSU}, PR_{RSU}) from their local RTA, called **home RTA**.
- **Vehicle (OBU):** For a vehicle to participate in VANET, it must be registered with its RTA, called **home RTA**. Each vehicle V is equipped with an On Board Unit (OBU) for communication with other vehicles and nearby RSUs. The OBU also stores the vehicle's public and private key pair (PU_V, PR_V) , pseudo-ID (PID_V) , certificate $(Cert_V)$, signed by the RTA, and MHT root generation timestamp $(T_{mhtRoot})$ generated and signed by the RTA as well as PU_{TA} . $Cert_V$ contains PU_V , PID_V , and certificate expiration time T_{exp} . $Cert_V$ and MMPT are used by RSUs to authenticate the vehicle V .

2.2.4 Design Goals

Following are the design goals behind our protocol:

- **Resilience to message tampering:** We aim to provide an efficient scheme for mutual authentication of vehicles and RSUs, to ensure that message comes from a legitimate source without its payload being manipulated.
- **Privacy preservation:** Privacy is a major concern in VANETs. So, the true identity of a vehicle should never be revealed in communication, even if all the RSUs are compromised. In our proposed scheme, only the RTA under which the vehicle is registered and the TA know the real identity of vehicles and can reveal the real identity of a vehicle to authorities in case the vehicle is determined to be malicious.
- **Non-repudiation:** Non-repudiation refers to the ability to ensure that a party sending a message cannot deny sending of the message. To ensure non-repudiation, we use the signature algorithms to sign messages during the communication process.
- **Authentication of RSUs and Vehicles:** Our scheme is designed to reduce the overhead related to using CRL for authenticating entities (Vehicles and RSUs) in VANETs. In our scheme, we use MHT to authenticate RSUs and MMPT to authenticate vehicles. We do not use CRLs for authentication.

2.2.5 Assumptions

We assume the following:

- The TA has more computing, communication, and storage capacity than RTAs; RTAs have more computing, communication, and storage capacity than RSUs; RSUs have more computing, communication, and storage capacity than OBU.
- Each RTA constructs an MHT of the public keys of RSUs registered with that RTA.
- The TA, RTAs, and RSUs use MMPT to store the public keys of revoked vehicles.
- RSUs are responsible for receiving messages from vehicles in its region and authenticating messages; if necessary, each RSU is also responsible for disseminating the messages to all vehicles in its region as well as its home RTA for further dissemination.

2.3 Proposed Scheme

In this section, we describe our privacy-preserving authentication and secure message dissemination scheme.

Table 2.2: Description of Notations used in this Chapter 2

Notation	Description
TA	Trusted Authority
RTA	Regional TA
RSU	Road Side Unit
OBU	On Board Unit
DSRC	Dedicated Short Range Communications
CRL	Certificate Revocation List
MHT	Merkle Hash Tree
MMPT	Modified Merkle Patricia Trie
MHV	Missing Hash Values
E	RSA 2048 bit Encryption Algorithm
PU_{TA}, PR_{TA}	Public and Private Keys of the TA
PU_{RTA}, PR_{RTA}	Public and Private Keys of RTA
PU_{RSU}, PR_{RSU}	Public and Private Keys of RSU
PU_V, PR_V	Public and Private Keys of Vehicle V
ID_{RSU}	ID of RSU
PID_V	Pseudo-ID of Vehicle V
RID_V	Real ID of Vehicle V
T_{exp}	Certificate expiration Time
S_k	Symmetric key between RSU and vehicle
$Cert_V$	Certificate of vehicle V
$signbyTA$	Signature of TA
$signbyRTA$	Signature of RTA
H	SHA-256 hash function
T_s	Message generation timestamp
$T_{mhtRoot}$	MHT root generation timestamp

2.3.1 System Initialization

When a vehicle V registers with its home RTA, the RTA encrypts a block containing the pseudo-ID PID_V , public key of the vehicle PU_V , certificate expiration time T_{exp} , etc. with its private key PR_{RTA} to generate a certificate $Cert_V$ for V, where $Cert_V = E((PID_V, PU_V, T_{exp}, \dots), PR_{RTA})$. A vehicle sends a request for new certificate to the RTA whenever the current certificate is about to expire.

Each vehicle gets its PID_V , PU_V , PR_V , $Cert_V$, PU_{RTA} , and PU_{TA} from its home RTA securely. A vehicle uses its PID_V in all communications to preserve its privacy. To protect privacy, many methods have been proposed in the literature [120, 85, 39, 98, 20, 113, 112, 127, 18] for changing pseudonym. Vehicles can use any of the existing methods to change their pseudonym. The real identity of a vehicle V can only be determined by its home RTA and the TA. The abbreviations used in this chapter are listed in Table 2.2.

2.3.2 Construction of MMPT for Storing Public Keys of Revoked Vehicles

The TA, RTAs, and RSUs use an MMPT to store public keys of revoked vehicles and the ID of the RSU that has reported the vehicle to be malicious. Each leaf node in the MMPT contains PU_V of a revoked vehicle as key, while value contains the ID of the RSU (ID_{RSU}) that has reported the vehicle as malicious. An RSU sends a misbehaving report to the RTA whenever a malicious vehicle is found in its region. *Details regarding how a vehicle is determined to be malicious are not addressed in this research. Many researchers have addressed the malicious node detection problem in Mobile Ad hoc NETWORKs (MANETs) and VANETs. RSUs can use any of those mechanisms to detect malicious vehicles.* After authenticating sender RSU and checking the integrity of the received message, RTA revokes the certificate of the malicious vehicle and inserts the PU_V of the revoked vehicle along with ID_{RSU} in MMPT and also forwards this update to the TA. The RTA broadcasts this update to all RSUs in its region as well. After receiving the update message, the TA and RSUs add an entry of PU_V of the revoked vehicle along with ID_{RSU} to their respective MMPTs. MMPT helps RSUs in determining quickly whether a vehicle's public key has been revoked, unlike CRL based revocation schemes.

2.3.3 Construction of MHT for Storing Public Keys of Registered RSUs

Each RTA constructs an MHT of the public keys of all RSUs registered under it. A sample MHT of sixty four registered RSUs under an RTA is shown in Fig. 2.5, where each leaf node stores the public key PU_{RSU} of one RSU registered under it. Initially, an RTA sends the following information to each of the RSUs in its region:

- its own public key signed by the TA ($PU_{RTA\text{signbyTA}}$).
- root value of the MHT signed by the RTA ($root_{\text{signbyRTA}}$).
- corresponding MHVs (described below) that fall along the authentication path of that RSU.

$RTA \rightarrow RSU : (E(PU_{RTA\text{signbyTA}}, root_{\text{signbyRTA}}, MHVs), PU_{RSU})$, where $root_{\text{signbyRTA}} = (\text{root of MHT} \parallel \text{RSA signature of RTA} \parallel T_{mhtRoot})$

MHVs corresponding to different RSUs in Fig. 2.5 are shown in Table 2.3. In the proposed scheme, an RTA reconstructs MHT by discarding previous root of MHT and $T_{mhtRoot}$ whenever a compromised RSU is found, and forwards the updated MHVs and MHT root by signing it along with updated $T_{mhtRoot}$ to all the RSUs in its region. Every RTA sends the MHT root generation timestamp ($T_{mhtRoot}$) to the TA whenever it constructs or reconstructs the MHT. The TA broadcasts latest value of $T_{mhtRoot}$ in the system to all RTAs, the RTAs in turn broadcast this value to all RSUs in its region and the RSUs broadcast to vehicles within their transmission range.

Table 2.3: Authentication Information

RSU_i	MHV_s
RSU_1	$H_2, H_{3,4}, H_{5,8}, H_{9,16}, H_{17,32}, H_{33,64}$
RSU_2	$H_1, H_{3,4}, H_{5,8}, H_{9,16}, H_{17,32}, H_{33,64}$
RSU_3	$H_4, H_{1,2}, H_{5,8}, H_{9,16}, H_{17,32}, H_{33,64}$
.....
RSU_{33}	$H_{34}, H_{35,36}, H_{37,40}, H_{41,48}, H_{49,64}, H_{1,32}$
.....
RSU_{64}	$H_{63}, H_{61,62}, H_{57,60}, H_{49,56}, H_{33,48}, H_{1,32}$

MHV_s corresponding to the public keys of various RSUs in Fig. 2.5 under an RTA are shown in Table 2.3.

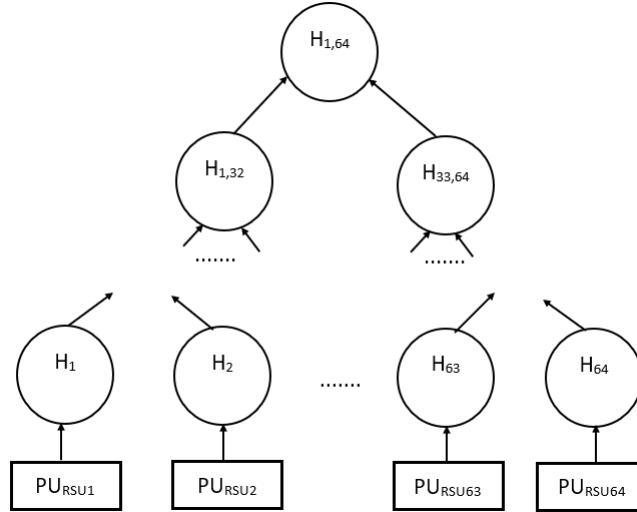


Figure 2.5: Merkle Hash Tree of public keys of RSUs

2.3.4 When a Newly Registered Vehicle V Enters the Region of an RSU under its Home RTA

Each RSU periodically broadcasts beacon messages containing its ID (ID_{RSU}) and public key (PU_{RSU}). When a vehicle V enters an area covered by an RSU, it retrieves PU_{RSU} from the beacon message and sends an authentication request message to verify the authenticity of PU_{RSU} . Firstly, a newly registered vehicle V sends an authentication request message to the RSU containing its PID_V , PU_V , $Cert_V$ and message generation timestamp (T_s). After receiving the message, the RSU checks the freshness of the message using T_s . Then the RSU retrieves the public key of the vehicle from $Cert_V$ and checks using MMPT if the vehicle's public key PU_V has been revoked. RSU will drop the received message if any of the above verifications fails.

After verification, RSU computes the hash value of PU_V and a symmetric key S_k to be used for secure communication between the vehicle and the RSU. When a vehicle V moves from one RSU to another RSU both of which are under V's home RTA, it presents this hash value to the new RSU for authentication. RSU also generates a group key G_k for sending messages to all vehicles within its region and updates the group key periodically. RSU sends a response message to V containing $PU_{RTA \text{ signbyTA}}$, $root_{\text{signbyRTA}}$, G_k , S_k , $Hash_{\text{signbyRSU}}$, and MHVs corresponding to the public key of the RSU along with the T_s , encrypted using the public key PU_V of V as follows:

$$RSU \rightarrow V: (E(PU_{RTA \text{ signbyTA}}, root_{\text{signbyRTA}}, MHVs, S_k, Hash_{\text{signbyRSU}}, G_k, T_s), PU_V), ID_{RSU}, PID_V), \text{ where } Hash_{\text{signbyRSU}} = E(H(PU_V, S_k), PR_{RSU})$$

Upon receiving the above reply from the RSU, the vehicle uses received T_s to verify freshness of the reply message. After that, it verifies the signature of the TA and RTA. Then, V compares $T_{mhtRoot}$ in reply message with the stored $T_{mhtRoot}$. If the received $T_{mhtRoot}$ is greater than or equal to the stored $T_{mhtRoot}$, then the vehicle recalculates the root of MHT using the received MHVs and hash value of the public key PU_{RSU} of the sender RSU received in the beacon message. Fig. 2.6 illustrates the authentication process between a newly registered vehicle V and an RSU when V enters the region of an RSU under its home RTA for the first time. Fig. 2.7 illustrates the mutual authentication process between an RSU and a vehicle V when V moves from the region covered by one RSU to the region covered by another RSU both of which are under its home RTA.

For example, to authenticate the public key PU_{RSU_2} of RSU_2 retrieved from the beacon message of RSU_2 depicted in Fig. 2.5, hash values are computed in the following manner using the MHVs ($H_1, H_{3,4}, H_{5,8}, H_{9,16}, H_{17,32}, H_{33,64}$) received in the response message from RSU_2 .

$$\begin{aligned} H_{1,2} &= H(H(PU_{RSU_2}), H_1) \\ H_{1,4} &= H(H_{1,2}, H_{3,4}) \\ H_{1,8} &= H(H_{1,4}, H_{5,8}) \\ H_{1,16} &= H(H_{1,8}, H_{9,16}) \\ H_{1,32} &= H(H_{1,16}, H_{17,32}) \\ H'_{1,64} &= H(H_{1,32}, H_{33,64}) \end{aligned}$$

Next, the computed root hash value $H'_{1,64}$ is compared with the root hash value $H_{1,64}$ (signed by the RTA) of the MHT, received in the response message from RSU_2 . If $H'_{1,64} = H_{1,64}$, RSU_2 is authenticated (i.e., the public key of RSU_2 retrieved from its beacon message is verified to be authentic); otherwise, authentication of RSU_2 fails. Then V stores updated $T_{mhtRoot}$ after successful authentication of the RSU. Next, V can send/receive messages encrypted using S_k/G_k to/from RSU.

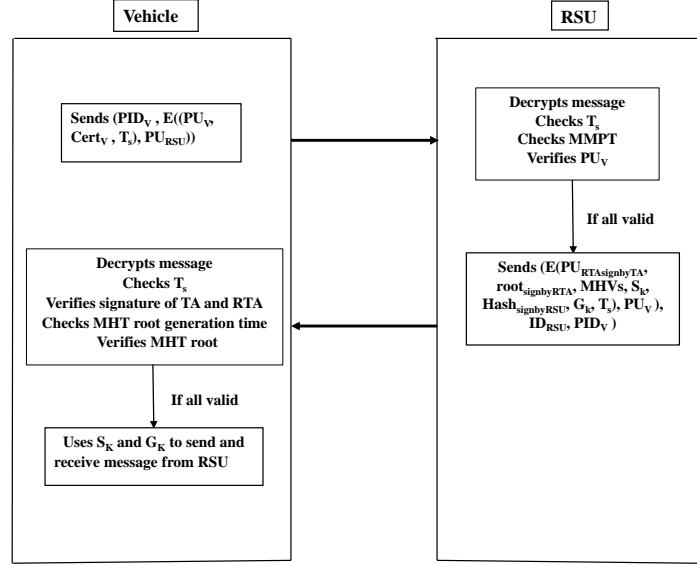


Figure 2.6: Mutual Authentication between an RSU and a newly registered vehicle V

2.3.5 When a Vehicle V Needs to Send a Message M to the Nearby RSU

When V wants to send a message M about an observed event to the nearby RSU, it signs and encrypts M and sends it to the RSU as follows:

$V \rightarrow RSU: (E((M, Sign_{msgM}, T_s), S_k), PID_V, ID_{RSU})$

where $Sign_{msgM} = E(H(M), PR_V)$, T_s is the timestamp, S_k is the symmetric key established between V and RSU and PID_V is the pseudo-ID of V.

RSU uses Signature $Sign_{msgM}$ for establishing non-repudiation whenever a malicious vehicle is found. If V does not find an RSU within its transmission range to send the message, it stores the message and carries it until it finds a nearby RSU. After finding a nearby RSU, V and RSU authenticate each other and exchange shared symmetric key and group key for communication. Then, V sends the message to the RSU by encrypting it using the newly established symmetric key S_k as described above. The above discussion is presented in detail in **Algorithm 1**.

2.3.6 When an RSU Receives a Message from a Vehicle V

When an RSU receives a message from a vehicle V about an observed event, it decrypts the message using the symmetric key S_k established before and checks the freshness of the message using T_s . It also determines the regions where the message needs to be disseminated based on nature of the message. The possible message dissemination scenario by RSU can be to-

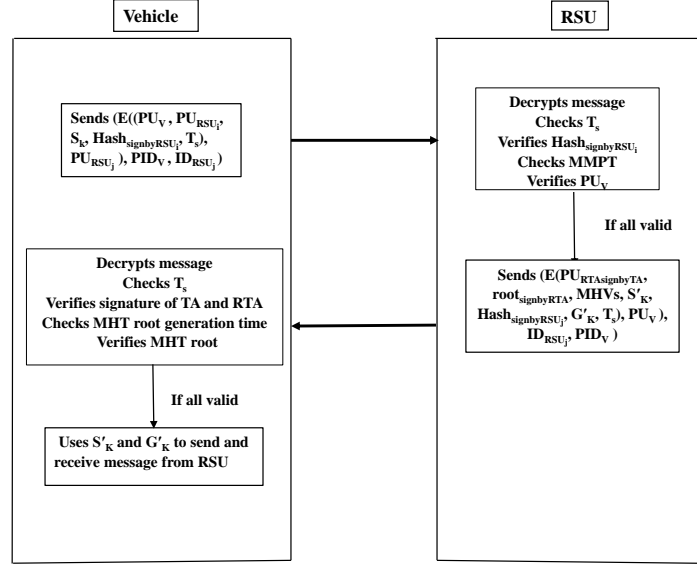


Figure 2.7: Mutual Authentication between an RSU and a vehicle when the vehicle moves from the region of RSU_i to the region of RSU_j where RSU_i and RSU_j are under the vehicle's home RTA.

Algorithm 1: When a vehicle V observes an event

- 1 **Case 1:**
 - 2 **if** V is within its transmission range of an authenticated RSU **then**
 - 3 V sends message M about the observed event by encrypting M using S_k as follows:
 - 4 Sends $(E((M, \text{Sign}_{msgM}, T_s), S_k), PID_V, ID_{RSU})$, where $\text{Sign}_{msgM} = E(H(M), PR_V)$;
 - 5 **end if**
 - 6 **Case 2:**
 - 7 **if** V does not find an RSU within its transmission range **then**
 - 8 It stores the message and carries it until it finds an RSU within its transmission range;
 - 9 After finding an RSU, V and RSU authenticate each other;
 - 10 Sends the message to the RSU as in **Case 1** ;
 - 11 **end if**
-

Case 1: all vehicles in its region,

Case 2: vehicles in other regions under its home RTA or other RTAs

Detailed description for message dissemination is given in **Algorithm 2**.

In case 1, RSU broadcasts the message to all vehicles within its transmission range

Algorithm 2: When an RSU receives a message M from a vehicle V

```
1 Decrypts it using symmetric key  $S_k$  and retrieves the message M;  
2 Checks the timestamp  $T_s$ ;  
3 if  $T_s$  is valid then  
4   Computes hash of the message M;  
5   Encrypts computed hash value with its private key  $PR_{RSU}$ ;  $Hash_{msgRSU}$ :  
    $E(H(M), PR_{RSU})$ ;  
6   Determines the regions where message requires dissemination;  
7   Case 1:  
8   if M needs to be disseminated to all vehicles in its region then  
9     Encrypts the message with group key  $G_k$  and broadcasts  $(E((M,$   
      $Hash_{msgRSU}, T_s), G_k), ID_{RSU})$ ;  
10  end if  
11  Case 2:  
12  if M needs to be disseminated to the vehicles in other regions covered by  
    its home RTA and/or other RTAs then  
13    Encrypts the message using the public key of its home RTA and sends  
    it to its home RTA as follows:  
14    Sends  $(E((M, Hash_{msgRSU}, T_s), PU_{RTA}), ID_{RSU})$ ;  
15  end if  
16 end if
```

by encrypting it with group key G_k as follows:

RSU \rightarrow Vehicles (within its transmission range): $(E((M, Hash_{msgRSU}, T_s), G_k), ID_{RSU})$

In case 2, the RSU sends the message to its home RTA for further dissemination as follows:

RSU \rightarrow RTA : $(E((M, Hash_{msgRSU}, T_s), PU_{RTA}), ID_{RSU})$

Encrypting large messages using public key cryptography is not efficient. In such cases, each RSU can establish a symmetric key between itself and its home RTA for encrypting large messages.

2.3.7 When a vehicle V Moves from one RTA's Region to Another RTA's Region

When a vehicle V moves from one RTA's region to another RTA's region, the RSU in that region communicates through its home RTA with the home RTA of V to verify the credentials of V. After verifying the credentials of V, RSU and V authenticate each other and V obtains symmetric key and group key from RSU after mutual authentication. Then V sends messages to the RSU using **Algorithm1**. When RSU receives the message, it first checks timestamp T_s to ensure the freshness of the message. After that, RSU disseminates the received message in a way similar to the one described in **Algorithm 2**. The above discussion is presented in detail in

Algorithm 3: When a vehicle V moves from one RTA's region to another RTA's region

```

1  RSU communicates with the home RTA of  $V$  through its own home RTA to
   get the credentials of  $V$ ;
2  if the credentials of  $V$  are valid then
3      Mutual authentication takes place between  $V$  and RSU;
4       $V$  sends message to the RSU using Algorithm 1;
5      RSU uses Algorithm 2 to disseminate the message received from  $V$ ;
6  end if

```

Algorithm3.

2.4 Analysis and Proof of Correctness of our Protocol

In this section, first we discuss how our protocol ensures security and supports scalability. Then, we present a formal correctness proof of our protocol using Burrows, Abadi, and Needham (BAN) [19] logic and the PKI-based extended BAN Logic [96]. Next, we also present the results of correctness verification using SPAN (Security Protocol ANimator) [50] and AVISPA (Automated Validation of Internet Security Protocols and Applications) [7] tools.

2.4.1 Security Analysis

Conditional Privacy Preservation

The real identity of a vehicle in VANETs should not be traceable by attackers to ensure privacy. However, a vehicle's real identity needs to be revealed when a vehicle is determined to be malicious. In our scheme, only RTA and the TA know the real ID of vehicles. A vehicle never uses its real ID to send messages. Each newly registered vehicle gets a certificate to facilitate the authentication process. The certificate contains only pseudo-ID PID_V , public key PU_V , and certificate expiration time T_{exp} . The vehicle encrypts the request message using the public key of RSU to avoid traceability by an attacker. Only the corresponding RSU will be able to decrypt this request message. After mutual authentication between vehicle and RSU, RSU computes a symmetric key for secure communication between the vehicle and RSU. After that, the vehicle uses this symmetric key to encrypt messages it sends to the RSU. When a vehicle moves from one RSU to another nearby RSU under its home RTA, it uses $Hash_{signRSU}$ signed by previous RSU to be authenticated by the nearby RSU. A vehicle can use any of the many existing pseudonyms changing techniques to change its pseudonym to protect privacy. The real identity of a vehicle cannot be traced, even if RSUs are compromised, because only RTA and TA know the real IDs of vehicles. RTA can reveal the real ID of a vehicle whenever it is found to be malicious.

Ensuring Message Integrity

In our protocol, RTA generates certificates for all vehicles registered under it. A newly registered vehicle uses this certificate to get a symmetric key and group key from an RSU. Later on, a vehicle uses $Hash_{signbyRSU}$ for authentication with a new RSU (under its home RTA) and receives a new symmetric key as well as a new group key for communication with that RSU. All messages from a vehicle to the RSU are encrypted using the symmetric key shared between the RSU and the vehicle. RSU can check the integrity and authenticity of the message using $Hash_{msgV}$. RSU disseminates this message by encrypting it with group key to other vehicles within its region. Vehicles can also check the integrity of the received message from RSU using $Hash_{msgRSU}$. Thus, it prevents the modification of messages by unauthorized or external attackers or internal attackers.

Resistance to Replay Attack

In case of a replay attack, attacker eavesdrops a transmitted message and re-transmits it several times to create confusion. In our protocol, vehicles append message generation timestamp with each message and encrypt messages together with their timestamp. RSU checks the message's freshness using the timestamp and disseminates it to other vehicles in its region, and its home RTA with encrypted message generation timestamp. Therefore, a replay attack is prevented for messages sent by vehicles as well as RSUs.

Non-repudiation

In our protocol, each message from the sender contains the sender's signature, and it also contains ID (PID in case of vehicles) of sender and message generation timestamp. A sender can never deny the action of sending the message because of the sender's signature. Besides, the sender cannot also deny the time of sending the message due to encrypted message generation timestamp. Although real ID of vehicles is never used for communication, RTA can track (based on PIDs) the real ID of a vehicle from the message whenever any dispute occurs. Thus, our protocol supports non-repudiation.

