2023

A Secure and Distributed Architecture for Vehicular Cloud and Protocols for Privacy-preserving Message Dissemination in Vehicular Ad Hoc Networks

Hassan Mistareehi
University of Kentucky, hassan.mistareehi@uky.edu
Author ORCID Identifier:
https://orcid.org/0000-0001-5906-3653
Digital Object Identifier: https://doi.org/10.13023/etd.2023.150

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.

Hassan Mistareehi, Student

Dr. D. Manivannan, Major Professor

Dr. Simone Silvestri, Director of Graduate Studies

A Secure and Distributed Architecture for Vehicular Cloud and Protocols for Privacy-preserving 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
Hassan Mistareehi
Lexington, Kentucky

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

Codirector: Dr. Sherali Zeadally, Professor of Communication and Information
Lexington, Kentucky
2023

ABSTRACT OF DISSERTATION

A Secure and Distributed Architecture for Vehicular Cloud and Protocols for Privacy-preserving Message Dissemination in Vehicular Ad Hoc Networks

Given the enormous interest in self-driving cars, Vehicular Ad hoc NETworks (VANETs) are likely to be widely deployed in the near future. Cloud computing is also gaining widespread deployment. Marriage between cloud computing and VANETs would help solve many of the needs of drivers, law enforcement agencies, traffic management, etc. The contributions of this dissertation are summarized as follows:

A Secure and Distributed Architecture for Vehicular Cloud: Ensuring security and privacy is an important issue in the vehicular cloud; if information exchanged between entities is modified by a malicious vehicle, serious consequences such as traffic congestion and accidents can occur. In addition, sensitive data could be lost, and human lives also could be in danger. Hence, messages sent by vehicles must be authenticated and securely delivered to vehicles in the appropriate regions. In this dissertation, we present a secure and distributed architecture for the vehicular cloud which uses the capabilities of vehicles to provide various services such as parking management, accident alert, traffic updates, cooperative driving, etc. Our architecture ensures the privacy of vehicles and supports secure message dissemination using the vehicular infrastructure.

A Low-Overhead Message Authentication and Secure Message Dissemination Scheme for VANETs: Efficient, authenticated message dissemination in VANETs are important for the