2.4.2 Scalability of the Proposed Protocol

Distributed and Scalable Architecture

Our architecture has two layers, the upper layer and the lower layer. The upper layer consists of TA and RTAs. Whereas, the lower layer consists of RSUs and vehicles. In our system, TA is the root, while each RTA acts as a lower-level local TA. Vehicles only need to store MHT root value and latest MHT root generation timestamp in the system to verify the authenticity of RSUs. On the other hand, the TA, RTAs, and RSUs use MMPT to store and search the public key of revoked vehicles efficiently. Messages are disseminated through RSUs after authentication. Our architecture is distributed and hence is scalable.

Reduced Message Transmission Overhead

In many of the existing schemes, when a vehicle observes an event, this event is disseminated to every vehicle within its transmission range, which in turn is forwarded to other vehicles using an underlying routing protocol. It can cause message transmission overhead due to the dissemination of the same event by several vehicles. However, in our scheme, a vehicle sends messages to authenticated RSUs for further dissemination. Only RSUs are responsible for disseminating messages to vehicles within its transmission range and other relevant RSUs through their home RTAs. So, RSUs can aggregate messages and prevent the propagation of redundant messages to vehicles.

Overhead Involved in Authentication and Message Dissemination

In our scheme, an MHT based RSU authentication scheme is used. Each RTA maintains an MHT of registered RSUs under it and sends missing hash values along with the latest MHT root generation timestamp to corresponding RSUs. A vehicle can authenticate an RSU by using this missing hash values and latest MHT root generation timestamp without storing CRLs of revoked RSUs. The TA, RTAs, and RSUs make use of an MMPT to store the public keys of revoked vehicles. In many of the existing works in VANETs, vehicles require enough storage to store CRLs. The size of CRLs increases as the number of revoked vehicles increase. It can cause significant computation and storage overhead for each vehicle. Our scheme reduces this overhead by not using CRLs; instead, it uses MHT and MMPT. Moreover, RSUs have more computational capability than vehicles. RSUs are responsible for authentication of vehicles within its transmission range and dissemination of messages to vehicles in its region and to its home RTA for further dissemination. *In addition to that, RSUs can aggregate duplicate messages before further dissemination. So, message dissemination is more scalable under our architecture.*

2.4.3 Proof of Correctness of our protocol using BAN Logic

Borrows, Abadi, and Needham (BAN) logic [19] has been widely used to formally verify the correctness of authentication protocols [67, 24, 33, 10, 62]. In this subsection, we present the formal verification of our proposed mutual authentication scheme using BAN logic and the PKI-based extended BAN Logic [96]. Firstly, we present a brief overview of the BAN logic in this subsection. Then, a formal idealization of the proposed scheme, the initial assumptions, goals, and logical derivation to achieve the goals using the inference rules for BAN logic in Appendix A are discussed.

BAN Logic Notation

Notations used in BAN logic to prove the correctness of our protocol are listed in Table 2.4.

Table 2.4: 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
$P \xleftrightarrow{S_k} Q$	P and Q share a secret key S_k
$\#(X)$	X is fresh
$\{X\}_k$	X is encrypted under the key k
$(X)_k$	X is hashed with the key k
$\wp\kappa(P, K_P)$	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

Protocol Idealization

The messages exchanged between RSU and Vehicle to achieve mutual authentication is idealized as follows:

- Simplified and idealized messages for mutual authentication between RSU_i and a newly registered vehicle V:
 M1: $V \rightarrow RSU_i : (PID_V, \{K_V, Cert_V(T_{exp}, (PID_V, K_V)), T_{s1}\}_{K_{RSU_i}})$
 M2: $RSU_i \rightarrow V : (ID_{RSU}, PID_V, \{\sigma(K_{RTA}, K_{TA}^{-1}), \sigma(MHTroot, K_{RTA}^{-1}), MHVs, RSU_i \xleftrightarrow{S_k} V, \sigma(\Re((K_V)_{S_k}, all), K_{RSU_i}^{-1}), G_k, T_{s2}\}_{K_V})$
- Simplified and idealized messages for mutual authentication when a vehicle moves from RSU_i to RSU_j :
 M3: $V \rightarrow RSU_j : (PID_V, \{K_{RSU_i}, RSU_i \xleftrightarrow{S_k} V, \sigma(\Re((K_V)_{S_k}, all), K_{RSU_i}^{-1}), T_{s3}\}_{K_{RSU_j}})$
 M4: $RSU_j \rightarrow V : (ID_{RSU}, PID_V, \{\sigma(K_{RTA}, K_{TA}^{-1}), \sigma(MHTroot, K_{RTA}^{-1}), MHVs, RSU_j \xleftrightarrow{S'_k} V, \sigma(\Re((K_V)_{S'_k}, all), K_{RSU_j}^{-1}), G'_k, T_{s4}\}_{K_V})$

Initial Assumptions

In this scheme, the public key of TA (K_{TA}) is distributed to all RTAs, RSUs, and Vehicles, and the private key (K_{TA}^{-1}) is stored securely. Every RTA generates the public and private key pairs for each RSU (K_{RSU}, K_{RSU}^{-1}) and vehicle (K_V, K_V^{-1}) registered under it. A newly registered vehicle gets a certificate ($Cert_V$) from its home RTA to take part in VANET communication, where the certificate has certificate expiration

time (T_{exp}) and the credential statement (st) (st contains PID_V and K_V). The initial assumptions of the protocol is summarized as follows:

- A1: $RTA| \equiv \wp\kappa(TA, K_{TA})$
- A2: $RTA| \equiv \Pi(K_{TA}^{-1})$
- A3: $RSU| \equiv \wp\kappa(TA, K_{TA})$
- A4: $RSU| \equiv \Pi(K_{TA}^{-1})$
- A5: $V| \equiv \wp\kappa(TA, K_{TA})$
- A6: $V| \equiv \Pi(K_{TA}^{-1})$
- A7: $RSU| \equiv \wp\kappa(RTA, K_{RTA})$
- A8: $RSU| \equiv \Pi(K_{RTA}^{-1})$
- A9: $V| \equiv \wp\kappa(RTA, K_{RTA})$
- A10: $V| \equiv \Pi(K_{RTA}^{-1})$
- A11: $V| \equiv Cert_V(T_{exp}, st)$
- A12: $V| \equiv \forall x RTA \Rightarrow Cert_x$
- A13: $RSU| \equiv \forall x RTA \Rightarrow Cert_x$
- A14: $RSU| \equiv \forall S RSU \Rightarrow RSU \xleftrightarrow{S_k} V$
- A15: $V| \equiv \forall S RSU \Rightarrow RSU \xleftrightarrow{S_k} V$
- A16: $RSU| \equiv RTA \Rightarrow MHTroot$
- A17: $V| \equiv RTA \Rightarrow MHTroot$
- A18: $RSU| \equiv \#(T_{s_i})$
- A19: $V| \equiv \#(T_{s_i})$

Goal of the Protocol

The goal of our protocol is to establish secure communication through mutual authentication between RSUs and Vehicles. The goals are illustrated as follows:

- G1: $RSU_i| \equiv K_V$
- G2: $V| \equiv MHTroot$
- G3: $V| \equiv RSU_i \xleftrightarrow{S_k} V$
- G4: $RSU_j| \equiv RSU_i \xleftrightarrow{S_k} V$

Logic Derivation

We provide a formal proof of our proposed scheme based on logical postulates in Appendix A and initial assumptions.

G1 can be deduced from M1 as follows:

- D1. $RSU_i \triangleleft (PID_V, \{ K_V, Cert_V(T_{exp}, (PID_V, K_V)), T_{s_1} \}_{K_{RSU_i}}) \text{ (From } R5(10) \text{)}$
- D2. $RSU_i \triangleleft Cert_V(T_{exp}, (PID_V, K_V)) \text{ (From D1 and } R5(8) \text{)}$
- D3. $RSU_i | \equiv \# (Cert_V(T_{exp}, (PID_V, K_V))) \text{ (From } R4 \text{ and } A18 \text{)}$
- D4. $RSU_i | \equiv RTA | \sim (Cert_V(T_{exp}, (PID_V, K_V))) \text{ (From } A7, A8, \text{ and } R1(1) \text{)}$
- D5. $RSU_i | \equiv RTA | \equiv (PID_V, K_V) \text{ (From D3, D4, and } R1(3) \text{)}$
- D6. $RSU_i | \equiv K_V \text{ (From } A13, D5, \text{ and } R3 \text{) (G1)}$

G2 and G3 can be deduced from M2 as follows:

- D7. $V \triangleleft (ID_{RSU}, PID_V, \{\sigma(K_{RTA}, K_{TA}^{-1}), \sigma(MHTroot, K_{RTA}^{-1}), MHVs, RSU_i \xleftarrow{S_k} V, \sigma(\mathfrak{R}((K_V)_{S_k}, all), K_{RSU_i}^{-1}), G_k, T_{s_2}\}_{K_V}) \text{ (From } R5(10) \text{)}$
- D8. $V \triangleleft (\sigma(K_{RTA}, K_{TA}^{-1}), \sigma(MHTroot, K_{RTA}^{-1}), RSU_i \xleftarrow{S_k} V, \sigma(\mathfrak{R}((K_V)_{S_k}, all), K_{RSU_i}^{-1})) \text{ (From D7 and } R5(10) \text{)}$
- D9. $V \triangleleft \sigma(K_{RTA}, K_{TA}^{-1}) \text{ (From D8 and } R5(8) \text{)}$
- D10. $V | \equiv \# (\sigma(K_{RTA}, K_{TA}^{-1})) \text{ (From } A17 \text{ and } R4 \text{)}$
- D11. $V | \equiv TA | \sim K_{RTA} \text{ (From } A5, A6, \text{ and } R1(1) \text{)}$
- D12. $V | \equiv \# \sigma(MHTroot, K_{RTA}^{-1}) \text{ (From } R4 \text{ and } A19 \text{)}$
- D13. $V | \equiv RTA | \sim MHTroot \text{ (From D11, A9, A10, and } R1 \text{ in (1))}$
- D14. $V | \equiv RTA | \equiv MHTroot \text{ (From D13, } R2, \text{ and } R4 \text{)}$
- D15. $V | \equiv MHTroot \text{ (From D14, } R4, \text{ and } A17 \text{) (G2)}$
- D16. $V \triangleleft (RSU_i \xleftarrow{S_k} V, \sigma(\mathfrak{R}((K_V)_{S_k}, all), K_{RSU_i})) \text{ (From D8 and } R5(8) \text{)}$
- D17. $V | \equiv RSU_i | \sim (RSU_i \xleftarrow{S_k} V, (K_V)_{S_k}) \text{ (From D16, } R5(9), \text{ and } R1(4) \text{)}$
- D18. $V | \equiv \# (RSU_i \xleftarrow{S_k} V, (K_V)_{S_k}) \text{ (From } R4 \text{ and } A19 \text{)}$
- D19. $V | \equiv RSU_i | \equiv (RSU_i \xleftarrow{S_k} V) \text{ (From D17, D18, } R2 \text{ and } R4 \text{)}$

- D20. $V \mid \equiv (RSU_i \xleftarrow{S_k} V)$ (From D19, A15 and R3) **(G3)**

G4 can be deduced from M3 as follows:

- D21. $RSU_j \triangleleft (K_{RSU_i}, RSU_i \xleftarrow{S_k} V, \sigma(\mathfrak{R}((K_V)_{S_k}, all), K_{RSU_i}^{-1}), T_{s_3})$ (From R5(10))
- D22: $RSU_j \triangleleft (RSU_i \xleftarrow{S_k} V, (K_V)_{S_k})$ (From R5(9))
- D23: $RSU_j \mid \equiv \# (RSU_i \xleftarrow{S_k} V, (K_V)_{S_k})$ (From A18 and R4)
- D24: $RSU_j \mid \equiv RSU_i \mid \sim (RSU_i \xleftarrow{S_k} V, (K_V)_{S_k})$ (From D21 and R1(4))
- D25: $RSU_j \mid \equiv RSU_i \mid \equiv RSU_i \xleftarrow{S_k} V$ (From D23, D24, R2 and R4)
- D26: $RSU_j \mid \equiv RSU_i \xleftarrow{S_k} V$ (From D25, A14 and R3) **(G4)**

G2 can also be achieved similarly from M4 using the following deduction.

- D27. $V \triangleleft (ID_{RSU_j}, PID_V, \{\sigma(K_{RTA}, K_{TA}^{-1}), \sigma(MHTroot, K_{RTA}^{-1}), MHVs, RSU_j \xleftarrow{S'_k} V, \sigma(\mathfrak{R}((K_V)_{S'_k}, all), K_{RSU_j}^{-1}), G'_k, T_{s_4}\}_{K_V})$ (From R5(10))
- D28. $V \triangleleft (\sigma(K_{RTA}, K_{TA}^{-1}), \sigma(MHTroot, K_{RTA}^{-1}), RSU_j \xleftarrow{S'_k} V, \sigma(\mathfrak{R}((P_V)_{S_k}, all), K_{RSU_j}^{-1}))$ (From D27 and R5(10))
- D29. $V \triangleleft \sigma(K_{RTA}, K_{TA}^{-1})$ (From D28 and R5(8))
- D30. $V \mid \equiv \# (\sigma(K_{RTA}, K_{TA}^{-1}))$ (From A17 and R4)
- D31. $V \mid \equiv TA \mid \sim K_{RTA}$ (From A5, A6, and R1(1))
- D32. $V \mid \equiv \# \sigma(MHTroot, K_{RTA}^{-1})$ (From R4 and A19)
- D33. $V \mid \equiv RTA \mid \sim MHTroot$ (From D31, A9, A10, and R1 in (1))
- D34. $V \mid \equiv RTA \mid \equiv MHTroot$ (From D33, R2, and R4)
- D35. $V \mid \equiv MHTroot$ (From D34, R4, and A17) **(G2)**

It shows that vehicles and RSUs can use the shared secret key and group key after mutual authentication through the above BAN analysis for secure communication.

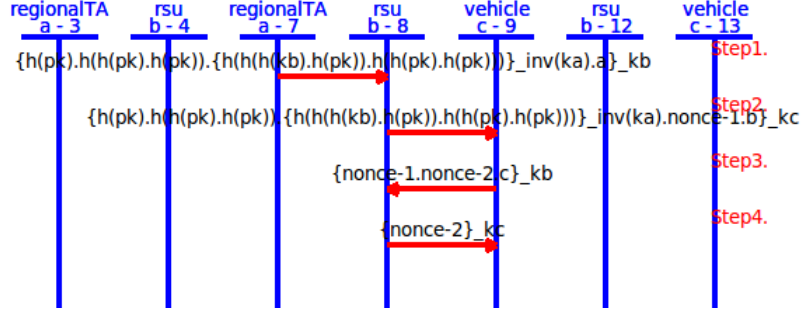


Figure 2.8: Message sequence chart of the proposed scheme using SPAN and AVISPA tools

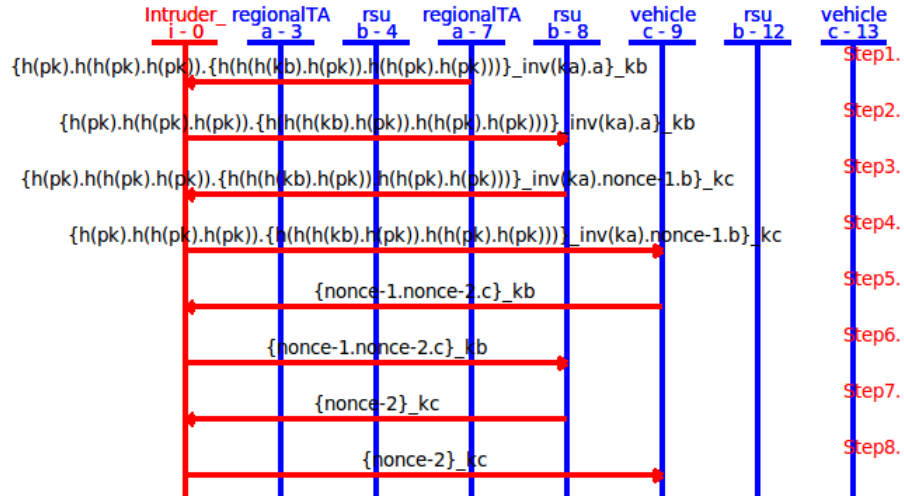


Figure 2.9: Message sequence chart of our scheme in the presence of an intruder

2.4.4 Proof of Correctness using Simulation Tools

SPAN (Security Protocol ANimator) [50] and AVISPA (Automated Validation of Internet Security Protocols and Applications) [7] tools are widely used in the literature [2, 6, 78] to verify the security of an authentication protocol against replay attack, man-in-the-middle attack, and impersonation attack. In this section, we present the results of verification of security of our protocol using SPAN and AVISPA tools.

In our model, we consider three basic roles which we call *regionalTA*, *rsu*, and *vehicle*, and are denoted by *a*, *b*, and *c* respectively. For verifying the security of our scheme, a scenario consisting of four RSUs under one RTA is considered. Here, *ka* represents the public key of the *regionalTA*, *kb* defines the public key of the *rsu* that is to be authenticated by *vehicle*, the public key of other registered RSUs under the *regionalTA* is represented by *pk*, *kc* denotes the public key of the *vehicle*, and *h* represents the cryptographic hash functions. In the case of the proposed protocol, *regionalTA* first activates the start signal and sends a message to *rsu* containing MHVs for the corresponding *rsu* and the root hash value which is encrypted with its private key *inv(ka)*. The MHVs are used by *vehicle* to authenticate the *rsu*. Message

sequence chart of the proposed scheme using SPAN and AVISPA tools is shown in Fig. 2.8.

The message sequence chart in Fig. 2.9 shows that the intruder is able to only listen and forward messages, but is unable to read and/or modify the messages. Sequence of exchanged messages in the presence of intruder i is described below:

- Step 1: The *regionalTA* initiates session and sends a message containing MHVs and root value (encrypted with its private key $inv(ka)$) to the *rsu*.
- Step 2: The intruder listens the message and passes it to the *rsu*.
- Step 3: The *rsu* sends the received message along with a nonce to *vehicle*.
- Step 4: The intruder views the message, but is unable to read as the message is encrypted using public key kc of *vehicle*. The intruder simply forwards the message to the *vehicle*.
- Step 5: Upon receiving the message, *vehicle* retrieves the root value using public key ka of the *regionalTA*. Next, it recalculates the root value using received MHVs in received message to verify the *rsu*. After that it sends back the received nonce along with another generated nonce to the *rsu*.
- Step 6: The intruder simply listens the message and forwards it to the *rsu*.
- Step 7: The *rsu* decrypts the message and sends back the received nonce to the *vehicle*.
- Step 8: The intruder views the message and passes it to the *vehicle*.

2.5 Performance Evaluation

In this section, we compare the computation and communication overhead of our protocol with that of ABAH [52] and LIAP [111].

2.5.1 Vehicle Authentication Overhead

In our scheme, each RSU, RTA, and the TA maintain an MMPT containing the public key of revoked vehicles. When an RSU receives an authentication request message from a vehicle, it looks up into the MMPT to check if the vehicle's public key PU_V is present. If so, then the vehicle is determined to be malicious, and drops the request message from the vehicle. For example, if N_V denotes the total number of the revoked vehicles, N_P denotes the total number of pseudonyms a vehicle holds, and N_{rev_V} denotes the size of the CRL, then the lookup operation in our scheme takes $O(\log N_V)$ to find a revoked vehicle. So, computational complexity involved in authenticating a vehicle in our approach is $O(\log N_V)$. On the other hand, each vehicle in LIAP [111] scheme holds a long term certificate. In this scheme, each RSU uses a linear search of vehicle certificate revocation list (VCRL) for authenticating vehicles. So, complexity involved in authenticating a vehicle in LIAP [111] is $O(N_{rev_V})$.

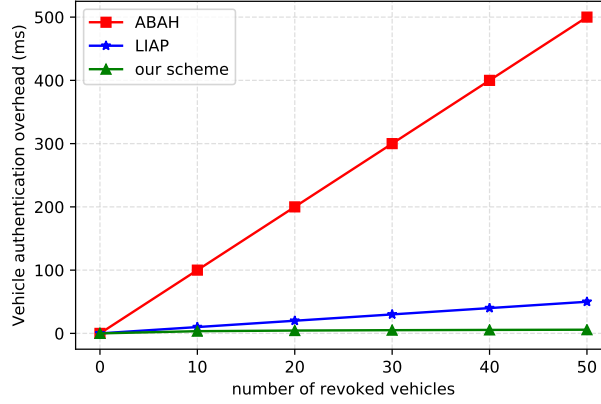


Figure 2.10: Vehicle Authentication Overhead Comparison

In ABAH [52], each vehicle obtains enough pseudonyms from TA for communication. RSUs check the revocation status of the vehicle using CRL to avoid communicating with a revoked vehicle. If the pseudonym is present in the CRL, then the message from the vehicle is dropped. However, when a malicious vehicle is found, then all of its pseudonyms should be revoked. Thus, the total number of revoked certificates under ABAH [52] is $N_{rev_V} = N_V \times N_P$. Besides, The European standard ETSI TS 102 867 recommends changing a pseudonym every five minutes while the American SAE J2735 standard recommends changing it every 120 s or 1 km, whichever comes last [35].

Fig. 2.10 shows that the total cost for vehicle authentication under both ABAH and LIAP increases linearly with increase in the number of revoked vehicles compared to our proposed scheme. We set $N_P=10$ in our comparison. The total computational cost involved in the authentication of a vehicle in ABAH is significantly higher than that of our scheme as well as LIAP.

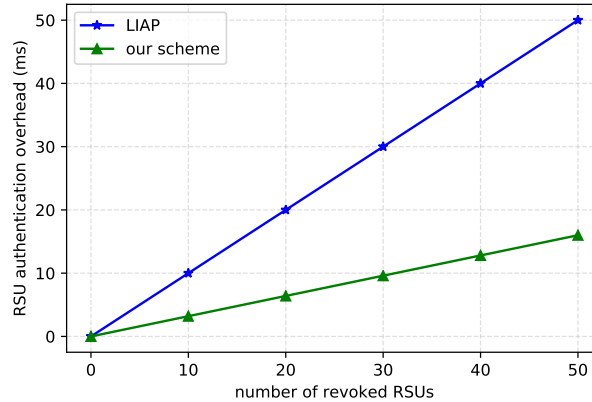


Figure 2.11: RSU Authentication Overhead comparison

2.5.2 RSU Authentication Overhead

In our scheme, when a vehicle receives the response message from an RSU, first it compares its stored value of $T_{mhtRoot}$ with that in the response message received from the RSU. If the received value of $T_{mhtRoot}$ is greater than or equal to the stored value of $T_{mhtRoot}$, then the vehicle verifies the signature of TA and RTA to retrieve the root of MHT from the received value. Next, the vehicle proceeds to validate the received root value of MHT. The time required for verifying RSA 2048 signature is 0.16 ms, and time required for computing SHA-256 hash is 111 MiB/s using Crypto++ 5.6.0 [31] that runs on an Intel Core 2 1.83 GHz processor under Windows Vista in 32-bit mode. Since the calculation of hash function incurs much less computation cost, the total cost of an RSU authentication depends on RSA signature verification. Moreover, in our scheme, computation and storage of certificates and CRL of RSUs are not required for authentication. A vehicle only needs to store the public key of the TA and the latest MHT root generation timestamp to authenticate RSUs. This approach is very scalable. On the contrary, a vehicle needs to check the RSU certificate revocation list (RCRL) using linear search to authenticate an RSU in LIAP [111]. In ABAH [52], RSUs periodically broadcast their certificates for authentication. If the certificate of an RSU is valid, then the vehicle successfully authenticates the RSU. However, they did not consider the management of revoked RSUs in their scheme. So, we consider only LIAP in Fig. 2.11 for comparing the overhead related to RSU authentication with that of our scheme.

Fig. 2.11 shows that for the LIAP, the RSU authentication cost significantly increases with the increase in the number of revoked RSUs. On the contrary, in our scheme, the authentication cost is considerably lower. For instance, when the number of revoked RSUs reaches 30, the authentication overhead under LIAP is almost three times that of our scheme.

Protocol Version	Type	Public Key	RSU ID	Signature	Timestamp
1 byte	1 byte	256 bytes	8 bytes	256 bytes	8 bytes

Figure 2.12: Revocation message format that RTA sends to RSUs

2.5.3 Communication Overhead

In both ABAH and LIAP, the TA(Trusted Authority)/CA (Certificate Authority) revokes the certificate of malicious vehicle and sends the updated CRLs of vehicles to RSUs. Generally, a CRL consists of a header, the current date, the date of the last update, the date of the next update, a complete list of revoked certificates which are signed by the certificate issuer [73]. Each revoked certificate in the CRL consists of 20 bits of certificate serial number and 48 bits of revocation date. A Certificate serial number is used to identify the revoked certificate. However, the CRL is expected to contain thousands of revoked certificates along with a CRL header of 51 bytes and

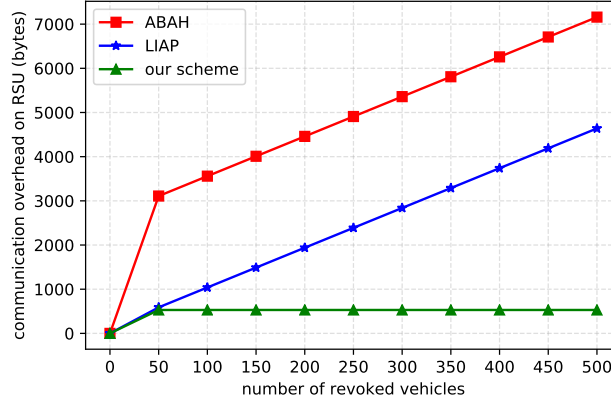


Figure 2.13: Comparison of communication overhead on RSU side

TA/CA’s signature of 60 bytes. It is found that CRLs are sent out very often, which tend to be very large. Thus, CRLs introduce additional significant communication overhead on RSU. In addition to that, congestion and overload of the network may occur when RSUs synchronously try to download the latest CRL to authenticate a vehicle. On the other hand, in our scheme, RTA sends a message to the RSUs containing the revoked vehicle’s public key and ID of the RSU. So, in our scheme, communication and computation overhead incurred by using CRLs for authentication is reduced. The format of the revocation message about a vehicle, sent by an RTA to the RSUs in its region, is given in Fig. 2.12.

Fig. 2.13 compares the communication overhead incurred by ABAH and LIAP (both of which use CRLs), with our scheme. As can be seen from this figure, the communication overhead on RSU in both ABAH and LIAP increases significantly as the number of revoked vehicles increases while our scheme has almost constant communication overhead (530 bytes) on RSU.

2.6 Summary

In this chapter, we presented a distributed and scalable privacy-preserving authentication and message dissemination scheme. Traditionally Certificates and CRLs were used for authenticating entities. However, as the number of entities grows, using CRLs for authentication incurs significant computation and communication overhead. Our protocol addresses this issue. In our scheme, a vehicle only needs to store the public key of the TA and the latest MHT root generation timestamp to authenticate RSUs. Similarly, MMPT is used by RSUs to authenticate vehicles, thus reducing the complexity involved in authenticating vehicles. Our scheme successfully replaced CRLs with two data structures MHT and MMPT, to reduce the communication and computation overhead related to the authentication of entities in VANET. We proved the correctness of our protocol using formal methods as well as simulation. We also analyzed the computational and communication overhead of our protocol and compared

it with that of two other protocols (ABAH and LIAP) presented in the literature. The proposed scheme outperforms the ABAH and LIAP schemes in terms of computation and communication overhead.

Chapter 3 Low Overhead Privacy-preserving Authentication Scheme in VANETs

In this Chapter, we present a cuckoo filter-based [37] lightweight authentication scheme for VANETs. The cuckoo filter contains only one entry for all pseudonyms of a revoked vehicle, thereby minimizing the overhead associated with CRL verification. Our scheme provides an efficient lookup operation for both vehicles and Road Side Units (RSUs) in a Vehicle to Infrastructure (V2I) scenario. Security analysis and verification demonstrate that our protocol is robust against man-in-the-middle attacks, replay attacks, and impersonation attacks. Performance evaluation also shows that our scheme has a significantly lower authentication overhead than other related schemes.

3.1 Background

3.1.1 Cuckoo Filter

Cuckoo filter (CF) [37] is a probabilistic data structure that uses a cuckoo hash table to speed up the set-membership test. Cuckoo filter stores fingerprint $F(x)$ of an element x instead of storing the original element x to improve space efficiency. A basic cuckoo filter consists of a set of m buckets where each bucket stores n number of entries. Each element x has two candidate buckets i and j determined by two hash functions as follows:

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

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

A sample cuckoo filter with eight buckets ($m = 8$), where each bucket has four entries ($n = 4$) is shown in Fig. 3.1(a). The cuckoo filter calculates the fingerprint $F(x)$ of x to insert an element x . Two hash functions ($H_1(x)$, $H_2(x)$) are used to find the candidate buckets in the filter. If any of the two candidate buckets is free, it inserts $F(x)$ in the free bucket. If both of the candidate buckets are occupied,

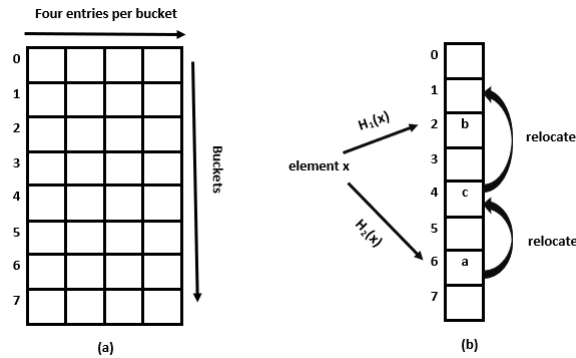


Figure 3.1: (a) A cuckoo filter (b) Insertion operation.

it randomly selects one of the buckets in the cuckoo filter and replaces the existing fingerprint $F(y)$ of the element y in the bucket with $F(x)$. Later on, inserts the $F(y)$ in its alternate candidate bucket. If the bucket is not free, a fingerprint in one of the buckets is displaced and repeats the above process until all the displaced fingerprints are re-inserted in the cuckoo filter. We can find the alternate bucket location by XORing the hash of the fingerprint of the element and its current bucket location. In Fig. 3.1(b), while inserting an element x , it finds that both the candidate buckets (bucket number 2 and bucket number 6) are occupied. Next, selects bucket number 6 randomly and replaces the existing element ("a") in the bucket with $F(x)$. The cuckoo filter relocates the kicked out element ("a") in its alternate candidate bucket number 4 by displacing the existing item ("c") in that bucket. After that, it re-inserts the kicked-out item ("c") in its alternate candidate bucket 1. To lookup, an element x , first, it computes the fingerprint of the element $F(x)$. Then, finds the two candidate buckets using $H_1(x)$ and $H_2(x)$, and checks the fingerprint $F(x)$ with the fingerprints stored in these buckets. If any of the fingerprints match $F(x)$, then the cuckoo filter returns a positive result. Otherwise, it returns a negative result. To delete an element x , firstly, the fingerprint $F(x)$ is searched using the lookup operation. When $F(x)$ is found in one of the buckets, deletes $F(x)$ from the cuckoo filter. The cuckoo filter has a time complexity of $O(1)$ for both lookup and delete operations.

3.2 System Model

In this section, we describe our system model, design goals, and initial assumptions.

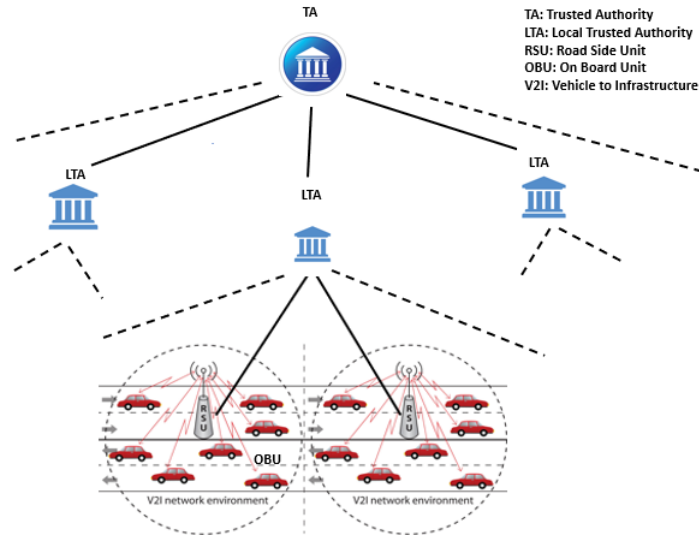


Figure 3.2: Proposed system model for V2I authentication in VANET.

3.2.1 System Model

Our system model depicted in Fig. 3.2 has four types of entities: Trusted Authority (TA), Local Trusted Authorities (LTAs), Road Side Units (RSUs), and Vehicles.

- **Trusted Authority(TA):** TA is the root of the entire system. TA generates its public and private key pairs (PU_{TA}, PR_{TA}) . Each entity knows the public key of TA PU_{TA} , while the private key PR_{TA} is kept secret. TA also generates the public and private key pairs (PU_{LTA}, PR_{LTA}) for each LTA registered with it.
- **Local Trusted Authority (LTA):** Each LTA acts as a local authority and is registered with the TA. Each LTA is responsible for registering and generating a pair of private and public keys for each RSU and vehicle registered under it.
- **Road Side Unit (RSU):** RSUs have a wider communication range than vehicles. RSUs are responsible for receiving and relaying messages from vehicles within their transmission range.
- **Vehicle:** Each vehicle registers using its real identity with its home LTA in VANET communication. The vehicle's On-Board-Unit (OBU) is loaded with a set of pseudonyms when it registers with an LTA. A vehicle uses its pseudonym for communication and changes its pseudonym frequently using an underlying algorithm to ensure message unlinkability. Vehicles can move from one LTA's region to another's region. Once a vehicle enters a new LTA's region, the vehicle receives a new set of pseudonyms to communicate in the new region. It is assumed that the LTAs in the different regions can securely communicate with each other.

3.2.2 Design Goals

Our protocol has the following design goals:

- **Privacy preservation:** In VANETs, privacy is a major concern. Thus even if all RSUs are compromised, it is necessary to protect the real identity of the vehicle. Only the LTA under which the vehicle is registered knows the real identity of vehicles in our proposed scheme. In the case of malicious vehicles, the LTA can reveal the real identity to ensure secure communication.
- **Low authentication overhead:** Both vehicles and RSUs authenticate each other before VANET communication in our scheme. Our scheme leverages cuckoo filters to reduce the overhead related to using CRLs for authentication.
- **Resilience to message tampering:** Our goal is to ensure that messages come from a legitimate source and messages are transmitted in a secure way.

Table 3.1: Notation and Description of abbreviation used in this Chapter 3

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
t_s	Message generation timestamp
E	Encryption algorithm
H	SHA-256 hash function
F	Fingerprint
CF	Cuckoo filter
B_{ndx}	Bucket index in positive cuckoo filter PCF_V
MHT	Merkle Hash Tree
MFV_s	Missing Fingerprint Values of MHT for 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

- **Resilience to message replay attack:** Attackers re-transmits the eavesdropped message several times to create confusion. In our protocol, both vehicles and RSUs append the message generation timestamp to each message to resist the message replay attack.

3.2.3 Assumptions

The necessary assumptions include:

- TAs possess more computing, communication, and storage capacity than LTAs; LTAs possess more computing, communication, and storage capacity than RSUs; RSUs possess more computing, communication, and storage capacity than OBUs.
- Each LTA maintains separate MHT of the set of pseudonyms assigned to each vehicle in its region.
- We assume that the clocks of TA, LTAs, RSUs, and vehicles are loosely synchronized (GPS can be used for this).

3.3 Our Proposed Scheme

Each vehicle registers with its home LTA to take part in VANET communication. Upon registration, each vehicle's OBU is loaded with a set of pseudonyms covering a fixed longer period, such as two weeks, one month, or three months. A pseudonym with a shorter lifetime provides higher unlinkability and more privacy. Our scheme leverages Merkle Hash Tree (MHT) and Cuckoo filter for efficient authentication of vehicles. The notations used in this chapter are listed in Table 3.1.

3.3.1 Construction of Merkle Hash Tree

In our scheme, each vehicle constructs an MHT of its set of pseudonyms. Fig. 3.3 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 [86] to retrieve the missing fingerprint values (MFVs) associated with a pseudonym of the vehicle PID_V to recalculate the MHT_{root} for verification. The table 3.2 presents the MFVs corresponding to each pseudonym of the vehicle in Fig. 3.3. 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.

3.3.2 Construction of Cuckoo Filters for Vehicle Authentication

Each LTA constructs a positive cuckoo filter (PCF_V) and a negative cuckoo filter (NCF_V) to facilitate the authentication of vehicles in its region. The insert, lookup, and delete operations on PCF_V and NCF_V occur in the following circumstances:

When a vehicle registers with its home LTA

The LTA generates a set of pseudonyms and constructs Merkle Hash Tree (MHT) of these pseudonyms for each vehicle registered with it. LTA also computes the finger-

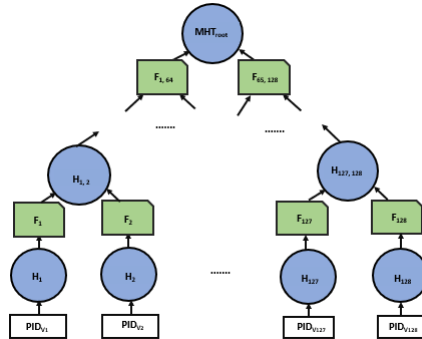


Figure 3.3: A sample Merkle Hash Tree of pseudonyms of a vehicle.

print of MHT root value $F(MHT_{root})$ corresponding to every valid vehicle registered under it and inserts the $F(MHT_{root})$ into PCF_V .

When the $Exp_{MHT_{root}}$ expires

The LTA uses the $Exp_{MHT_{root}}$ to check the priority queue periodically to determine if there is an expired set of pseudonyms associated with a vehicle's MHT_{root} in its region. The priority queue is sorted by the $Exp_{MHT_{root}}$. When the MHT_{root} value corresponding to a vehicle expires, the LTA removes the expired MHT_{root} from the priority queue. Next, the LTA looks up for $F(MHT_{root})$, and deletes it from both PCF_V and NCF_V .

When an LTA finds a malicious vehicle in its region

When a misbehaving vehicle is found in an LTA's region, a misbehaving report is sent to the LTA. The misbehaving report contains the pseudo ID of the sender $PID_{V_{sender}}$ and pseudo ID of the misbehaving vehicle $PID_{V_{malicious}}$. On receiving the misbehaving report, the LTA firstly verifies the validity of $PID_{V_{sender}}$. If the LTA determines the misbehaving vehicle as malicious, it deletes the $F(MHT_{root})$ of the vehicle from the positive cuckoo filter PCF_V . Many methods have been proposed for identifying a vehicle as malicious. LTA can use any of them to detect a malicious vehicle. After that, LTA inserts the $F(MHT_{root})$ of the malicious vehicle into the negative filter NCF_V .

When a vehicle leaves an LTA's region

A vehicle sends a message that includes its current pseudo ID PID_v , MHT root of its pseudonyms MHT_{root} , and the bucket index B_{ndx} for $F(MHT_{root})$ in positive cuckoo filter before leaving an LTA's region. When the LTA receives the message, it verifies the credentials of the vehicle and deletes the vehicle's corresponding $F(MHT_{root})$ from PCF_V .

Table 3.2: 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}	$F_{127}, F_{123,126}, F_{121,124}, F_{113,120}, F_{97,112}, F_{65,96}, F_{1,64}$

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. The request message contains the current pseudo ID of the vehicle PID_v , 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 positive cuckoo filter PCF_V .

3.3.3 Construction of Cuckoo Filters for RSU Authentication

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 a negative cuckoo filter NCF_{RSU} . LTA also inserts the fingerprint of public key of every new RSU PCF_{RSU} installed under its region in a positive cuckoo filter PCF_{RSU} .

3.4 V2I Authentication Phase

In our proposed authentication scheme, the LTA uses positive cuckoo filter PCF_V to store the fingerprint of MHT root $F(MHT_{root})$ corresponding to every valid vehicle in its region. The LTA also initializes a negative cuckoo filter NCF_V to store the fingerprint of MHT root $F(MHT_{root})$ of malicious vehicle in its region. Each LTA maintains a positive cuckoo filter PCF_{RSU} of valid RSUs and a negative cuckoo filter NCF_{RSU} of compromised RSU in its region for efficient authentication. LTA signs the Cuckoo Filters $Sig_{LTA}(PCF_{RSU}, NCF_{RSU})$ and $Sig_{LTA}(PCF_V, NCF_V)$, and periodically broadcasts both the positive and negative cuckoo filter. Vehicles and RSUs use the latest Cuckoo Filters to authenticate each other in V2I communication.

In our scheme, an RSU periodically broadcasts a beacon message. The beacon message contains the ID of RSU ID_{RSU_i} , public key of the RSU PU_{RSU_i} , and message generation timestamp t_s . When vehicle V_i receives the beacon message, it first checks t_s for freshness. After that, V_i computes the fingerprint of PU_{RSU_i} and looks up into the PCF_{RSU} and NCF_{RSU} . V_i verifies the validity of PU_{RSU} using the **Algorithm 4**.

After authenticating the PU_{RSU_i} , V_i sends a message to the 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 . When RSU_i receives the message from V_i , it first decrypts the message using its private key PR_{RSU_i} . Next, it checks the freshness of the message using t_s . If the message is fresh, then the RSU_i recalculates the MHT

Algorithm 4: V_i authenticates PU_{RSU_i}

```
1 RSU periodically broadcasts beacon message  $(ID_{RSU_i}, PU_{RSU_i}, t_s)$ 
2 Upon receiving the beacon message  $V_i$  verifies  $PU_{RSU_i}$  as follows
3 Checks the freshness of the received message using  $t_s$ ;
4 if  $t_s$  is valid then
5   Looks up into the positive CF  $PCF_{RSU}$  and negative CF  $NCF_{RSU}$  for
    $F(PU_{RSU_i})$ ;
6   if  $F(PU_{RSU_i}) \in PCF_{RSU}$  then
7     if  $F(PU_{RSU_i}) \notin NCF_{RSU}$  then
8        $PU_{RSU_i}$  is considered valid;
9     else
10       $PU_{RSU_i}$  sends message to LTA for verification ;
11    end if
12  end if
13  if  $F(PU_{RSU_i}) \notin PCF_{RSU}$  then
14    if  $F(PU_{RSU_i}) \in NCF_{RSU}$  then
15       $PU_{RSU_i}$  is considered malicious;
16    else
17       $V_i$  waits for updated CFs from LTA;
18    end if
19  end if
20 else
21   Drops the beacon message;
22 end if
```

root value corresponding to the V_i using the received $MFVs$ and PID_{V_i} . After that, RSU_i checks if the fingerprint of the MHT root value $F(MHT_{root})$ exists in the bucket index B_{ndx} in the positive cuckoo filter PCF_V and $F(MHT_{root})$ exists in the negative cuckoo filter NCF_V . A detailed description of the authentication process is presented in **Algorithm 5**.

Table 3.3 presents the four possible cases of the query results of positive and negative cuckoo Filter(CF). A receiver determines whether the sender is valid or not using the results of both positive and negative cuckoo filters in Table 3.3.

3.5 Performance Analysis and Comparison

In this section, firstly, we analyze the security of our proposed authentication scheme. Next, we verify the security of our scheme against replay attack, man-in-middle attack, and impersonation attack using SPAN (Security Protocol ANimator) [50] and AVISPA (Automated Validation of Internet Security Protocols and Applications) [7] tools. After that, we compare our scheme's authentication overheads with that of LIAP [111], NERA [9], and HDMA [109] schemes.

Algorithm 5: Authentication of V_i by RSU_i

```
1  $V_i$  sends  $E((PID_{V_i}, MFVs, B_{ndx}, PU_{V_i}, ID_{RSU_i}, t_s), PU_{RSU_i})$  to  $RSU_i$ 
2 Upon receiving the above message  $RSU_i$  verifies  $V_i$  as follows
3 Decrypts the message using private key  $PR_{RSU_i}$ ;
4 Checks the freshness of the received message using  $t_s$ ;
5 if  $t_s$  is valid then
6   Calculates the  $MHT_{root}$  value using MFVs and  $PID_{V_i}$ ;
7   Lookup into the positive cuckoo filter  $PCF_V$  with  $B_{ndx}$  and negative
   cuckoo filter  $NCF_V$  for  $F(MHT_{root})$  corresponding to  $V_i$  ;
8   if  $F(MHT_{root}) \in PCF_V$  with  $B_{ndx}$  then
9     if  $F(MHT_{root}) \notin NCF_V$  then
10       $V_i$  is considered valid;
11    else
12       $RSU_i$  sends message to LTA for verification ;
13    end if
14  end if
15  if  $F(MHT_{root}) \notin PCF_V$  with  $B_{ndx}$  then
16    if  $F(MHT_{root}) \in NCF_V$  then
17       $V_i$  is considered malicious;
18    else
19       $RSU_i$  waits for updated CFs from LTA;
20    end if
21  end if
22 else
23   Drops the received message;
24 end if
```

3.5.1 Security analysis

Mutual authentication:

In our scheme, both vehicles and RSUs authenticate each other using cuckoo filter before VANET communication. The V2I mutual authentication makes VANET communication more secure.

Table 3.3: Possible Cases and Results with Cuckoo Filter

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 LTA for verification

Identity anonymity:

Vehicles use pseudonyms for communication in our scheme. A vehicle's home LTA only knows its real ID. Therefore, even if all RSUs are compromised, the real ID of a vehicle is still secure.

Resistance to message modification attack:

Vehicles authenticate the public key of the RSUs received in the beacon message. Then, vehicles encrypt messages using the public key of RSU for mutual authentication. Since the attackers do not know the private key of RSUs, only the intended RSUs can read/modify the messages.

Resistance to replay attack:

In our scheme, both vehicles and RSUs append message generation timestamps with the messages. Upon receiving the messages, the receivers first check the message generation timestamp to resist replay attack. We assume that the clocks of RSUs and vehicles are loosely synchronized (this can be done using GPS).

3.5.2 Verification using SPAN and AVISPA tools

We verify the security of our protocol using SPAN and AVISPA tools [7]. 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. Fig. 3.4 presents the message sequence chart of our proposed scheme generated by SPAN and AVISPA tools.

Fig. 3.5 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.

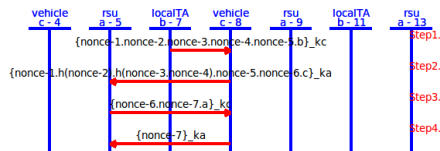


Figure 3.4: Message sequence chart of our scheme using SPAN and AVISPA tools.

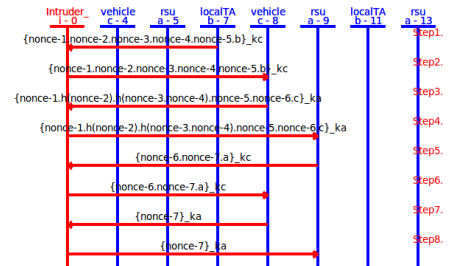


Figure 3.5: Message sequence chart of our scheme in the presence of an intruder.

The intruder only listens and forwards the messages. We describe the sequence of exchanged messages in the presence of an intruder i as follows:

- Step 1: The *localTA* initiates 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 up 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 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*.

3.5.3 V2I Authentication Overheads Analysis

In our scheme, a vehicle looks up into the PCF_{RSU} and NCF_{RSU} to verify the public key of RSU received in RSU's beacon message. After that, the vehicle sends a message to the RSU that includes the PID_V , *MFVs*, and B_{ndx} in PCF_V . RSU recalculates the MHT_{root} using the *MFVs* and looks up into the cuckoo filter with B_{ndx} in PCF_V for authentication. On the other hand, both vehicles and RSUs check Certificate Revocation Lists (CRLs) to authenticate each other in LIAP [111]. In NERA [9], RSUs check the CRL that contains the real ID of malicious vehicles to authenticate vehicles. In both LIAP [111] and NERA [9], bilinear pairing and Map-To-Point operations are used, which is costly. RSUs look up whether the pseudonym of vehicles is added in the CRL in HDMA scheme [109] for V2I authentication. A comparison of the authentication overheads of our scheme with LIAP, NERA, and HDMA schemes is presented in Table 3.4. In Table 3.4, T_{RSA-E} represents the time for RSA-1048 bit encryption operation, T_{RSA-D} refers the decryption operation time of the RSA-1048 bit algorithm, T_{RSA-V} represents RSA 1048 Verification time, SHA-256 hash operation time is represented by H , n denotes the number of pseudonyms assigned to a vehicle, T_{MUL} denotes the time for performing one point multiplication, T_{MTP} denotes the time for performing a MapToPoint hash operation, and T_{PAR} denotes the time for

Table 3.4: V2I Authentication Overheads

Schemes	Vehicle's Side	RSU's Side
LIAP	$T_{MUL}+T_{MTP}+3T_{PAR}$	$T_{MUL}+T_{MTP}+3T_{PAR}$
HDMA	$T_{RSA.V}+T_{RSA.D}$	$T_{RSA.V}+T_{RSA.D}+T_{RSA.E}$
NERA	$T_{MUL}+T_{MTP}+3T_{PAR}$	$T_{MUL}+T_{MTP}+3T_{PAR}$
Proposed authentication scheme	$T_{RSA.E}$	$(\log n + 1)H + T_{RSA.D}$

performing a pairing operation. In both LIAP [111] and NERA [9], computation overhead is $T_{MUL}+T_{MTP}+3T_{PAR}$ (where, $T_{MUL}= .39$ ms, $T_{MTP}=.09$ ms and $T_{PAR}=3.21$ ms). The required calculation time for $T_{RSA.E}=.08$ ms, $H=111$ MiB/s, $T_{RSA.D}=1.46$ ms, and $T_{RSA.V}=.07$ ms using Crypto++ 5.6.0 [31] that runs on an Intel Core 2 1.83 GHz processor. Fig. 3.6 shows the comparison of authentication overhead of vehicles on RSU in the proposed scheme with that of LIAP [111], NERA [9], and HDMA [109]. There is a significant increase in computation cost with the increased number of vehicles in both LIAP and NERA. On the contrary, the vehicle authentication cost is low under HDMA and very low under the proposed authentication scheme. It is observed in Fig. 3.7 that our proposed authentication scheme has significantly lower RSU authentication overhead compared to LIAP, NERA, and HDMA. For example, when the number of RSUs reaches 30, the computation cost is approximately 115 ms for both LIAP [111] and NERA [9] and 45 ms for HDMA [109], whereas it is only 2.4 ms for the proposed authentication scheme.

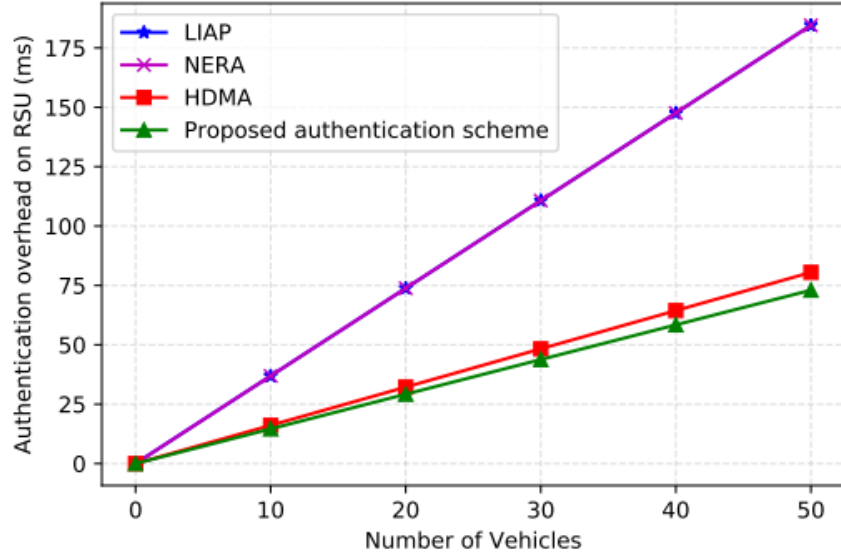


Figure 3.6: Computation overhead on RSU for authentication.

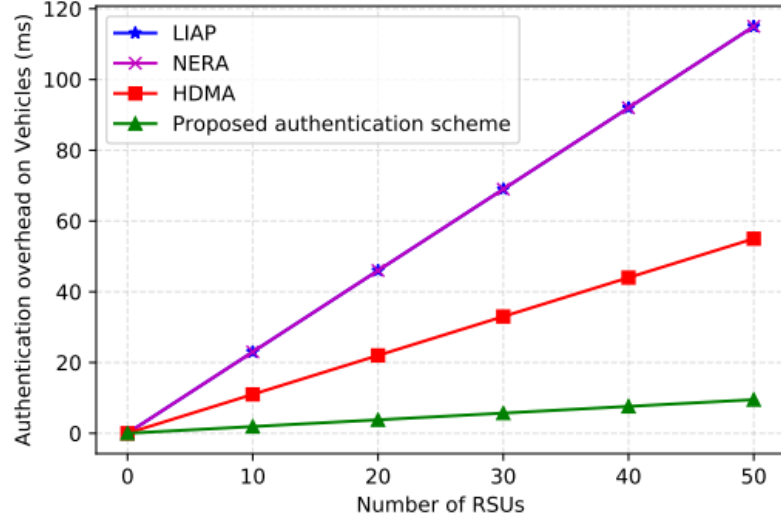


Figure 3.7: Computation overhead on vehicle for authentication.

3.6 Related Works

The necessity of removing compromised or misbehaving entities from the VANET is widely accepted because such entities can compromise communication security and harm transportation efficiency. Many of the existing privacy-preserving authentication schemes [71, 118, 51, 142] in VANET use Certificate Revocation Lists to store the information of revoke entities. However, the size of CRL increases significantly with the increased number of revoked entities, which adds computation and communication overhead for verification. A vehicle requires 43,800 pseudonyms annually (3,600 pseudonyms per month) if it changes its pseudonym every minute *et al.*'s [88]. When a vehicle is found to be malicious, certificates of all the unexpired pseudonyms of a vehicle are added to CRL. Therefore, the size of CRL grows linearly with the number of revoked pseudonyms.

Zhu *et al.* [142] proposed a privacy-preserving authentication scheme based on group signature [25] to address the computational and storage overhead caused by CRLs. They divided a region into several domains, and an RSU is responsible for authenticating vehicles in its domain. Vehicles receive group keys for hash message authentication code (HMAC) computation to facilitate message authentication. This scheme assumes RSUs are trustworthy. However, malicious entities can pose as RSUs to steal sensitive information from vehicle owners [74].

Wang and Yuo [111] proposed a local identity-based anonymous message authentication protocol (LIAP) for VANET. In this scheme, CA (Certificate Authority)

generates long-term certificates for both vehicles and RSUs to facilitate mutual authentication. CA also maintains and distributes the Certificate Revocation List of RSUs (RCRL) and vehicles (VCRL). Each vehicle and RSU checks RCRL and VCRL for V2I authentication. However, the CRL checking process causes computation and storage overheads. Wang *et al.* [109] proposed a hybrid D2D message authentication (HDMA) scheme for 5G enabled VANETs. In this scheme, each vehicle gets a set of pseudonyms and corresponding certificates from CA. When a vehicle enters an RSU's or RSBS's (Road Side Base Station) region, it sends a pseudonym certificate to the RSU or RSBS for authentication. The RSU or RSBS lookups CRL list to check whether the pseudonym has been revoked. A zero-knowledge proof algorithm is used in this scheme for V2I authentication. Bayat *et al.* [9] proposed an efficient RSU-based authentication scheme using bilinear pairing and Map-To-Point operation. In this scheme, a vehicle joins an RSU's region to take part in VANET communication. RSU generates a set of pseudo-IDs and the corresponding secret keys for each vehicle within its region after the mutual authentication. In this scheme, only the real IDs of malicious vehicles are added to CRLs, reducing the size of CRLs compared to traditional CRLs. However, message generation timestamps are not attached to every message in this scheme. As a result, it is susceptible to replay attacks.

In recent years, many of the research works[29, 60, 80, 122] on VANET use cuckoo filter (CF) [37] due to its efficient computational performance. Cui *et al.* [28] proposed an edge computing-based message authentication scheme in VANET using a cuckoo filter. In this scheme, RSU selects a number of vehicles as edge computing vehicles (ECVs) in its region. The ECVs assist RSU in authenticating the message signature sent by different vehicles within the RSU's region. RSU verifies the authentication result received from the ECVs. A cuckoo filter is then used to broadcast the authentication information to all vehicles in the RSU's region. Vehicles only need to query the cuckoo filter to verify the authenticity of a received message, which reduces the authentication overhead. Zhang *et al.* [132] proposed an efficient pseudonym certificate revocation scheme in VANET using a cuckoo filter. In this scheme, CA stores the fingerprints of the certificates of all unexpired pseudonyms of all revoked vehicles in a cuckoo filter. CA broadcasts the cuckoo filter to the network to facilitate the authentication process. This scheme significantly reduces the computation overhead compared to the traditional CRLs. Since the CA puts all unexpired pseudonyms into a cuckoo filter, this can add storage and computation overhead.

Motivated by the above drawbacks of the authentication schemes, we propose a lightweight privacy-preserving authentication scheme using cuckoo filters in VANETs. In our scheme, only a single entry (MHT_{root} corresponding to a vehicle's set of pseudonyms) for all pseudonyms of a revoked vehicle is placed into the cuckoo filter, which significantly reduces the overhead for verification related to CRL. Our scheme provides an efficient lookup operation for both vehicles and RSUs in V2I authentication.

3.7 Summary

In this chapter, we presented a lightweight privacy-preserving V2I authentication scheme based on cuckoo filters. The use of cuckoo filters reduces the storage, communication, and computation overhead in contrast to Certificate Revocation Lists (CRLs) based solution for authentication. We showed that our proposed authentication scheme meets the basic security requirements in VANETs through security analysis and verification. Performance evaluation of the proposed scheme shows that it performs better than some of the other related schemes.

Chapter 4 Efficient Management of Pseudonyms for Privacy-preserving Authentication in VANETs

In this chapter, we present CREASE, a Certificateless and REused-pseudonym based Authentication Scheme for Enabling Privacy in VANETs that uses Merkle Hash Tree (MHT) and Modified Merkle Patricia Trie (MMPT) to store and manage vehicles' pseudonyms efficiently. One significant advantage of our proposed scheme is that it enables all vehicles to change their pseudonyms frequently based on a pseudonym expiration time and to reuse their pseudonyms in a secure way. We also propose an efficient pseudonym status update mechanism that speeds up the pseudonym changing frequency to avoid traceability. A detailed security analysis is carried out to demonstrate the effectiveness of our scheme. SPAN and AVISPA tools are used to verify the security of our proposed protocol against replay attack, man-in-the-middle attack, and impersonation attack. Besides, a formal security proof using BAN logic is presented. Results show that our scheme outperforms contemporary NERA [9], ASPA [4], and LIAP [111] schemes in terms of security features and protocol overhead.

4.1 Related Works

The security and privacy problems in VANETs have attracted great attention from researchers in both academia and industry. Many privacy preserving authentication schemes that include pseudonym-based schemes, group signature-based schemes, ID-based schemes, symmetric cryptography-based schemes, and anonymous certificate based schemes have been proposed in recent years.

The pseudonym-based schemes mostly use Public Key Infrastructure (PKI). Raya and Hubaux [88] proposed a pseudonymous scheme, where the CA generates public-private key pairs and corresponding certificates for vehicles. In this scheme, each vehicle requires to preload a huge quantity of public-private key pairs and corresponding public-key certificates. Next, the vehicle selects one of the certificates to sign a message in each communication. This scheme provides message authentication and conditional privacy-preservation. However, a huge storage space is needed to store keys and corresponding certificates of all vehicles, while the certificate authority also needs to store certificates of all vehicles.

Jiang *et al.* [52] proposed an anonymous batch authentication scheme (ABAH) based on hash message authentication code (HMAC). In this scheme, they divide a large area into several domains, and each RSU manages the vehicles in its domain in a localized manner. TA generates enough pseudonyms for each vehicle to take part in VANET. Vehicles use pseudonym to send a join request to an RSU. The RSU checks the revocation status of the vehicle's pseudonym in the CRL for authentication. Next, the RSU sends a group key to the authentic vehicle. In this scheme, vehicles calculate HMAC using the group key and include it in the safety-related messages for secure

communication. However, the revocation status checking process using the CRL of vehicles incurs overhead.

Wang and Yuo [111] proposed a local identity-based anonymous message authentication protocol (LIAP) using bilinear pairing. In this scheme, both vehicles and RSUs get long-term certificates from CA (Certificate Authority) during registration. A vehicle uses its long-term certificate when it enters an RSU's region for authentication. The RSU checks the stored certificate revocation list of vehicles (VCRL) to authenticate the vehicle. Similarly, vehicles also authenticate the RSU using the certificate revocation list of RSUs (RCRL). After mutual authentication, vehicles get keys from RSUs to generate pseudonyms for V2V communication. However, the CA still needs to distribute the certificate revocation list of RSUs (RCRL) and vehicles (VCRL) in this scheme.

Paruchuri and Durresi [83] proposed a certificate-based scheme that uses smart cards to provide anonymous authentication. The smart card stores vehicle's real identity, certificate, and required cryptographic keys. In this scheme, a vehicle uses its certificate to authenticate itself to an RSU for receiving and sending messages. The RSU generates a session key and sends it to all the vehicles that have been authenticated by it. Authentic Vehicles under an RSU share the same session key for communication. The Vehicles do not need to store the computation-intensive CRLs. However, they did not discuss how the RSUs share the information of misbehaving vehicles to verify the authenticity of the vehicles.

Ali *et al.* [4] proposed a pseudonym-based authentication scheme that allows vehicles with a valid pseudonym for communication. In this scheme, firstly, a vehicle gets an initial pseudonym from Vehicular Manufacturing Company (VMC) using the VMC's pre-loaded secret key. Next, it gets a long term certificate (LTC) from the Certificate Authority (CA), which is used by the LTC Authority to issue a Pseudonym Certificate (PC) for the vehicle. Then, a vehicle requests for pseudonyms from the Pseudonym Provider (PP) directly or through RSUs. PP sends multiple pseudonyms to the vehicle and they are all valid at the same time interval. However, this scheme is not secure against Sybil attacks as multiple pseudonyms of a vehicle are valid at the same time interval. Besides, CA checks the credentials of the vehicle in CRL for authentication.

Most of the existing pseudonym based schemes [79, 66, 127, 124] mainly focus on the frequency of pseudonym change that is most effective, or the best situation for changing pseudonyms. To the best of our knowledge, existing schemes did not address the efficient management of pseudonyms while preserving privacy for authentication. Our scheme provides an efficient privacy-preserving authentication scheme that considers the efficient management of pseudonyms to address the challenges mentioned above. In CREASE, each RSU maintains an MHT combined with MMPT to store the vehicles' pseudonym efficiently and allow vehicles to reuse pseudonyms securely from their stored set of pseudonyms. Moreover, our scheme helps in replacing certificate revocation lists (CRLs) with two data structures, Merkle Hash Tree (MHT) and Modified Merkle Patricia Trie (MMPT), to reduce the overhead involved in authentication.

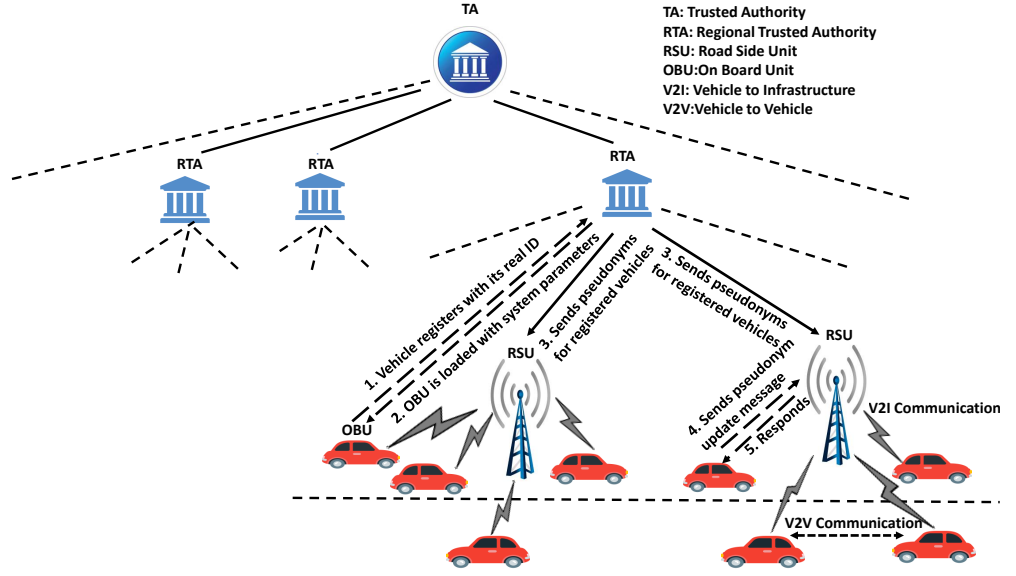


Figure 4.1: Proposed VANET architecture for CREASE

4.2 System Model

4.2.1 System Model

Prior to presenting the proposed CREASE scheme, we briefly introduce our system model and its components shown in Fig. 4.1. Our system model has two layers-1) the Upper layer and 2) the lower layer. The Trusted Authority (TA) and Regional Trusted Authorities (RTAs) constitutes the upper layer, and the lower layer consists of RSUs and OBUs. The TA is the root of the whole system, while each RTA acts as a lower-level local TA for its region. Table 4.1 presents the notations used in this section.

In CREASE, the TA generates its own public and private key pairs (PU_{TA}, PR_{TA}) . Each RTA is registered with the TA and gets its public and private key pairs (PU_{RTA}, PR_{RTA}) from the TA. The RTA acts as a local TA and is responsible for generating public and private key pairs for each vehicle and RSU registered with it. Each RTA maintains a Merkle Hash Tree (MHT) [72] of all RSUs registered with it.

Each vehicle is registered with its home RTA to participate in VANET. When a vehicle registers with its home RTA, its On Board Unit (OBU) is loaded with its public-private key pairs (PU_V, PR_V) , a set of pseudonyms $\{PID_1, \dots, PID_n\}$, and an initial pseudonym $PID_{initial}$ signed by its home RTA $(E(H(PID_{initial}) || t_{exp}), PR_{RTA})$ during registration, where $PID_{initial} \in \{PID_1, \dots, PID_n\}$ and t_{exp} is the expiration time of $PID_{initial}$. The OBU also stores MHT root generation timestamp $(T_{mhtRoot})$ generated and assigned by the RTA as well as the public key of the TA (PU_{TA}) . To preserve its privacy, a vehicle never uses its real identity in its communications. There are various approaches for generating and assigning pseudonyms

for vehicles [85, 39, 113, 112]. However, we do not address this issue in this research.

4.2.2 Basic Idea

In CREASE, each vehicle registers with its home RTA with its real ID to take part in VANET. Each RTA generates and assigns pseudonyms for all vehicles registered with it. The vehicle's OBU is loaded with its public-private key pair, the set of pseudonyms, and an initial pseudonym signed by its home RTA during registration. RTAs also send the registered vehicle's pseudonyms to all RSUs in its region using a secure protocol such as Transport Layer Security (TLS). RSUs in our scheme maintain MMPT combined with an MHT to facilitate the efficient management and authentication of pseudonyms of vehicles (how this is done is explained later). When a vehicle enters an area covered by an RSU for the first time after registration, it uses the credentials in the RSU's beacon message to authenticate the public key of the RSU. Upon successful authentication of the public key of the RSU, the vehicle sends its public key and initial pseudonym signed by its home RTA to authenticate itself. After mutual authentication, RSU sets a new pseudonym expiration time for the initial pseudonym and generates a symmetric key for encrypting and decrypting messages between RSU and the vehicle. The RSU also generates a group key

Table 4.1: Notation and Description of abbreviations used in this Chapter 4

Notation	Description
TA	Trusted Authority
RTA	Regional TA
RSU	Road Side Unit
OBU	On Board Unit
E	Encryption algorithm
PU_{TA}, PR_{TA}	Public and Private Keys of the TA
PU_{RTA}, PR_{RTA}	Public and Private Keys of RTA
PU_{RSU}, PR_{RSU}	Public and Private Keys of RSU
PU_V, PR_V	Public and Private Keys of vehicle V
ID_{RSU}	ID of RSU
$PID_{V_{initial}}$	Initial pseudonym assigned to Vehicle V
$PID_{V_{curr}}$	Pseudonym currently used by Vehicle V
$PID_{V_{new}}$	New Pseudonym activated for Vehicle V
t_{exp}	Initial Pseudonym Expiration Time of V
t_{new}	Current Pseudonym Expiration Time of V
t'_{new}	Newly Activated Pseudonym Expiration Time of V
t_s	Message generation timestamp
$signbyTA$	Signature of TA
$signbyRTA$	Signature of RTA
H	SHA-256 hash function
MHT	Merkle Hash Tree
MHV_s	Missing Hash Values of MHT for corresponding RSU

to be used by all vehicles within its region for vehicle to vehicle (V2V) communication. Next, the RSU sends the new pseudonym expiration time, the symmetric key, and the group key to the vehicle using DSRC standard protocol. RSUs assist vehicles in their region to change their pseudonym by providing a pseudonym expiration time associated with the pseudonym. CREASE allows vehicles to reuse their pseudonym without interrupting communication. When the validity time of a vehicle's current pseudonym is about to expire, the vehicle will again communicate with its RSU to activate a new pseudonym from a pool of pseudonyms received from its home RTA during initial registration. This pseudonym changing strategy speeds up the pseudonym changing frequency to avoid traceability of the vehicle. These steps involved in the authenticated communication process is depicted in Fig. 4.1.

4.2.3 Assumptions

We make the following assumptions in this work:

1. Each vehicle knows PU_{TA} and PU_{RTA} of the RTA under which it is registered. These are loaded into the vehicle's On Board Unit (OBU) during its initial registration with an RTA.
2. RTA generates and assigns a pool of pseudonyms for each vehicle registered under it.
3. Each RTA constructs an MHT of the public keys of RSUs that are registered with the RTA (described below). It then distributes the corresponding MHVs (described below) to each RSU in its region.
4. Each RSU maintains the MHT accompanied by MMPT to manage the pseudonyms of vehicles efficiently. RSU also distributes the symmetric key S_k and group key G_k securely to the vehicles in its region for communication.
5. RSUs registered under the same RTA know the public keys of each other.
6. Vehicle's OBU is tamper resistant and has enough storage capacity to store a large set of pseudonyms. This is not a serious restriction considering the current hardware capabilities.
7. The clocks of all entities (TA, RTAs, RSUs, and Vehicles (OBUs)) are loosely synchronized. This synchronization can be done using GPS.

4.2.4 Construction of Modified Merkle Patricia Trie for Efficient Management of Vehicle's Pseudonyms

Next we explain how an RSU uses an MMPT to store the pseudonyms (in hexadecimal representation) of a vehicle along with their status. Table 4.2 contains a sample list of four pseudonyms of a vehicle and their current status. Fig. 4.2 shows the MMPT

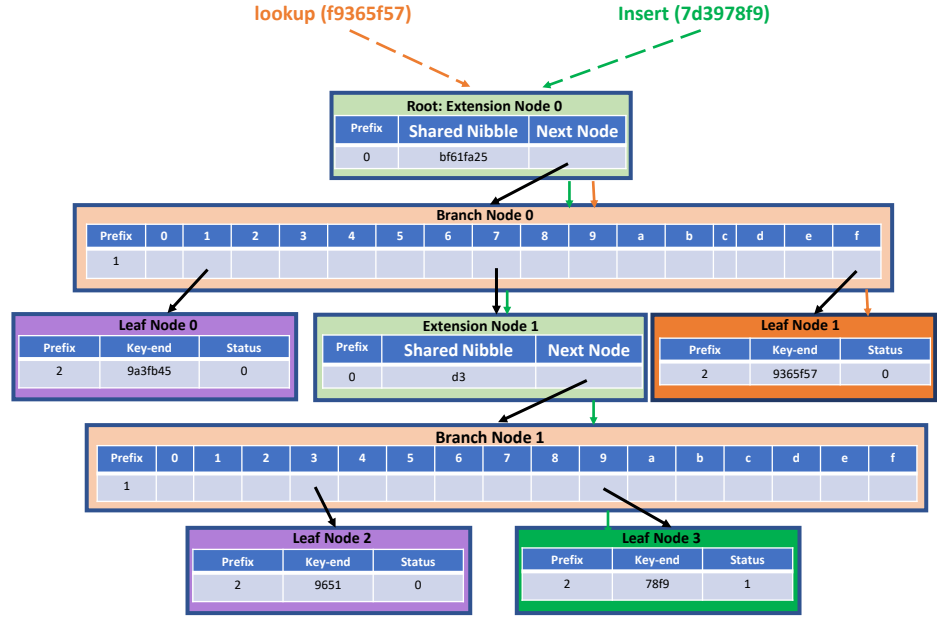


Figure 4.2: Modified Merkle Patricia Trie for storing the contents of Table 4.2

storing these four pseudonyms along with their status. In the MMPT, the root node is an extension node that contains the shared nibble *bf61fa25*, the public key of the vehicle which is concatenated with its pseudonyms. The root node's "next node" field points to the node right after it, which in our case is a branch node (Branch Node 0). If we look at the second pseudonym in Table 4.2 after concatenation, we can find *f* after *bf61fa25*. With this *f*, we can proceed into the next level, the leaf node (Leaf Node 1 in Fig. 4.2), which stores both the remaining value of the key and its current status. Therefore, we must start the search at the root node to lookup a key in MMPT and then proceed to the subsequent nodes based on the shared nibbles and remaining nibbles in the key. Finally, we can find the pseudonym and its status at a leaf node.

The insertion operation creates an entry for a pseudonym of a vehicle and the current status of the pseudonym in MMPT. We should first start from the root node to insert a key-value pair. Next, determine the current node's prefix value and its

Table 4.2: Pseudonyms and Current Status

Pseudonym	Status
19a3fb45	0
f9365f57	0
7d339651	0
7d3978f9	1

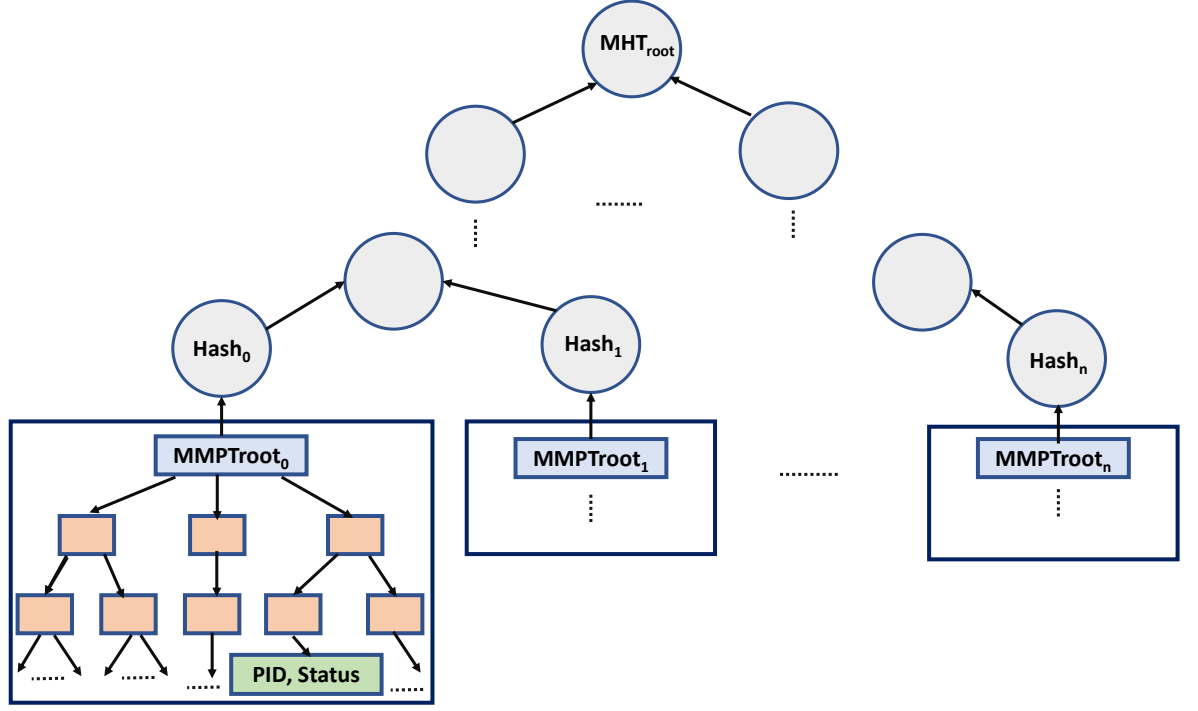


Figure 4.3: MMPT combined with MHT for storing pseudonyms of vehicles

nibbles. If the current node has a prefix of 1, then check whether the slot following the next nibble points to NULL. If this is the case, generate a new leaf node or a new extension node based on the residual nibbles left in the key to be inserted. Otherwise, navigate to the next node. If the current node's prefix is 0, find the shared nibbles and remaining nibbles left in the key. Next, a new leaf node, or a new branch node, or a new extension node is generated based on the remaining nibbles left in the key after sharing. For example, in Fig. 4.2, we first start from the root node to insert key "7d3978f9" after concatenating it with the public key *bf61fa25* of the vehicle. Next, we check the prefix of the current node. In our case, the root node's prefix is 0, and it is Extension Node 0. After that, we traverse to the Branch Node 0, pointed by the root node's next node field. Since the slot corresponding to the next nibble in the Branch Node 0 is not NULL and the remaining nibbles left in the key are greater than 1. Therefore, we travel down to Extension Node 1, where the partial path has diverged at Branch Node 1. We find that the slot corresponding to the Branch Node 1 is NULL. Next, we generate a new leaf node (Leaf Node 3) into this branch and set the status to 1 to indicate that this is the current pseudonym used by the vehicle with public key *bf61fa25*. The branch nodes, extension nodes, and leaf nodes used in MMPT shortens the length of the unique path to leaf nodes and makes it more efficient for inserting, retrieving, and removing pseudonyms. MMPT has worst case complexity of $O(n)$ for lookup, insert, and delete, where n is the length of the pseudonym (in hexadecimal representation).

MMPT is combined with the conventional blockchain to store the certificates

Algorithm 6: Distribution of Pseudonyms by *RTA*

- 1 **When a vehicle V registers with its home RTA**
 - 2 V 's *OBU* is loaded with its (PU_V, PR_V) , a set of pseudonyms $\{PID_1, \dots, PID_n\}$, and a $PID_{V_{initial}}$ signed by the RTA $(E(H(PID_{V_{initial}})||t_{exp}), PR_{RTA})$, where $PID_{V_{initial}} \in \{PID_1, \dots, PID_n\}$ and t_{exp} is the expiration time of $PID_{V_{initial}}$;
 - 3 RTA sends $\{PID_1, \dots, PID_n\}$, $PID_{initial}$, and PU_V to all $RSUs$ in its region using a secure protocol such as TLS;
-

of vehicles for authentication in BPPA [69]. In CREASE, each RSU maintains an MHT combined with MMPT as shown in Fig. 4.3 to store and manage vehicles' pseudonyms for efficient privacy-preserving authentication. Each RSU also maintains a database containing the public key of vehicle PU_V , set of pseudonyms of the vehicle $\{PID_1, \dots, PID_n\}$ associated with the PU_V , and corresponding MHVs. RSUs use the MHVs of MHT to verify the presence/absence of the pseudonym in the MMPT. Each MMPT stores the pseudonyms of a vehicle along with their latest status (active/inactive). The lookup operation in MMPT makes it efficient for an RSU to check and update the status of a specific pseudonym of a vehicle.

4.3 Detailed Description of CREASE

Distribution of Pseudonyms to Vehicles by their home RTA

In CREASE, RTAs generate pseudonyms to be used by all vehicles registered with it. Initially, each vehicle V registers with its real ID with its home RTA . Each vehicle's *OBU* is loaded with its public-private key pair (PU_V, PR_V) , a set of pseudonyms $\{PID_1, \dots, PID_n\}$, and one of the pseudonym is designated as an initial pseudonym $PID_{V_{initial}}$ after registration. $PID_{V_{initial}}$ along with its expiration time t_{exp} is signed by its home RTA $(E(H(PID_{initial})||t_{exp}), PR_{RTA})$, where $PID_{initial} \in \{PID_1, \dots, PID_n\}$ and t_{exp} is the expiration time of $PID_{V_{initial}}$. V uses this $PID_{V_{initial}}$ to communicate with an RSU that it encounters for the first time after the registration. RTA also sends $\{PID_1, \dots, PID_n\}$, $PID_{V_{initial}}$, and PU_V of V to all RSUs within its region using a secure protocol such as TLS. Upon receiving this, RSUs concatenate the PU_V with the pseudonyms and inserts them along with their status into their MMPT. Initially, the status of all pseudonyms is set to 0 (inactive) except for $PID_{V_{initial}}$. Thereby, all the RSUs under an RTA get the pseudonyms of all vehicles registered under the RTA . Detailed description of the pseudonym distribution process is presented in **Algorithm 6**.

When a vehicle V enters an area covered by an RSU after registration

V listens to the beacon message of the RSU which includes its ID ID_{RSU} , public key PU_{RSU} , PU_{RTA} signed by TA $(PU_{RTA}(\text{signbyTA}))$, root value of the MHT signed by RTA $(root_{\text{signbyRTA}})$ (where $root_{\text{signbyRTA}}$ contains (root of MHT || RSA signa-

Algorithm 7: When a Vehicle V enters an area covered by an RSU

```
1  Upon receiving the beacon message  $(ID_{RSU}, PU_{RTA \text{ signby } TA}, MHVs,$   
    $root_{\text{signby } RTA}, t_s)$  from the  $RSU$ ,  $V$  authenticates the  $RSU$  in the  
   following way  
2  if  $t_s$  is valid then  
3      Verifies the signatures of  $TA$  and  $RTA$ ;  
4      Retrieves root value and root generation timestamp of MHT from  
        $root_{\text{signby } RTA}$ ;  
5      if root generation timestamp is valid then  
6          Calculates root value using  $MHVs$  and  $PU_{RSU}$ ;  
7          if ( calculated root value == received root value ) then  
8              RSU's public key is authenticated;  
9          else  
10             RSU's public key is not valid;  
11         end if  
12     end if  
13 end if  
14 After authenticating the  $RSU$ ,  $V$  sends  $(E(PID_{initial},$   
    $(E(H(PID_{V_{initial}})||t_{exp}), PR_{RTA}), PU_V, t_s), PU_{RSU})$  to the  $RSU$ ;  
15 When the  $RSU$  receives the above message from  $V$   
16 Decrypts the message using its private key  $PR_{RSU}$ ;  
17 if  $t_s$  is valid then  
18     Verifies the signature of  $RTA$ ;  
19     Retrieves  $t_{exp}$ ;  
20     if  $t_{exp}$  is valid then  
21         Retrieves  $H(PID_{V_{initial}})$  and checks it against the calculated hash of  
            $PID_{V_{initial}}$ ;  
22         if the hash values match then  
23             if  $PID_{V_{initial}} \in MMPT$  then  
24                 Sets the status of the  $PID_{V_{initial}}$  to 1;  
25                 Sets new expiration time  $t_{new}$  for  $PID_{V_{initial}}$ ;  
26                 Sends  $(E(ID_{RSU}, (E(H(PID_{V_{initial}})||t_{new}), PR_{RSU}), S_k, G_k,$   
                    $t_s), PU_V)$  to  $V$ , where  $G_k$  is the group key and  $S_k$  is the  
                   symmetric key between  $V$  and  $RSU$ ;  
27             else  
28                  $PID_{V_{initial}}$  is not valid;  
29             end if  
30         else  
31             Authentication fails;  
32         end if  
33     end if  
34 end if
```

ture $\parallel T_{mhtRoot}$)), MHVs corresponding to the public key (PU_{RSU}) of the RSU, and timestamp t_s . V first checks the freshness of the beacon message from RSU using t_s . Next, V verifies the signature of the TA and RTA. A vehicle can move across different regions covered by several RTAs. We do not require vehicles to store all of the public keys for all the RTAs. CREASE requires V only to store the PU_{TA} to get the PU_{RTA} , which then can be used to verify the public key of the RSU. After verifying PU_{TA} and PU_{RTA} , V compares $T_{mhtRoot}$ in beacon message with the stored value of $T_{mhtRoot}$. If the received value of $T_{mhtRoot}$ is greater than or equal to the stored value of $T_{mhtRoot}$, then the vehicle recalculates the root of MHT using the MHVs and hash value of the public key PU_{RSU} of the sender RSU received in the beacon message. Next, V compares the received MHT root value with the calculated root value of MHT. If the two values are equal, PU_{RSU} is considered authentic. The detailed description of the authentication of RSU using MHT is presented in first part of **Algorithm 2**. After authenticating the RSU, V sends the following message to the RSU:

$V \rightarrow RSU: (E(PID_{initial}, (E(H(PID_{V_{initial}}) \parallel t_{exp}), PR_{RTA}), PU_V, t_s), PU_{RSU})$

RSU, upon receiving the above message, uses received t_s to verify the freshness of the above message. After that, RSU verifies the signature of the RTA. Next, it retrieves the pseudonym expiration time t_{exp} . If t_{exp} is valid, then it calculates the hash of the received initial pseudonym $PID_{V_{initial}}$ and compares this calculated hash value with received hash value $H(PID_{V_{initial}})$. If these two hash values are equal, then RSU concatenates public key of the vehicle PU_V with the $PID_{V_{initial}}$ and sets the status of the $PID_{V_{initial}}$ to 1 in the MMPT. RSU also sets new expiration time t_{new} for $PID_{V_{initial}}$ and signs it. RSU generates a symmetric key S_k for encrypting and exchanging messages between the V and RSU. After that, it sends the following message to the V :

$RSU \rightarrow V: (E(ID_{RSU}, (E(H(PID_{V_{initial}}) \parallel t_{new}), PR_{RSU}), S_k, G_k, t_s), PU_V)$

where, G_k is the group key to be used by all vehicles authenticated by the same RSU. A detailed description of the above discussion is presented in the second part of the **Algorithm 7**. After obtaining the group key G_k , vehicles under an RSU use G_k to securely communicate with each other. A vehicle includes its current pseudonym $PID_{V_{curr}}$ signed by the RSU ($E(H(PID_{V_{curr}}) \parallel t_{exp}), PR_{RSU}$) to its transmitted message m for vehicle to vehicle (V2V) authentication. It also appends message generation timestamp t_s to prevent a replay attack. Whenever a vehicle needs to transmit a message m to another vehicle, it encrypts m as follows:

$V(Sender) \rightarrow V(Receiver): (E(PID_{V_{curr}}, (E(H(PID_{V_{curr}}) \parallel t_{new}), PR_{RSU}), m, t_s), G_k)$

The receiver vehicle can verify the authenticity of the received message by checking the current pseudonym expiration time t_{new} and checking the received $H(PID_{V_{curr}})$ against its calculated hash of received $PID_{V_{curr}}$ in the above message. If both verifications are successful, the received message is considered authentic and valid. Otherwise, the receiver vehicle ignores the message.

4.3.1 Updating the Status of Pseudonym of a vehicle by RSU

Each Vehicle is supposed to change its pseudonym frequently to ensure privacy. The US-based SAE J2735 standard [95] recommends changing pseudonym every 120 sec-

Algorithm 8: Updating the Status of Pseudonyms by RSU

```
1 When the validity time of  $V$ 's current pseudonym  $PID_{V_{curr}}$  expires
2  $V$  selects a new pseudonym  $PID_{V_{new}}$  from the set of pseudonyms allocated to
   it;
3 Marks  $PID_{V_{curr}}$  as inactive;
4 Sends  $(E(PID_{V_{curr}}, PID_{V_{new}}, PU_V, t_s), S_k)$  to the  $RSU$ ;
5 When the  $RSU$  receives the above update message
6 Decrypts the message using secret key  $S_k$ ;
7 Checks the freshness of the received message using  $t_s$ ;
8 if  $t_s$  is valid then
9   | Lookup into MMPT for  $PID_{V_{curr}}$  and  $PID_{V_{new}}$ ;
10  | if  $PID_{V_{curr}} \in MMPT$  and  $PID_{V_{new}} \in MMPT$  then
11  |   | Sets the status of these pseudonyms to 0 and 1 respectively;
12  |   | Sets expiration time  $t'_{new}$  for  $PID_{V_{new}}$ ;
13  |   | Sends  $(E(E(H(PID_{V_{new}})||t'_{new}), PU_{RSU}), t_s), S_k)$  to  $V$ ;
14  | else
15  |   | Does not update the pseudonyms and ignores the message;
16  | end if
17 else
18  | Drops the received message;
19 end if
```

onds or after 1 km distance traveled (whichever comes last), while the European standard ETSI TS 102 867 [36] recommends changing pseudonym every five minutes. While a vehicle is parked, it is probably not necessary to change pseudonym that frequently. So, a vehicle needs 720 pseudonyms in 24 hours and 262,800 pseudonyms in 1 year according to the US-based SAE J2735 standard. In CREASE, we assume the vehicle's OBU is loaded with sufficient number of pseudonyms so that it will not need to reuse a pseudonym within a year. The size of each pseudonym is 16 bytes. Therefore, a vehicle requires approximately 4 MB of storage for storing its pseudonyms. We assume that the vehicles have enough storage capability to store its pseudonyms considering the current hardware capabilities. RSUs in our scheme assist the vehicles in its region to change pseudonyms by attaching an expiration time for each pseudonym. The expiration time indicates when a vehicle needs to change its current pseudonym. Once this expiration time elapses, the vehicle will again communicate with its RSU to activate a new pseudonym from the pool of pseudonyms received from its home RTA during initial registration. RSU determines the time when all the vehicles in its region need to perform the pseudonym change. **Thus, vehicles within the same RSU's region change their pseudonym simultaneously, resulting in reducing the chance of linkability between the new pseudonym and the old pseudonym.** We assume that the clocks of TA, RTAs, RSUs and Vehicles(OBUs) are loosely synchronized. When the validity time of a vehicle's current pseudonym $PID_{V_{curr}}$ is about to expire, it selects a new pseudonym

$PID_{V_{new}}$ from its stored set of pseudonyms and informs the RSU securely about its new pseudonym. After receiving this pseudonym update message, the RSU concatenates the public key PU_V of the vehicle with $PID_{V_{curr}}$ and $PID_{V_{new}}$ and looks up into its MMPT. RSU sets the status of the pseudonyms $PID_{V_{curr}}$ and $PID_{V_{new}}$ to 0 and 1 in the MMPT respectively after successful verification. The RSU also sets expiration time t'_{new} for $PID_{V_{new}}$ and sends it to V . The detailed description of updating the status of a vehicle's pseudonym is presented in **Algorithm 9**.

When a vehicle V moves from one RSU's region to another RSU's region

When a vehicle V moves from the region covered by one RSU RSU_i to the region covered by another RSU RSU_j , V first verifies the authenticity of the RSU_j as described in **Algorithm 7**. Next, V sends following message to RSU_j to authenticate itself for communication:

$$V \rightarrow RSU_j: (E(PID_{V_{curr}}, (E(H(PID_{V_{curr}}) || t_{new}), PR_{RSU_i}), PU_V, t_s), PU_{RSU_j})$$

The following two cases arise.

Case 1: RSU_j is registered with the V 's home RTA: RSU_j , upon receiving the above message, uses received t_s to check the freshness of the message and verifies the signature of the RSU_i . If the verification is successful, then the RSU_j verifies the authenticity of the current pseudonym $PID_{V_{curr}}$ of the V as described in **Algorithm 7**.

Case 2: RSU_j is not in the region covered by V 's home RTA: The RSU_j first checks the freshness of the received message using t_s and forwards the received message to its home RTA. Next, the RSU_j 's home RTA communicates with the V 's home RTA and gets the set of pseudonyms allocated to the V , public key of the vehicle PU_V , and public key of RSU_i PU_{RSU_i} . RSU_j 's home RTA sends the required credentials to the all RSUs in its region using a secure protocol such as TLS.

After obtaining the public key of RSU_i , RSU_j verifies the received hash value from V as described in **Algorithm 7**. Next, RSU_j inserts the set of pseudonyms of V concatenating with its PU_V along with their status in its MMPT.

After authenticating V , RSU_j sets the new expiration time t'_{new} for $PID_{V_{curr}}$ and sends the following message to the V :

$$RSU_j \rightarrow V: (E(ID_{RSU_j}, (E(H(PID_{V_{curr}}) || t'_{new}), PR_{RSU_j}), S'_k, G'_k, t_s), PU_V)$$

where, G'_k is the group key shared by all authenticated vehicles in the region of the RSU_j and S'_k is the shared symmetric key between the V and RSU_j . The detailed description of the above discussion is presented in **Algorithm 9**.

4.4 Performance Evaluation

In this section, we first analyze the security of CREASE. Then, we present a formal proof of correctness of CREASE using BAN logic [19]. Next, a verification of the security of our scheme is presented using SPAN [50] and AVISPA [7] tools. Finally, we compare our protocol with LIAP [111] and ASPA [4] protocols with respect to security features and protocol overhead.

Algorithm 9: When a Vehicle V moves from RSU_i to RSU_j

```
1 After verifying the authenticity of the  $RSU_j$  as described in  
   Algorithm 2,  $V$  sends  $(E(PID_{V_{curr}}), (E(H(PID_{V_{curr}}) || t_{new}), PR_{RSU_i}),$   
    $PU_V, t_s), PU_{RSU_j})$  to  $RSU_j$   
2 When the  $RSU_j$  receives the above message from  $V$ 

---

3 Decrypts the message using its private key  $PR_{RSU_j}$ ;  
4 Case 1:  $RSU_j$  is registered with  $V$ 's home  $RTA$   
5 if  $t_s$  is valid then  
6   Verifies the signature of  $RSU_i$  and retrieves  $t_{new}$ ;  
7   if  $t_{new}$  is valid then  
8     Retrieves  $H(PID_{V_{curr}})$  and checks against received  $PID_{V_{curr}}$  ;  
9     if the hash values match then  
10      if  $PID_{V_{curr}} \in MMPT$  then  
11        Sets the status of the  $PID_{V_{curr}}$  to 1;  
12        Sets new expiration time  $t'_{new}$  for  $PID_{V_{curr}}$ ;  
13      else  
14        Ignores the update message;  
15      end if  
16    else  
17       $PID_{V_{curr}}$  is not valid;  
18    end if  
19  end if  
20 end if  
21 Case 2:  $RSU_j$  is not registered with  $V$ 's home  $RTA$   
22 if  $t_s$  is valid then  
23   Gets the set of pseudonyms allocated to the  $V$ ,  $PU_V$ , and  $PU_{RSU_i}$  from  
   its home  $RTA$  through a secure protocol such as TLS;  
24    $RSU_j$  verifies the signature of  $RSU_i$  and retrieves  $t_{new}$ ;  
25   if  $t_{new}$  is valid then  
26     Retrieves  $H(PID_{V_{curr}})$  and checks against received  $PID_{V_{curr}}$  ;  
27     if the hash values match then  
28       Inserts the pseudonyms of the  $V$  into its MMPT;  
29       Sets the status of all pseudonyms to 0 except  $PID_{V_{curr}}$  which is  
       set to 1;  
30       Sets expiration time  $t'_{new}$  for  $PID_{V_{curr}}$ ;  
31     else  
32        $PID_{V_{curr}}$  is not valid;  
33     end if  
34   end if  
35 end if
```

4.4.1 Security Analysis

In this subsection, we discuss the security features of CREASE.

Mutual Authentication

Under CREASE, a vehicle V and RSU authenticate each other before communicating with each other as follows. After obtaining the Missing Hash Values (MHVs), MHT root signed by RTA $root_{signbyRTA}$, public key of RTA signed by TA $PU_{RTA signbyTA}$, and public key of the RSU PU_{RSU} , the vehicle recalculates the root value of the MHT and compares it with the received $root_{signbyRTA}$. If the two values are equal PU_{RSU} is considered authentic. Next, the vehicle sends a message containing its initial pseudonym $PID_{V_{initial}}$ and initial pseudonym signed by the RTA along with pseudonym expiration time $E((H(PID_{V_{initial}})||t_{exp}), PR_{RTA})$. Upon receiving the message, RSU calculates the hash of received $PID_{V_{initial}}$ and compares this calculated hash value with received hash value $H(PID_{V_{initial}})$. If these two hash values are equal, then RSU lookup into the MMPT and set the status of the $PID_{V_{initial}}$ to 1. After mutual authentication, RSU sends a symmetric key S_k and a group key G_k to the vehicle. The group key G_k is used by all vehicles authenticated by the same RSU for secure communication.

Vehicle Anonymity

A sender uses only its pseudonym in all communication, which ensures that the receivers cannot obtain the sender's real identity. Receivers authenticate the sender based on temporary credentials. Only trusted authority should know the real identities and ensure conditional anonymity to achieve accountability. In CREASE, vehicles use a pseudonym and pseudonym expiration time to send messages. Receivers authenticate the sender vehicle based on the pseudonym and its expiration time. The real identity of a vehicle is never used in communication. The RTA only know the real identity of the vehicle. The pseudonym of a vehicle is resolvable to its real identity only by its RTA.

Unlinkability

Unlinkability requires that an adversary cannot link messages sent with two different pseudonyms by the same vehicle. In CREASE, a vehicle uses its initial pseudonym $PID_{V_{initial}}$ for mutual authentication with an RSU after registration. Next, the RSU sets a new expiration time t_{new} for $PID_{V_{initial}}$. Once this expiration time is about to expire, the vehicle communicates with the RSU using the symmetric key S_k established between the vehicle and RSU during the mutual authentication process to activate a new pseudonym from the pool of pseudonyms allocated to it. **RSU assists vehicles in its region to change their pseudonym simultaneously and frequently by associating an expiration time for a registered pseudonym. Therefore, our scheme reduces the chance of linkability between the same vehicle's two pseudonyms.**

Non-repudiation

All messages sent by a vehicle contain its current pseudonym and the hash of its current pseudonym $PID_{V_{curr}}$ along with the pseudonym expiration time signed by its RSU $E((H(PID_{V_{curr}})||t_{new}), PR_{RSU})$ in CREASE. The receiver firstly verifies the signature of the RSU. Next, it checks the pseudonym expiration time. If it is valid then it computes the hash of the received pseudonym and check it against the received $H(PID_{V_{curr}})$. If two hash values are same, then the sender is considered authentic. Since the vehicle uses pseudonym from its stored set of pseudonym and registers it with an RSU for communication. Therefore, the vehicle cannot deny the messages sent by it. Besides, it is not possible for an attacker to forge the signature of the RSU.

Resistance to Replay Attack

The message generation timestamp t_s is encrypted along with all messages in CREASE to resist the replay attack. The TA, RTAs, RSUs, and Vehicles' clocks are assumed to be loosely synchronized (this can be achieved using GPS) in our scheme. Due to this addition of t_s , each entity can detect whether the message is fresh enough to prevent a replay attack.

Resistance to Sybil Attack

A Sybil attack occurs when a malicious vehicle uses multiple pseudonyms in parallel to impersonate a number of vehicles. Therefore, the number of pseudonyms and their validity period that a vehicle can use should be limited. In CREASE, each vehicle's OBU is loaded with a set of pseudonyms, and the RTA signs one initial pseudonym along with the pseudonym expiration time $E((H(PID_{V_{initial}})||t_{exp}), PR_{RTA})$. When a vehicle enters an RSU's region after registration with an RTA, it uses this initial pseudonym for authentication. The RSU sets a new expiration time t_{new} for $PID_{V_{initial}}$ and sends it to the vehicle. When the t_{new} is about to expire, the vehicle communicates with the RSU to activate a new pseudonym from its pool of pseudonyms. Then the RSU activates and signs a new pseudonym for the vehicle along with the expiration time for that pseudonym $E((H(PID_{V_{new}})||t'_{new}), PR_{RSU})$. Therefore, only one pseudonym of a vehicle is valid at a time in CREASE and hence it resists the Sybil attack.

Resistance to message injection Attack

Under CREASE, when a vehicle enters an RSU's region after registration with an RTA, it first verifies the signatures of TA $PU_{RTA\text{signbyTA}}$ and RTA $root_{\text{signbyRTA}}$. Then, it computes the MHT root value using MHVs and compares it with received root value of MHT signed by the RTA. RSU is considered as legitimate if the two values are equal, otherwise a message injection attack is detected. After authenticating the RSU, the vehicle sends its initial pseudonym $PID_{V_{initial}}$ and the $PID_{V_{initial}}$ signed by the RTA along with the expiration time t_{exp} . The receiver RSU verifies the

Table 4.3: BAN logic notation

Notation	Description
$P \equiv X$	P believes X
$P\triangleleft X$	P sees X
$P \sim X$	P once said X
$P\Rightarrow X$	P controls X
$P\overset{S_k}{\longleftrightarrow} Q$	Only P and Q know the shared secret key S_k
$\#(X)$	X is fresh
$\{X\}_k$	X is encrypted with the key k
$\wp\kappa(P, K_P)$	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}

signature of the RTA and then it checks the t_{exp} . If t_{exp} is valid then it computes the hash of the received pseudonym and checks it against the received $H(PID_{V_{initial}})$. If the two values are not equal, then a message injection attack is detected. It is not possible for an attacker to forge the signature of TA or RTA.

4.4.2 Formal proof of correctness of CREASE based on BAN logic

Borrows, Abadi, and Needham (BAN) logic [19] is a popular authentication protocols analysis model to formally verify the correctness of authentication protocols [67, 24, 33, 62, 61, 75]. In this subsection, we analyze CREASE using BAN logic and the PKI-based extended BAN logic [96] and demonstrate its correctness. First, we present a brief overview of the BAN logic and the inference rules for BAN logic in this subsection. Next, we discuss a formal idealization of the proposed CREASE scheme's messages, the list of initial assumptions, goals of our protocol, and logical derivation to achieve the goals.

BAN Logic Notation

The list of BAN logic notations used in this section are presented in Table 4.3.

Protocol Idealization

For the formal analysis, the messages exchanged between RSU and Vehicle to achieve mutual authentication is simplified and formally idealized as follows:

When a vehicle V enters an area covered by RSU_i after registration following messages are exchanged for mutual authentication:

- M1: RSU_i broadcasts the message: $\langle ID_{RSU_i}, \sigma(K_{RTA}, K_{TA}^{-1}), MHVs, \sigma(MHTroot, K_{RTA}^{-1}), K_{RSU_i}, t_s \rangle$

- M2: $V \rightarrow RSU_i$: $\langle \{PID_{V_{initial}}, \{H(PID_{V_{initial}}) \parallel t_{exp}\}_{K_{RTA}^{-1}}, K_V, t_s\}_{K_{RSU_i}} \rangle$
- M3: $RSU_i \rightarrow V$: $\langle \{ID_{RSU_i}, \{H(PID_{V_{initial}}) \parallel t_{new}\}_{K_{RSU_i}^{-1}}, RSU_i \xleftrightarrow{S_k} V, G_k, t_s\}_{K_v} \rangle$

When a vehicle moves from RSU_i to RSU_j following messages are exchanged for mutual authentication:

- M4: RSU_j broadcasts the message: $\langle ID_{RSU_j}, \sigma(K_{RTA}, K_{TA}^{-1}), MHVs, \sigma(MHTroot, K_{RTA}^{-1}), K_{RSU_j}, t_{s_{RSU_j}} \rangle$
- M5: $V \rightarrow RSU_j$: $\langle \{PID_{V_{curr}}, \{H(PID_{V_{curr}}) \parallel t_{new}\}_{K_{RSU_j}^{-1}}, K_V, t_s\}_{K_{RSU_j}} \rangle$
- M6: $RSU_j \rightarrow V$: $\langle \{ID_{RSU_j}, \{H(PID_{V_{curr}}) \parallel t'_{new}\}_{K_{RSU_j}^{-1}}, RSU_j \xleftrightarrow{S'_k} V, G'_k, t_s\}_{K_v} \rangle$

Initial Assumptions

In CREASE, the TA and RTA are trusted by both RSUs and vehicles. RTAs, RSUs, and vehicles know the public key of TA (K_{TA}). RTA is registered with the TA and gets its public and private key pairs (K_{RTA}, K_{RTA}^{-1}). Every RTA generates the public and private key pairs for each RSU (K_{RSU}, K_{RSU}^{-1}) and vehicle (K_V, K_V^{-1}) registered under it. Each vehicle's OBU is loaded with a set of pseudonyms and an initial pseudonym ($PID_{V_{initial}}$) signed by its home RTA. The initial assumptions of the protocol is summarized as follows:

- A1: $V \mid \equiv \wp\kappa(TA, K_{TA})$
- A2: $V \mid \equiv \Pi(K_{TA}^{-1})$
- A3: $RSU_i \mid \equiv \wp\kappa(RTA, K_{RTA})$
- A4: $RSU_i \mid \equiv \Pi(K_{RTA}^{-1})$
- A5: $RSU_j \mid \equiv \wp\kappa(RTA, K_{RTA})$
- A6: $RSU_j \mid \equiv \Pi(K_{RTA}^{-1})$
- A7: $V \mid \equiv \wp\kappa(RTA, K_{RTA})$
- A8: $V \mid \equiv \Pi(K_{RTA}^{-1})$
- A9: $RSU_j \mid \equiv \wp\kappa(RSU_i, K_{RSU_i})$
- A10: $RSU_j \mid \equiv \Pi(K_{RSU_i}^{-1})$

- A11: $V| \equiv \wp\kappa(V, K_V)$
- A12: $V| \equiv \Pi(K_V^{-1})$
- A13: $RSU_i| \equiv RTA \Rightarrow MHTroot$
- A14: $RSU_j| \equiv RTA \Rightarrow MHTroot$
- A15: $V| \equiv RTA \Rightarrow MHTroot$
- A16: $RSU_i| \equiv RSU_i \Rightarrow (RSU_i \xleftarrow{S_k} V, G_k)$
- A17: $V| \equiv RSU_i \Rightarrow (RSU_i \xleftarrow{S_k} V, G_k)$
- A18: $RSU_j| \equiv RSU_j \Rightarrow (RSU_j \xleftarrow{S'_k} V, G'_k)$
- A19: $V| \equiv RSU_j \Rightarrow (RSU_j \xleftarrow{S'_k} V, G'_k)$
- A20: $RSU_i| \equiv RTA \Rightarrow (H(PID_{V_{initial}}) || t_{exp})$
- A21: $RSU_j| \equiv RTA \Rightarrow (H(PID_{V_{initial}}) || t_{exp})$
- A22: $RSU_j| \equiv RSU_i \Rightarrow (H(PID_{V_{curr}}) || t_{new})$
- A23: $RSU_i| \equiv \#(t_s)$
- A24: $RSU_j| \equiv \#(t_s)$
- A25: $V| \equiv \#(t_s)$

Goal of CREASE

The goals of CREASE are:

- G1: $V| \equiv RTA| \equiv MHTroot$
- G2: $V| \equiv MHTroot$
- G3: $RSU_i| \equiv RTA| \equiv (H(PID_{V_{initial}}) || t_{exp})$
- G4: $RSU_i| \equiv (H(PID_{V_{initial}}) || t_{exp})$
- G5: $V| \equiv RSU_i| \equiv (RSU_i \xleftarrow{S_k} V, G_k)$
- G6: $V| \equiv (RSU_i \xleftarrow{S_k} V, G_k)$
- G7: $RSU_j| \equiv RSU_i| \equiv (H(PID_{V_{curr}}) || t_{new})$
- G8: $RSU_j| \equiv (H(PID_{V_{curr}}) || t_{new})$
- G9: $V| \equiv RSU_i| \equiv (RSU_i \xleftarrow{S'_k} V, G'_k)$
- G10: $V| \equiv (RSU_i \xleftarrow{S'_k} V, G'_k)$

Derivation of the above goals

On the basis of logical postulates in Appendix A and initial assumptions, we derive the above goals as follows.

From message M1, we deduce G1 and G2 as follows:

- D1. $V \triangleleft \langle ID_{RSU_i}, \sigma(K_{RTA}, K_{TA}^{-1}), MHVs, \sigma(MHTroot, K_{RTA}^{-1}), K_{RSU_i}, t_s \rangle$
- D2. $V \triangleleft \sigma(K_{RTA}, K_{TA}^{-1})$ (From D1 and $R5$ (8))
- D3. $V \models \# (\sigma(K_{RTA}, K_{TA}^{-1}))$ (From A25 and $R4$)
- D4. $V \models TA \mid \sim K_{RTA}$ (From A1, A2, and $R1$ (1))
- D5. $V \models \# \sigma(MHTroot, K_{RTA}^{-1})$ (From $R4$ and A21)
- D6. $V \models RTA \mid \sim MHTroot$ (From D11, A9, A10, and $R1$ (1))
- D7. $V \models RTA \mid \equiv MHTroot$ (From D5, D6, and $R2$) (G1)
- D8. $V \models MHTroot$ (From A15, D7, and $R3$) (G2)

From message M2, we deduce G3 and G4 as follows:

- D9. $RSU_i \triangleleft \langle \{PID_{V_{initial}}, \{H(PID_{V_{initial}}) \parallel t_{exp}\}_{K_{RTA}^{-1}}, K_V, t_s\}_{K_{RSU}} \rangle$
- D10. $RSU_i \triangleleft \langle \{H(PID_{V_{initial}}) \parallel t_{exp}\}_{K_{RTA}^{-1}} \rangle$ (From D9 and $R5$ (8))
- D11. $RSU_i \models \# (H(PID_{V_{initial}}) \parallel t_{exp})$ (From A23 and $R4$)
- D12. $RSU_i \models RTA \mid \sim (H(PID_{V_{initial}}) \parallel t_{exp})$ (From A3, A4, and $R1$ (1))
- D13. $RSU_i \models RTA \mid \equiv (H(PID_{V_{initial}}) \parallel t_{exp})$ (From D11, D12, and $R2$) (G3)
- D14. $RSU_i \models (H(PID_{V_{initial}}) \parallel t_{exp})$ (From A18, D13, and $R3$) (G4)

From message M3, we deduce G5 and G6 as follows:

- D15. $V \triangleleft \langle \{ID_{RSU_i}, \{H(PID_{V_{initial}}) \parallel t_{new}\}_{K_{RSU_i}^{-1}}, RSU_i \xrightarrow{S_k} V, G_k, t_s\}_{K_v} \rangle$
- D16. $V \models \# (RSU_i \xrightarrow{S_k} V, G_k)$ (From A25 and $R4$)
- D17. $V \models RSU_i \mid \sim (RSU_i \xrightarrow{S_k} V, G_k)$ (From A11, A12, D15, and $R1$ (3))
- D18. $V \models RSU_i \mid \equiv (RSU_i \xrightarrow{S_k} V, G_k)$ (From D16, D17, and $R2$ (G5)
- D19. $V \models (RSU_i \xrightarrow{S_k} V, G_k)$ (From A17, D18, and $R3$) (G6)

From message M4, we deduce G1 and G2 as follows:

- D20. $V \triangleleft \langle ID_{RSU_j}, \sigma(K_{RTA}, K_{TA}^{-1}), MHVs, \sigma(MHTroot, K_{RTA}^{-1}), K_{RSU_j}, t_s \rangle$

- D21. $V \triangleleft \sigma(K_{RTA}, K_{TA}^{-1})$ (From D20 and $R5$ (8))
- D22. $V | \equiv \# (\sigma(K_{RTA}, K_{TA}^{-1}))$ (From A25 and $R4$)
- D23. $V | \equiv TA | \sim K_{RTA}$ (From A1, A2, and $R1$ (1))
- D24. $V | \equiv \# \sigma(MHTroot, K_{RTA}^{-1})$ (From $R4$ and A25)
- D25. $V | \equiv RTA | \sim MHTroot$ (From D20, A9, A10, and $R1$ (1))
- D26. $V | \equiv RTA | \equiv MHTroot$ (From D24, D25, and $R2$) (G1)
- D27. $V | \equiv MHTroot$ (From A15, D26, and $R3$) (G2)

From message M5, we deduce G7 and G8 as follows:

- D28. $RSU_j \triangleleft \langle \{PID_{V_{curr}}, \{H(PID_{V_{curr}}) || t_{new}\}_{K_{RSU_j}^{-1}}, K_V, t_V\}_{K_{RSU_j}^{-1}} \rangle$
- D29. $RSU_j \triangleleft \langle \{H(PID_{V_{curr}}) || t_{new}\}_{K_{RSU_i}^{-1}} \rangle$ (From D28 and $R5$ (8))
- D30. $RSU_i | \equiv \# (H(PID_{V_{curr}}) || t_{new})$ (From A24 and $R4$)
- D31. $RSU_j | \equiv RSU_i | \sim (H(PID_{V_{curr}}) || t_{new})$ (From A5, A6, and $R1$ (1))
- D32. $RSU_j | \equiv RSU_i | \equiv (H(PID_{V_{curr}}) || t_{new})$ (From D30, D31, and $R2$)
..... (G7)
- D33. $RSU_j | \equiv (H(PID_{V_{curr}}) || t_{new})$ (From A21, D32, and $R3$) (G8)

From message M6, we deduce G9 and G10 as follows:

- D34. $V \triangleleft \langle \{ID_{RSU_j}, \{H(PID_{V_{curr}}) || t'_{new}\}_{K_{RSU_j}^{-1}}, RSU_j \xleftarrow{S'_k} V, G'_k, t_{RSU_j}\}_{K_v} \rangle$
- D35. $V | \equiv \# (RSU_j \xleftarrow{S'_k} V, G'_k)$ (From A25 and $R4$)
- D36. $V | \equiv RSU_j | \sim (RSU_j \xleftarrow{S'_k} V, G'_k)$ (From A11, A12, D34, and $R1$ (3))
- D37. $V | \equiv RSU_i | \equiv (RSU_j \xleftarrow{S'_k} V, G'_k)$ (From D35, D36, and $R2$) (G9)
- D38. $V | \equiv (RSU_j \xleftarrow{S'_k} V, G'_k)$ (From A19, D37, and $R3$) (G10)

The above BAN logic analysis shows that our protocol achieves all the goals (G1-G10) and vehicles can get the correct symmetric key and group key after a mutual authentication process for secure communication.

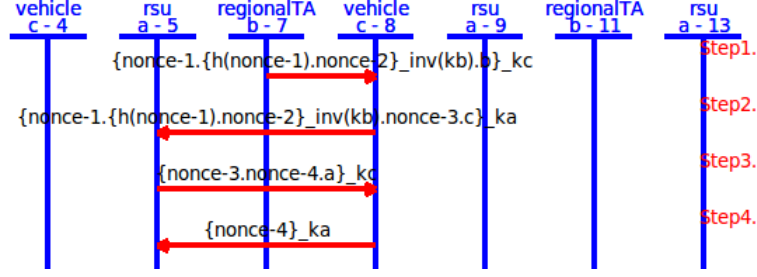


Figure 4.4: Message sequence chart of our scheme generated by SPAN and AVISPA tool

4.4.3 Automated verification of CREASE based on SPAN and AVISPA tools

SPAN (Security Protocol ANimator)[50] and AVISPA (Automated Validation of Internet Security Protocols and Applications)[7] tools are widely used in literature [2, 6, 78, 10] for analysing the security of the protocols. In this research, we also verify the security of our proposed CREASE scheme against replay attack, man-in-the-middle attack, and impersonation attack using SPAN and AVISPA tools.

In our model, we consider three basic roles which we call *rsu*, *regionalTA*, and *vehicle*, and are denoted by *a*, *b*, and *c* respectively. Here, *ka*, *kb*, and *kc* represent the public keys of the *rsu*, *regionalTA*, and *vehicle* respectively, and *h* represents the cryptographic hash function. In the proposed protocol, *regionalTA* first activates the start signal and sends a message to *vehicle* containing hash of initial pseudonym *nonce-1* and the expiration time *nonce-2* of *nonce-1* which are encrypted with its private key *inv(kb)*. The encrypted hash value and the initial pseudonym expiration time are used by *rsu* to authenticate the initial pseudonym *nonce-1* of the *vehicle*.

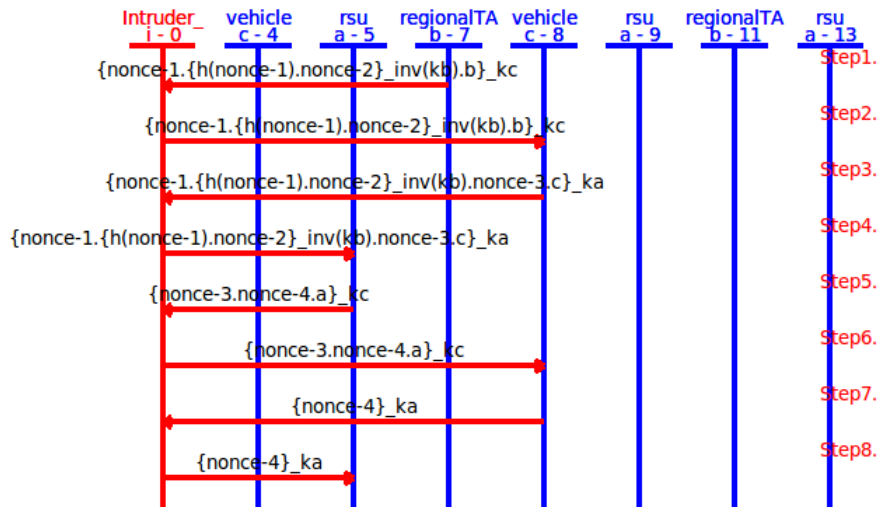


Figure 4.5: Message sequence chart of our scheme in the presence of intruder generated by SPAN and AVISPA tool

Fig. 2.8 shows the message sequence chart of the proposed scheme using SPAN and AVISPA tools.

The message sequence chart in the presence of an intruder is presented in Fig. 4.5. This sequence chart demonstrates that the intruder is unable to read and/or modify the messages. The intruder is only able to listen and forward the messages. We describe the sequence of exchanged messages in presence of an intruder i as follows:

- Step 1: The *regionalTA* initiates session and sends a message containing the *vehicle*'s initial pseudonym $nonce-1$ and the expiration time $nonce-2$ of the $nonce-1$ (where, $h(nonce-1)$ and $nonce-1$ is encrypted with its private key $inv(kb)$) 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 message 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: Upon receiving the message, *rsu* retrieves the hash value and initial pseudonym expiration time using public key kb of the *regionalTA*. Firstly, *rsu* verifies the validity of initial pseudonym expiration time. Next, it recalculates the hash of initial pseudonym received in the message to verify the initial pseudonym of the *vehicle*. After that it sends back the received nonce along with another generated nonce to the *vehicle*.
- Step 6: The intruder only listens the message and passes it to the *vehicle*.
- Step 7: The *vehicle* decrypts the message using its private key $inv(ka)$ and retrieves the nonces. Next, the *vehicle* 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*.

4.4.4 Comparison with other related protocols

In this section, we compare the security features and the protocol overheads of CREASE with those of the NERA [9], LIAP [111] and ASPA [4] schemes.

Security features comparison

In CREASE, RSUs provide pseudonym expiration time t_{exp} for each authentic vehicle within its transmission range. Each vehicle uses its current pseudonym $PID_{V_{curr}}$ signed by the RSU $E((H(PID_{V_{curr}})|| t_{exp}), PR_{RSU})$ for communication. It is not possible for an attacker to forge the signature of the RSU. Besides, the message generation timestamp is encrypted along with the message in our scheme to resist the

replay attack. We assume that the clocks of the TA, RTAs, RSUs and vehicles are loosely synchronized (this can be done using GPS). In LIAP [111], an RSU manages and assigns a local master key to every vehicle in its region after the mutual authentication. A vehicle uses this master key to generate pseudonyms for VANET communication. In ASPA [4], vehicles get multiple short-time pseudonym certificates from the Pseudonym Provider (PP) and they are all valid at the same time interval. However, both LIAP and ASPA are not secure against Sybil attacks as multiple pseudonyms of a vehicle are valid at the same interval. We present the comparison of the security features of the CREASE with LIAP and ASPA in Table 4.4. The results show that CREASE is more secure than the other two protocols.

Protocol Overhead Analysis

In CREASE, a vehicle first verifies the signature of the TA and RTA. Then, the vehicle calculates the MHT root value using the Missing Hash Values (MHVs) received in the beacon message of the RSU for authentication. On an Intel Core 2 1.83GHz processor machine running Windows Vista in 32-bit mode, RSA 2048 signature verification takes 0.16 ms, and SHA-256 hash computation takes 111 MiB/s using Crypto++ 5.6.0 [31]. With much lower computation costs for the hash function calculation, the total cost of RSU authentication in CREASE depends mainly on RSA signature verification. On the other hand, a vehicle uses a linear search to check the RSU certificate revocation list (RCRL) for authentication in the LIAP scheme [111]. NERA scheme [9] uses the bilinear pairing and Map-To-Point operations, which cause overhead in RSU authentication. Since the authentication of RSUs by vehicles is not taken into consideration in ASPA protocol [4]. Therefore, Fig. 4.6 presents the comparison of RSU authentication overhead involved in CREASE with that of LIAP and NERA. There is a significant increase in RSU authentication cost when the number of revoked RSUs increases in LIAP. On the contrary, the RSU authentication cost is low under NERA and very low under CREASE. Fig. 4.6 shows that the authentication overhead under LIAP is almost three times as much as CREASE when the number of revoked RSUs reaches 30.

In CREASE, RSU first decrypts the message from a vehicle using its private key PR_{RSU} . Then it verifies the signature of the RTA. Then, it concatenates the public key of the vehicle with the vehicle's initial pseudonym and looks up into the MMPT.

Table 4.4: Comparison of Security Features

	CREASE	NERA	ASPA	LIAP
Mutual Authentication	Yes	Yes	No	Yes
Vehicle Anonymity	Yes	Yes	Yes	Yes
Uninkability	Yes	Yes	Yes	Yes
Non-repudiation	Yes	Yes	Yes	Yes
Resistance to Replay Attack	Yes	No	Yes	Yes
Resistance to Sybil Attack	Yes	No	No	No
Resistance to Message Injection Attack	Yes	Yes	Yes	Yes

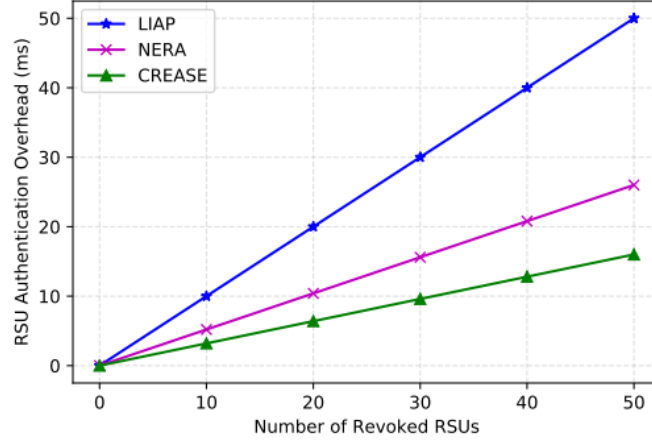


Figure 4.6: RSU Authentication Overhead Comparison

In both LIAP [111] and NERA [9], RSU checks the Vehicle Certificate Revocation List (VCRL) after decrypting the message from vehicle. Next, it verifies the signature of the Certificate Authority (CA) to authenticate the vehicle. The signature verification overhead in LIAP and NERA protocol using bilinear pairing is $T_{mul} + T_{mtp} + 3T_{par}$. Here, T_{mul} denotes the time of performing one point multiplication ($T_{mul} = 0.39$ ms), T_{mtp} denotes the time for performing a MapToPoint hash operation ($T_{mtp} = 0.09$ ms), and T_{par} denotes the time for performing a pairing operation ($T_{par} = 3.21$ ms) [111]. In ASPA protocol, a vehicle uses its initial pseudonym provided by the Vehicular Manufacturing Company (VMC) to request a longterm certificate (LTC) from the Certificate Authority (CA). CA checks the CRL to issue a LTC for the vehicle. Next, the vehicle gets a Pseudonym Certificate (PC) from the LTC Authority using the LTC. The vehicle sends a message to the Pseudonym Provider (PP) directly or through RSU for pseudonyms using the PC. After verifying the PC of the vehicle, Pseudonym Provider sends multiple pseudonyms to the vehicle. The signature verification cost in this scheme using Digital Signature Algorithm (DSA) is 0.37 ms [4]. A comparison of the signature verification overhead of CREASE with LIAP [111], NERA [9], and ASPA [4] is presented in Fig. 4.7. It is observed that CREASE has significantly lower signature verification overhead compared to LIAP, NERA, and ASPA. For example, when the number of vehicles reaches 30, the overall signature verification cost is approximately 92 ms for both LIAP [111] and NERA [9] and 22 ms for ASPA [4], whereas it is only 4.8 ms for CREASE.

In CREASE, vehicle and RSU authenticates each other without using certificate and certificate revocation lists (CRLs). Vehicles use Missing Hash Values (MHVs) and Public key of the RSU PU_{RSU} received in the RSU's beacon message to recalculate the MHT root value. Next, the vehicle checks this hash value against the received root value of MHT signed by the RTA $root_{signbyRTA}$ for authentication. After

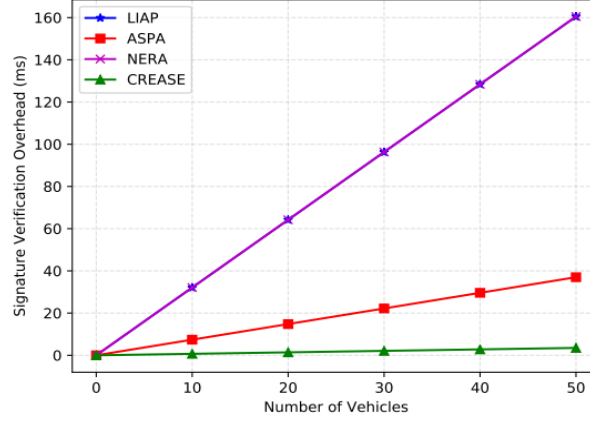


Figure 4.7: Signature Verification Overhead Comparison

authenticating the RSU, the vehicle sends its initial pseudonym signed by the RTA along with the pseudonym expiration time $E((H(PID_{V_{initial}})||t_{exp}), PR_{RTA})$, initial pseudonym $PID_{V_{initial}}$, and its public key PU_V for authentication. RSU first checks if t_{exp} is valid. After that, it computes the hash of the $PID_{V_{initial}}$ in the received message and compares it with the received $H(PID_{V_{initial}})$. If the hash values matches then $PID_{V_{initial}}$ is considered as valid. On the contrary, vehicles use long-term certificates for authentication in both LIAP [111] and ASPA [4] schemes. Next, the RSUs or Pseudonym Providers (PP) check the latest Certificate Revocation List (CRLs) of vehicles to verify the vehicles' authenticity. In NERA scheme [9] the TA revokes the malicious vehicle and adds its real ID to the CRL. In these schemes, the CA (Certificate Authority) or TA maintains a certificate revocation list (CRL) and distributes the updated CRL of vehicles to all entities in VANET. A CRL typically consists of a header, the current date, the last time it has been updated, the next time it will be updated, and a complete list of revoked certificates signed by the CA [73]. Moreover, the size of CRL increases significantly as the number of entities grows. Therefore, the use of CRLs for authentication incurs significant computation and communication overhead. Besides, the CA needs to send the CRLs very often to keep the communication updated and secure. So, the LIAP, ASPA, and NERA schemes introduce significant additional communication overhead on RSUs or Pseudonym Providers (PPs).

4.5 Summary

We presented a novel and efficient privacy-preserving authentication scheme that leverages both Merkle Hash Tree and Modified Merkle Patricia Trie; our scheme allows vehicles to get a set of pseudonyms once and pick one random pseudonym from this set at a time and use it to preserve privacy. MHT and MMPT data structures

help in managing and storing these pseudonyms efficiently for verifying the authenticity of the vehicles while at the same time preserving their privacy. MHT of public keys of RSUs also help in authenticating RSUs efficiently without certificates. Our scheme does not require the RSUs and vehicles store the certificates and CRL for authentication. Our scheme is robust against replay attacks, man-in-the-middle attacks, and impersonation attacks. We compared the security properties and protocol overhead of our scheme with those of other related protocols and also presented a formal proof of correctness.

Chapter 5 Conclusion and Future Works

In the last decade, there has been growing interest in Vehicular Ad Hoc Networks (VANETs). Today, car manufacturers have already started to equip vehicles with sophisticated sensors that can provide many assistive features such as front collision avoidance, automatic lane tracking, partial autonomous driving, suggestive lane changing, and so on. Such technological advancements are enabling the adoption of VANETs not only to provide safer and more comfortable driving experience but also to provide many other useful services to the driver as well as passengers of a vehicle. However, privacy, authentication, and secure message dissemination are some of the main issues that need to be thoroughly addressed and solved for the widespread adoption/deployment of VANETs. In this chapter, we summarize the results of our dissertation and also discuss future works.

5.1 Summary of the Dissertation

First, we proposed a distributed and scalable privacy-preserving authentication scheme based on two-layer architecture. The upper layer consists of Trusted authority (TA) and Regional TAs (RTAs), while the lower layer consists of Road Side Units (RSUs) and vehicles. TA is the root of our system model, while each RTA acts as a lower-level local TA. Our scheme uses Merkle Hash Tree (MHT) root value combined with the latest MHT root generation timestamp (TmhtRoot) to reduce the storage and computational overhead associated with the traditional Certificate Revocation List (CRL) based authentication schemes for authenticating RSUs. In our scheme, a vehicle needs to store only the public key of the Trusted Authority (TA) and the latest Merkle Hash Tree (MHT) root generation timestamp to authenticate the public key of RSUs, thereby authenticating the RSUs themselves. We use Modified Merkle Patricia Trie (MMPT) to eliminate the space and computation-intensive CRL for efficient authentication of vehicles by RSUs. In many of the existing schemes, when a vehicle observes an event, this event is disseminated to every vehicle within its transmission range, which in turn is forwarded to other vehicles using an underlying routing protocol. It can cause message transmission overhead due to the dissemination of the same event by several vehicles. However, in our scheme, a vehicle sends messages to authenticated RSUs for further dissemination. Only RSUs are responsible for disseminating messages to vehicles within their transmission range and other relevant RSUs through their home RTAs. So, RSUs can aggregate messages and prevent the propagation of redundant messages to vehicles.

Second, we present a low overhead and efficient privacy-preserving pseudonym-based authentication scheme by leveraging cuckoo filters (CF). Authentication schemes based on pseudonyms that assign each vehicle multiple identities can improve privacy significantly. However, the overhead of maintaining such a large volume of identities is overwhelming for traditional Certificate Revocation List (CRL) based solutions. In our scheme, the Local Trusted Authority (LTA) assigns a set of pseudonyms to

all vehicles in its region. We use Merkle Hash Tree (MHT) to maintain the set of pseudonyms allocated to a vehicle. LTA inserts the MHT root value associated with the vehicle's set of pseudonyms in its region in a positive cuckoo filter. If a vehicle is found to be malicious, the LTA, instead of inserting all the unexpired pseudonyms, only inserts the MHT root value associated with the vehicle's set of pseudonyms in a negative cuckoo filter. LTA also maintains positive and negative cuckoo filters to manage legitimate RSUs and compromised RSUs. Both vehicles and RSUs only need to store and check the cuckoo filters from the LTA for mutual authentication. The use of cuckoo filters supports efficient insert, delete, and lookup operation, which significantly improves the authentication efficiency.

Third, we proposed a distributed and decentralized certificateless authentication scheme for efficient management of pseudonyms of vehicles. We combined MHT with Modified Merkle Patricia Trie to store vehicles' pseudonyms along with their corresponding 'current status' values. In our scheme, RSUs use missing hash values of MHT to verify the presence or absence of a specific pseudonym of a vehicle in the MMPT. The use of branch nodes, extension nodes, and leaf nodes provides an efficient way to insert, delete, and update a specific pseudonym and its corresponding status. In our scheme, RSUs assist vehicles in their region to change their pseudonyms simultaneously. This is achieved by assigning the same expiration time to the pseudonyms of all vehicles in an RSU's region. Expiration time associated with a pseudonym helps vehicles within the same RSU's region to simultaneously and frequently change their pseudonyms. Therefore, it reduces the chance of message linking attack as well as vehicle's traceability by an attacker.

5.2 Future Works

The following are some of the future works that we intend to do for moving this research forward:

- Although our scheme provides an efficient way to authenticate and disseminate messages to relevant vehicles, it still depends on RSUs for communication. However, with the increasing number of RSUs, the overhead will grow. Therefore, we need to improve our authentication and message dissemination scheme for large-scale applications.
- Considering the potential benefit of cuckoo filter, we plan to use the cuckoo filters for an efficient vehicle to vehicle (V2V) authentication to enhance the security of V2V communication.
- We will also explore quantum-resistant cryptography for VANET system as recent advancements in quantum computing have made most traditional cryptographic algorithms vulnerable.

Appendix

Appendix A: Inference Rules for BAN Logic

BAN logic postulates are used to derive the goal from the protocol idealization and the initial assumption. The inference rules are listed next:

- *R1:Message meaning rule*

$$\frac{P| \equiv \wp k(Q, K_Q), P| \equiv \Pi(K_Q^{-1}), P \triangleleft \sigma(X, K_Q^{-1})}{P| \equiv Q| \sim X} \quad (1)$$

Here P and Q are communication entities. If P believes that Q has public key K_Q and private key K_Q^{-1} , and Q sees X encrypted with K_Q^{-1} , then P believes that Q generated X.

$$\frac{P| \equiv \wp k(Q, K_Q), P| \equiv \Pi(K_Q^{-1}), P \triangleleft \sigma(\Re(X, P), K_Q^{-1})}{P| \equiv Q| \sim X} \quad (2)$$

Similarly, if P believes that Q has public key K_Q and private key K_Q^{-1} , and Q sees X encrypted with K_Q^{-1} for which P is the intended recipient, then P believes that Q once said X.

$$\frac{P| \equiv Q| \sim (Cert(T_{exp}, st)), P| \equiv Q| \equiv T_{exp}}{P| \equiv Q| \equiv st} \quad (3)$$

In the above rule, Q acts as the issuer of a certificate. If P believes that Q once said about a certificate statement st which is still valid for a duration of time T_{exp} , then P believes in the statement st for the duration T_{exp} .

$$\frac{P| \equiv P \xleftarrow{S_k} Q, P \triangleleft \{X\} S_k}{P| \equiv Q| \sim X} \quad (4)$$

If P believes that Q shared a key S_k with it and P sees X encrypted with S_k , then P believes Q once said X.

- *R2:Nonce Verification rule*

$$\frac{P| \equiv \#(X), P| \equiv Q| \sim X}{P| \equiv Q| \equiv X} \quad (5)$$

The above rule concerns with the freshness of the message with respect to the time. If P believes that X could have been uttered only recently and Q once said X, then P believes Q believes in the freshness of X.

- *R3:Jurisdiction rule*

$$\frac{P| \equiv Q \Rightarrow X, P| \equiv Q| \equiv X}{P| \equiv X} \quad (6)$$

If P believes that Q has jurisdiction over X and P believes Q believes X, then P believes X.

- *R4:Freshness rule*

$$\frac{P| \equiv \#(X)}{P| \equiv \#(X, Y)} \quad (7)$$

This rule states that if one of the components of a message is fresh, then the entire message is fresh.

- *R5:Sees rule*

$$\frac{P \triangleleft (X, Y)}{P \triangleleft (X)} \quad (8)$$

$$\frac{P \triangleleft \sigma(\Re(X, all), K_Q^{-1})}{P \triangleleft \sigma(\Re(X, P), K_Q^{-1})} \quad (9)$$

$$\frac{P| \equiv \wp\kappa(Q, K_Q), P| \equiv \Pi(K_Q^{-1}), P \triangleleft \{X\}_{K_Q^{-1}}}{P \triangleleft X} \quad (10)$$

$$\frac{P| \equiv \wp\kappa(Q, K_Q), P| \equiv \Pi(K_Q^{-1}), P \triangleleft \sigma(X, K_Q^{-1})}{P \triangleleft X} \quad (11)$$

$$\frac{P| \equiv \wp\kappa(P, K_P), P| \equiv \Pi(K_P^{-1}), P \triangleleft \{X\}_{K_P^{-1}}}{P \triangleleft X} \quad (12)$$

$$\frac{P| \equiv P \xleftarrow{S_k} Q, P \triangleleft \{X\}_{S_k}}{P \triangleleft X} \quad (13)$$

This rule states that if P sees a message, then it can see the components of the message as P knows the necessary key.

Bibliography

- [1] I. 1609.2. *Trial-use standard for wireless access in vehicular environments-security services for applications and management messages*. IEEE Standards, 2006.
- [2] S. Adhikari, S. Ray, G. P. Biswas, and M. S. Obaidat. Efficient and secure business model for content centric network using elliptic curve cryptography. *International Journal of Communication Systems*, 32(1):e3839, 2019.
- [3] S. Al-Sultan, M. M. Al-Dooriand, A. H. Al-Bayatti, and H. Zedan. A comprehensive survey on vehicular ad hoc network. *Journal of network and computer applications*, 37:380–392, January 2014.
- [4] Q. E. Ali, N. Ahmad, A. H. Malik, W. U. Rehman, A. U. Din, and G. Ali. ASPA: Advanced Strong Pseudonym based Authentication in Intelligent Transport System. *PloS one*, 14(8):e0221213, 2019.
- [5] H. Alshaer. Securing vehicular ad-hoc networks connectivity with roadside units support. In *Proceedings of IEEE 8th GCC Conference and Exhibition (GCCCE 2015), March 2015*. IEEE, 2015.
- [6] R. Amin, S. H. Islam, M. S. Obaidat, G. Biswas, and K.-F. Hsiao. An anonymous and robust multi-server authentication protocol using multiple registration servers. *International Journal of Communication Systems*, 30(18):e3457, 2017.
- [7] Avispa. AVISPA. <http://www.avispa-project.org>, 2002.
- [8] P. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *Proceedings of 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, December 4-8, 2005 (ASIACRYPT, 2005), Lecture Notes in Computer Science book series (LNCS, volume 3788)*, pages 515–532. Springer, 2005.
- [9] M. Bayat, M. Pournaghi, M. Rahimi, and M. Barmshoory. NERA: A new and efficient RSU based authentication scheme for VANETs. *Wireless networks*, 26(5):3083–3098, 2020.
- [10] L. Benarous, B. Kadri, S. Bitam, and A. Mellouk. Privacy-preserving authentication scheme for on-road on-demand refilling of pseudonym in VANET. *International Journal of Communication Systems*, page e4087, 2019.
- [11] S. Biswas and J. Mišić. Deploying proxy signature in VANETs. In *Proceedings of the IEEE Global Telecommunications Conference GLOBECOM 2010 (GLOBECOM 2010)*. IEEE, 2010.

- [12] S. Biswas and J. Mišić. Establishing trust on VANET safety messages. In *Proceedings of the Second International Conference on Ad Hoc Networks (AD-HOCNETS 2010)*, Victoria, BC, Canada: Springer, 2010., 2010.
- [13] S. Biswas and J. Mišić. Proxy signature-based RSU message broadcasting in VANETs. In *Proceedings of the 25th Biennial Symposium on Communications (QBSC)*, 2010. Kingston, ON, Canada: IEEE, 2010, pages 5–9. IEEE, 2010.
- [14] S. Biswas, J. Mišić, and V. Mišić. ID-based safety message authentication for security and trust in vehicular networks. In *Proceedings of 31st ICDCS Workshops*, pages 323–331, Minneapolis, MN, 2011. IEEE.
- [15] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970.
- [16] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Advances in Cryptology—CRYPTO, Lecture Notes in Computer Science, vol. 3152*, Berlin, Germany, 2004. Springer-Verlag.
- [17] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In Kilian J. (eds.) *Advances in Cryptology — CRYPTO 2001. Lecture Notes in Computer Science, vol 2139*. Springer, pages 213–229, Berlin, Heidelberg, 2001. Springer.
- [18] A. Boualouache, S. M. Senouci, and S. Moussaoui. Towards an efficient pseudonym management and changing scheme for vehicular Ad-Hoc networks. In *Proceedings of 2016 IEEE Global Communications Conference*. IEEE, 2016.
- [19] M. Burrows, M. Abadi, and R. M. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, February 1990.
- [20] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Liou. Efficient and robust pseudonymous authentication in VANET. In *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, pages 19–28. ACM, 2007.
- [21] J. Camenisch, S. Hohenberger, and M. Pedersen. Batch verification of short signatures. In *Advances in Cryptology-Eurocrypt, vol. 4515, Lecture Notes in Computer Science*, pages 246–263, Berlin, Germany, 2007. Springer-Verlag.
- [22] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast security: a taxonomy and some efficient construction. In *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, (INFOCOM’99)*, volume 2, pages 708–716. IEEE, IEEE, March 1999.
- [23] G. Caronni, M. Waldvogel, D. Sun, and B. Plattner. The versakey framework: versatile group key management. *IEEE Journal on selected areas in communications*, 17(9):1614–1631, September 1999.

- [24] K. Chain, K.-H. Chang, W.-C. Kuo, and J.-F. Yang. Enhancement authentication protocol using zero-knowledge proofs and chaotic maps. *International Journal of Communication Systems*, 30(1):e2945, 2017.
- [25] D. Chaum and E. van Heyst. Group signatures. In *Proceedings of Eurocrypt, vol. 547, Lecture Notes in Computer Science, New York, 1991*, pages 257–265, 1991.
- [26] J. Cheon and J. Yi. Fast batch verification of multiple signatures. In *Public-Key Cryptography-PKC, vol. 4450, Lecture Notes in Computer Science*, pages 442–457, Berlin, Germany, 2007. Springer-Verlag.
- [27] T. W. Chim, S. M. Yiu, L. Hui, and V. Li. SPECS: Secure and privacy enhancing communications schemes for VANETs. *Ad Hoc Networks*, 9(2):189–203, Mar. 2011.
- [28] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong. An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1621–1632, 2019.
- [29] J. Cui, J. Zhang, H. Zhong, and Y. Xu. SPACF: A secure privacy-preserving authentication scheme for VANET with cuckoo filter. *IEEE Transactions on Vehicular Technology*, 66(11):10283–10295, November 2017.
- [30] R. L. R. D. Chaum and A. T. Sherman, editors. *Blind Signatures for Untraceable Payments*, Advances in Cryptology, Boston, MA, 1983. Springer.
- [31] W. Dai. Crypto++ library 5.6.0. <http://www.cryptopp.com/benchmarks.html>, 2009.
- [32] S. Dietzel, E. Schoch, F. Kargl, B. Könings, and M. Weber. Resilient secure aggregation for vehicular networks. *IEEE Network*, 24(1):26–31, Jan.-Feb. 2010.
- [33] S. Dolev, L. Krzywiecki, N. Panwar, and M. Segal. Dynamic attribute based vehicle authentication. *Wireless Networks*, 23(4):1045–1062, 2017.
- [34] X. Dong, L. Wei, H. Zhu, Z. Cao, and L. Wang. EP2DF: An efficient privacy-preserving data-forwarding scheme for service-oriented vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 60(2):580–591, Feb. 2011.
- [35] K. Emara, W. Woerndl, and J. Schlichter. On evaluation of location privacy preserving schemes for VANET safety applications. *Computer Communications*, 63:11–23, Jun. 2015.
- [36] T. ETSI. 102 941 v1. 1.1—Intelligent Transport Systems (ITS); security; trust and privacy management. *Standard, TC C-ITS*, 2012.

- [37] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher. Cuckoo filter: Practically better than bloom. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 75–88, 2014.
- [38] A. Ferrara, M. Green, S. Hohenberger, and M. Pedersen. Practical short signature batch verification. In *Proceedings of CT-RSA, vol. 5473, Lecture Notes in Computer Science*, pages 309–324, Berlin, Germany, 2009. Springer-Verlag.
- [39] D. Förster, H. Löhr, A. Grätz, J. Petit, and F. Kargl. An evaluation of pseudonym changes for vehicular networks in large-scale, realistic traffic scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 19(10):3400–3405, 2017.
- [40] X. L. H. Zhu, R. Lu, P. Ho, and X. Shen. AEMA: An aggregated emergency message authentication scheme for enhancing the security of vehicular ad hoc networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, pages 1436–1440. IEEE, 2008.
- [41] J. J. Haas, Y.-C. Hu, and K. P. Laberteaux. Efficient certificate revocation list organization and distribution. *IEEE Journal on Selected Areas in Communications*, 29(3):595–604, 2011.
- [42] M. Hamid, M. S. Islam, and C. S. Hong. Developing security solutions for wireless mesh enterprise networks. In *Proceedings of IEEE Wireless Communications and Networking Conference(WCNC)*, pages 2549–2554. IEEE, 2008.
- [43] M. A. Hamid, M. Abdullah-Al-Wadud, C. S. Hong, O. Chae, and S. Lee. A robust security scheme for wireless mesh enterprise networks. *Annals of Telecommunications*, 64(5-6):401–413, June 2009.
- [44] Y. Hao, Y. Chen, C. Zhou, and S. Wei. A distributed key management framework with cooperative message authentication in VANETs. *IEEE Journal on Selected Areas in Communications*, 29(3):616–629, March 2011.
- [45] Y. Hao, Y. Cheng, and K. Ren. Distributed key management with protection against RSU compromise in group signature based VANETs. In *Proceedings of IEEE GLOBECOM, New Orleans, LA, USA, 2008*. IEEE, 2008.
- [46] Y. Hao, C. Chengcheng, C. Zhou, and W. Song. A distributed key management framework with cooperative message authentication in VANETs. *IEEE Journal on Selected Areas in Communications*, 29(3):616–629, March 2011.
- [47] H. Harney and C. Muckenhirn. Group key management protocol GKMP architecture. Technical Report RFC 2094, ietf.org, July 1997.
- [48] H.-C. Hsiao, A. Studer, C. Chen, A. Perrig, F. Bai, B. Bellur, and A. Iyer. Flooding-resilient broadcast authentication for VANETs. In *Proceedings of the*

- 17th Annual International Conference on Mobile Computing and Networking (MOBICOM 2011)*, pages 193–204. ACM, 2011.
- [49] IEEE. 1609.12-2016 - IEEE standard for wireless access in vehicular environments (WAVE) - identifier allocations, 2016.
 - [50] Irisa. SPAN. <http://people.irisa.fr/Thomas.Genet/span/>, 2006.
 - [51] S. Jiang, X. Zhu, and L. Wang. A conditional privacy scheme based on anonymized batch authentication in vehicular ad hoc networks. In *Proceedings of IEEE WCNC*, pages 2375–2380, Shanghai, China, Apr. 2013, 2013. IEEE.
 - [52] S. Jiang, X. Zhu, and L. Wang. An efficient anonymous batch authentication scheme based on HMAC for VANETs. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2193–2204, 2016.
 - [53] S. Jiang, X. Zhu, and L. Wang. An efficient anonymous batch authentication scheme based on HMAC for VANETs. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2193–2204, Aug. 2016.
 - [54] H. Krawczyk and T. Rabin. Chameleon signatures. In *Network and Distributed System Security Symposium (NDSS)*, February 2000.
 - [55] K. Laberteaux, J. Haas, and Y.-C. Hu. Security certificate revocation list distribution for VANET. In *Proceedings of the fifth ACM international workshop on vehicular inter-networking, 15 September 2008*, pages 88–89. ACM, 2008.
 - [56] C. Lai, R. Lu, and D. Zheng. Achieving secure and seamless IP Communications for group-oriented software defined vehicular networks. In *12th International Conference on Wireless Algorithms, Systems, and Applications*, pages 356–368. Springer, Springer, June 2017.
 - [57] C. Lai, K. Zhang, N. Cheng, H. Li, and X. Shen. SIRC: a secure incentive scheme for reliable cooperative downloading in highway VANETs. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1559–1574, June 2017.
 - [58] C. Lai, H. Zhou, N. Cheng, and X. S. Shen. Secure group communications in vehicular networks: A software-defined network-enabled architecture and solution. *IEEE Vehicular Technology Magazine*, 12(4):40–49, 2017.
 - [59] C.-T. Li, M.-S. Hwang, and Y. P. Chu. A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks. *Computer Communications*, 31(12):2803–2814, 2008.
 - [60] K. Li, W. F. Lau, and M. H. Au. A secure and efficient privacy-preserving authentication scheme for vehicular networks with batch verification using cuckoo filter. In *International Conference on Network and System Security*, pages 615–631. Springer, 2019.

- [61] X. Li, T. Liu, M. S. Obaidat, F. Wu, P. Vijayakumar, and N. Kumar. A lightweight privacy-preserving authentication protocol for VANETs. *IEEE Systems Journal*, 2020.
- [62] X. Li, Y. Liu, and X. Yin. An anonymous conditional privacy-preserving authentication scheme for VANETs. In *Proceedings of 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 1763–1770. IEEE, 2019.
- [63] X. Lin and X. Li. Achieving efficient cooperative message authentication in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 62(7):3339–3348, Sept. 2013.
- [64] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho, and X. Shen. TSVC: Timed efficient and secure vehicular communications with privacy preserving. *IEEE Transactions on Wireless Communications*, 7(12):4987–4998, December 2008.
- [65] D. Y. W. Liu, J. K. Liu, Y. Mu, W. Susilo, and D. Wong. Revocable ring signature. *Journal of Computer Science and Technology*, 22:78–794, November 2007.
- [66] X. Liu, H. Zhao, M. Pan, H. Yue, X. Li, and Y. Fang. Traffic-aware multiple mix zone placement for protecting location privacy. In *Proceedings of IEEE INFOCOM, Orlando, FL, USA, 2012*, pages 972–980. IEEE, 2012.
- [67] Y. Liu, W. Guo, Q. Zhong, and G. Yao. LVAP: Lightweight V2I authentication protocol using group communication in VANETs. *International Journal of Communication Systems*, 30(16):e3317, 2017.
- [68] R. Lu, X. Lin, and X. Shen. Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks. In *Proceedings of IEEE INFOCOM, San Diego, CA, Mar. 2010*, pages 1–9, 2010.
- [69] Z. Lu, Q. Wang, G. Qu, H. Zhang, and Z. Liu. A blockchain-based privacy-preserving authentication scheme for VANETs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(12):2792–2801, 2019.
- [70] Y.-C. L. M.-S. Hwang, C.-C. Lee. An untraceable blind signature scheme. *IEICE Transactions on Fundamentals on Electronics, Communications and Computer Sciences*, E86-A(7):1902–1906, 2003.
- [71] D. Manivannan, S. S. Moni, and S. Zeadally. Secure authentication and privacy-preserving techniques in Vehicular Ad-hoc NETWORKS (VANETs). *Vehicular Communications*, pages 330–348, 2020.
- [72] R. C. Merkle. Protocols for public key cryptosystems. In *Proceedings of 1980 IEEE Symposium on Security and Privacy*, pages 122–134. IEEE, 1980.

- [73] S. Micali. Certificate revocation system, Sept. 9 1997. US Patent 5,666,416.
- [74] S. S. Moni and D. Manivannan. An efficient RSU authentication scheme based on Merkle Hash Tree for VANETs. In *Proceedings of 2020 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2020.
- [75] S. S. Moni and D. Manivannan. A scalable and distributed architecture for secure and privacy-preserving authentication and message dissemination in VANETs. *Internet of Things*, 13:100350, 2020.
- [76] S. S. Moni and D. Manivannan. CREASE: Certificateless and REused-pseudonym based Authentication Scheme for Enabling Security and Privacy in VANETs. *Pervasive and Mobile Computing Journal (PMC)*, 2021 revised and submitted.
- [77] S. S. Moni and D. Manivannan. A lightweight privacy-preserving v2i mutual authentication scheme using cuckoo filter in vanets. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pages 815–820. IEEE, 2022.
- [78] J. Moon, Y. Lee, J. Kim, and D. Won. Improving an anonymous and provably secure authentication protocol for a mobile user. *Security and Communication Networks*, 2017.
- [79] B. Palanisamy and L. Liu. Mobimix: Protecting location privacy with mix-zones over road networks. In *Proceedings of IEEE 27th International Conference on Data Engineering, April 2011 Hanover, Germany*, pages 494–505. IEEE, 2011.
- [80] J. Pan, J. Cui, L. Wei, Y. Xu, and H. Zhong. Secure data sharing scheme for VANETs based on edge computing. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):1–11, 2019.
- [81] P. Papadimitratos, G. Mezzour, and J.-P. Hubaux. Certificate revocation list distribution in vehicular communication systems. In *Proceedings of The Fifth ACM International Workshop on Vehicular Inter-NETworking (VANET 2008), San Francisco, CA, USA, September 15, 2008*. ACM, 2008.
- [82] M.-H. Park, G.-P. Gwon, S.-W. Seo, and H.-Y. Jeong. RSU-based distributed key management (RDKM) for secure vehicular multicast communications. *IEEE Journal on Selected Areas in Communications*, 29(3):644–658, 2011.
- [83] V. Paruchuri and A. Duresi. PAAVE: Protocol for anonymous authentication in vehicular networks using smart cards. In *Proceedings of GLOBECOM*. IEEE, 2010.
- [84] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. The TESLA broadcast authentication protocol. *RSA CryptoBytes*, 5:Available: <https://users.ece.cmu.edu/adrian/projects/tesla-cryptobytes/paper/index.html>, Summer 2002.

- [85] J. Petit, F. Schaub, M. Feiri, and F. Kargl. Pseudonym schemes in vehicular networks: A survey. *IEEE Communications Surveys and Tutorials*, 17(1):228–255, 2015.
- [86] L. Ramabaja and A. Avdullahu. Compact merkle multiproofs. *arXiv preprint arXiv:2002.07648*, 2020.
- [87] M. Raya and J.-P. Hubaux. The security of vehicular ad hoc networks. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 11–21. ACM, 2005.
- [88] M. Raya and J. P. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1):39–68, 2007.
- [89] R. L. Rivest et al. RFC 1321: The MD5 message-digest algorithm, 1992.
- [90] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1985, Vol.196, pages 47–53. Springer-Verlag, New York, Inc., 1984.
- [91] A.-N. Shen, S. Guo, D. Zeng, and M. Guizani. A lightweight privacy preserving protocol using chameleon hashing for secure vehicular communications. In *Proceedings of IEEE Wireless Communications and Networking Conference*, pages 2543–2548. IEEE, 2012.
- [92] P.-Y. Shen, V. Liu, M. Tang, and C. William. An efficient public key management system: An application in vehicular ad hoc networks. In *Proceedings of Pacific Asia Conf. Inf. Syst. (PACIS 2011)*, 2011.
- [93] A. T. Sherman and D. A. McGrew. Key establishment in large dynamic groups using one-way function trees. *IEEE transactions on Software Engineering*, 29(5):444–458, May 2003.
- [94] K.-A. Shim. CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks. *IEEE Transactions on Vehicular Technology*, 61(4):1874–1883, May 2012.
- [95] S. V. Standard. Dedicated short range communications (DSRC) message set dictionary. *SAE International*, November, 2009.
- [96] Sufatrio and R. H. C. Yap. Extending BAN logic for reasoning with modern PKI-based protocols. In *Proceedings of 2008 IFIP International Conference on Network and Parallel Computing*, pages 190–197. IEEE, 2008.
- [97] X. Sun, X. Lin, and P.-H. Ho. Secure vehicular communications based on group signature and id-based signature scheme. In *Proceedings of IEEE International Conference on Communications, June 2007*, pages 1539–1545. IEEE, 2007.

- [98] Y. Sun, R. Lu, X. Lin, X. Shen, and J. Su. An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications. *IEEE Transactions on Vehicular Technology*, 59(7):3589–3603, 2010.
- [99] Y. Sun, W. Trappe, and K. J. R. Liu. An efficient key management scheme for secure wireless multicast. In *Proceedings of IEEE International Conference on Communications (ICC), 2002*, volume 2, pages 1236–1240. IEEE, IEEE, April 2002.
- [100] Y. Sun, W. Trappe, and K. J. R. Liu. Topology-aware key management schemes for wireless multicast. In *Global Telecommunications Conference, GLOBECOM '03*, pages 1471–1475. IEEE, December 2003.
- [101] Y. Sun, W. Trappe, and K. J. R. Liu. A scalable multicast key management scheme for heterogeneous wireless networks. *IEEE/ACM Transactions on Networking*, 12:653 – 666, August 2004.
- [102] L. Sweeney. K-ANONYMITY: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, 10(5):557–570, 2002.
- [103] H. Tan, M. Ma, H. Labiod, A. Boudguiga, J. Zhang, and P. H. J. Chong. A secure and authenticated key management protocol (SA-KMP) for vehicular networks. *IEEE Transactions on Vehicular Technology*, 65(12):9570–9584, 2016.
- [104] S. Tangade, S. S. Manvi, and P. Lorenz. Decentralized and scalable privacy-preserving authentication scheme in VANETs. *IEEE Transactions on Vehicular Technology*, 67(9):8647–8655, 2018.
- [105] F. A. Teixeira, V. F. e Silva, J. L. Leoni, D. F. Macedo, and J. M. S. Nogueira. Vehicular networks using the IEEE 802.11p standard: An experimental analysis. *Vehicular Communications*, 1(2):91–96, April 2014.
- [106] R. van der Heijden, S. Dietzel, and F. Kargl. SeDyA: secure dynamic aggregation in VANETs. In *Proceedings of ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 131–142. ACM, 2013.
- [107] P. Vijayakumar, M. Azees, A. Kannan, and J. D. Lazarus. Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1015–1028, Apr. 2016.
- [108] L. Wang and X. Liu. Secure cooperative communication scheme for vehicular heterogeneous networks. *Vehicular Communications*, 11:46–56, January 2018.
- [109] P. Wang, C.-M. Chen, S. Kumari, M. Shojafar, R. Tafazolli, and Y.-N. Liu. HDMA: Hybrid D2D message authentication scheme for 5G-enabled VANETs. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

- [110] S. Wang and N. Yao. LIAP: A local identity-based anonymous message authentication protocol in VANETs. *Computer Communications*, 112:154–164, November 2017.
- [111] S. Wang and N. Yao. LIAP: A local identity-based anonymous message authentication protocol in VANETs. *Computer Communications*, 112:154–164, 2017.
- [112] S. Wang and N. Yao. A RSU-aided distributed trust framework for pseudonym-enabled privacy preservation in VANETs. *Wireless Networks*, 25(3):1099–1115, 2019.
- [113] S. Wang, N. Yao, N. Gong, and Z. Gao. A trigger-based pseudonym exchange scheme for location privacy preserving in VANETs. *Peer-to-Peer Networking and Applications*, 11(3):548–560, May 2018.
- [114] A. Wasef, Y. Jiang, and X. Shen. ECMV: Efficient certificate management scheme for vehicular networks. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM), 2008*. IEEE, 2008.
- [115] A. Wasef, Y. Jiang, and X. Shen. DCS: An efficient distributed certificate service scheme for vehicular networks. *IEEE Transactions on Vehicular Technology*, 59(2):533–549, Feb. 2010.
- [116] A. Wasef, R. Lu, X. Lin, and X. Shen. Complementing public key infrastructure to secure vehicular ad hoc networks. *IEEE Wireless Communications*, 17(5):22–28, Oct. 2010.
- [117] A. Wasef and X. Shen. MAAC: Message authentication acceleration protocol for vehicular ad hoc networks. In *Proceedings of IEEE GlobeCom, Nov. 2009*. IEEE, 2009.
- [118] A. Wasef and X. Shen. EMAP: Expedite message authentication protocol for vehicular ad hoc networks. *IEEE Transactions on Mobile Computing*, 12(1):78–89, 2013.
- [119] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn. A security credential management system for V2V communications. In *Proceedings of the 5th IEEE Vehicular Networking Conference (VNC 2013)*. IEEE, 2013.
- [120] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos. Privacy in inter-vehicular networks: why simple pseudonym change is not enough. In *Proceedings of 2010 Seventh International Conference on Wireless On-demand Network Systems and Services (WONS)*. IEEE, 2010.
- [121] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

- [122] Z. Xie, W. Ding, H. Wang, Y. Xiao, and Z. Liu. D-ary cuckoo filter: A space efficient data structure for set membership lookup. In *2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, pages 190–197. IEEE, 2017.
- [123] H. Xiong, K. Beznosov, Z. Qin, and M. Ripeanu. Efficient and spontaneous privacy-preserving protocol for secure vehicular communication. In *Proceedings of 2010 IEEE International Conference on Communications (ICC), May 2010*. IEEE, 2010.
- [124] B. Ying and D. Makrakis. Pseudonym changes scheme based on candidate-location-list in vehicular networks. In *2015 IEEE International Conference on Communications (ICC)*, pages 7292–7297. IEEE, 2015.
- [125] B. Ying and A. Nayak. Anonymous and lightweight authentication for secure vehicular networks. *IEEE Transactions on Vehicular Technology*, 66(12):10626–10636, 2017.
- [126] B. D. Ying and A. Nayak. Efficient authentication protocol for secure vehicular communications. In *Proceedings of IEEE 79th Vehicular Technology Conference*. IEEE, 2014.
- [127] R. Yu, J. Kang, X. Huang, S. Xie, Y. Zhang, and S. Gjessing. Mix-Group: Accumulative pseudonym exchanging for location privacy enhancement in vehicular social networks. *IEEE Transactions on Dependable and Secure Computing*, 13(1):93–105, Jan.-Feb. 2016.
- [128] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan. Vehicular ad hoc networks (VANETs): Status, results, and challenges. *Telecommunication Systems*, 50(4):217–241, 2012.
- [129] C. Zhang, X. Lin, R. Lu, and P. Ho. RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, pages 1451–1457. IEEE, 2008.
- [130] C. Zhang, X. Lin, R. Lu, P.-H. Ho, and X. Shen. An efficient message authentication scheme for vehicular communications. *IEEE Transactions on Vehicular Technology*, 57(6):3357–3368, Nov. 2008.
- [131] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen. An efficient identity-based batch verification scheme for vehicular sensor networks. In *Proceedings of IEEE 27th Conference on Computer Communications, (INFOCOM)*, pages 246–250. IEEE, April 2008.
- [132] H. Zhang, D. Zhang, H. Chen, and J. Xu. Improving efficiency of pseudonym revocation in VANET using cuckoo filter. In *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, pages 763–769. IEEE, 2020.

- [133] L. Zhang, C. Hu, Q. Wu, J. Domingo-Ferrer, and B. Qin. Privacy preserving vehicular communication authentication with hierarchical aggregation and fast response. *IEEE Transactions on Computers*, 65(8):2562–2574, 2016.
- [134] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu. Distributed aggregate privacy-preserving authentication in VANETs. *IEEE Transactions on Intelligent Transportation Systems*, 18(3):516–526, 2017.
- [135] L. Zhang, Q. Wu, B. Qin, and J. Domingo-Ferrer. Identity-based authenticated asymmetric group key agreement protocol. In *Proceedings of International Computing and Combinatorics Conference*, pages 510–519, 2010.
- [136] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer. A scalable robust authentication protocol for secure vehicular communications. *IEEE Transactions on Vehicular Technology*, 59(4):1606–1617, 2010.
- [137] L. Zhang and F. Zhang. A new certificateless aggregate signature scheme. *Computer Communications*, 32(6):1079–1085, 2009.
- [138] X. L. Zheng, C. T. Huang, and M. Matthews. Chinese remainder theorem based group key management. In *Proceedings of 45th ACMSE, Winston-Salem, NC, USA*, pages 266–271, 2007.
- [139] Y. Zheng. Digital signcryption or how to achieve $\text{cost}(\text{signature} \ \& \ \text{encryption}) \ll \text{cost}(\text{signature}) + \text{cost}(\text{encryption})$. In *Proceedings of 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997, Lecture Notes in Computer Science book series (LNCS, volume 1294)*, pages 165–179. Springer-Verlag, 1997.
- [140] Zhou and Y. H. Ou. Key tree and chinese remainder theorem based group key distribution scheme. *J. Chin. Inst. Eng.*, 32(7):967–974, Oct. 2009.
- [141] T. Zhou, R. Choudhury, P. Ning, and K. Chakrabarty. P2DAP - sybil attacks detection in vehicular ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 29(3):582–594, March 2011.
- [142] X. Zhu, S. Jiang, L. Wang, H. Li, W. Zhang, and Z. Li. Privacy preserving authentication based on group signature for VANETs. In *Proceedings of IEEE Global Telecommunications Conference Workshops, Dec. 2013, Atlanta, GA*. IEEE, 2013.

Vita

Shafika Showkat Moni

Education:

- University of Kentucky, Lexington, KY Aug. 2017 – May 2022
Ph.D in Computer Science
- University of Kentucky, Lexington, KY Aug. 2017 – May 2020
M.Sc in Computer Science
- Rajshahi University of Engineering and Technology Mar. 2010 - Dec. 2014
Rajshahi, Bangladesh
B.Sc. in Computer Science and Engineering

Work Experience:

- Graduate Teaching Assistant Aug. 2017 – May 2022
Department of Computer Science, University of Kentucky
- Lecturer Sep.2015 - Aug. 2017
Department of Computer Science, RUET, Bangladesh
- Research Assistant and Programmer Jan. 2015 - Sep. 2015
Institute of Information & Communication Technology, BUET, Bangladesh

Honors and Awards

- Best Ph.D student, Department of Computer Science, University of Kentucky, Spring 2022
- Best Ph.D student, College of Engineering, University of Kentucky, Spring 2022
- University of Kentucky Woman's Club Fellowship, Fall 2021
- CMD-IT/ACM Richard Tapia Scholarship, July 2021
- Nominated for Prestigious Presidential Fellowship, Spring 2021
- ACM-W Scholarship, May 2020
- Travel Grant from the Department of Computer Science, University of Kentucky, June 2020
- CRA-WP Graduate Cohort Travel Grant, March 2020
- Women in Cybersecurity (WiCyS) Student Scholarship, June 2020

- Grace Hopper Celebration (GHC) Scholar, Sept. 2019
- Dean's Award List, March 2012 – Dec.2014
- Student of the Year, Department of Computer Science and Engineering, RUET, July 2013
- Best Athlete, Mymensingh Girls' Cadet College (middle & high school), Dec. 2004 & Dec. 2005
- Merit scholarship in 5th Grade, June 2002

List of Selected Publications:

- **Shafika Showkat Moni** and D. Manivannan, "LEPA: Low-overhead and Efficient Privacy-preserving Authentication Scheme in VANETs", Elsevier Ad Hoc Network Journal, 2022 (to be submitted).
- **Shafika Showkat Moni** and D. Manivannan, "A lightweight Privacy-Preserving V2I Mutual Authentication Scheme using Cuckoo Filter in VANETs", 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC).
- **Shafika Showkat Moni** and D. Manivannan, "CREASE: Certificateless and REused-pseudonym based Authentication Scheme for Enabling Security and Privacy in VANETs", Elsevier Pervasive and Mobile Computing Journal (PMC), 2021 (Revised and Submitted).
- **Shafika Showkat Moni** and D. Manivannan, "A Scalable and distributed architecture for privacy -preserving authentication and message dissemination in VANETs", Internet of Things Journal, vol. 13, March 2021.
- **Shafika Showkat Moni** and D. Manivannan, "An Efficient RSU Authentication Scheme based on Merkle Hash Tree for VANETs", in Proceedings of 2020 IEEE International Conference on Communications (ICC 2020), 7-11 June 2020, Dublin, Ireland.
- D. Manivannan, **Shafika Showkat Moni**, and Sherali Zeadally. "Secure Authentication and Privacy-Preserving Techniques in Vehicular Ad-hoc NETWORKS (VANETs)", Vehicular Communications Journal, vol. 25, October 2020, Elsevier.
- **Shafika Showkat Moni** and D. Manivannan, "Privacy and Authentication in VANETs", International Journal of Next-Generation Computing, 11(2), 98-124, July 2020.
- Md. Abubakar Siddik, **Shafika Showkat Moni**, Mohammad Shah Alam, and William A. Johnson, "SAFEMAC: Speed Aware Fairness Enabled MAC Protocol for Vehicular Ad-hoc Networks", MDPI Sensors 2019.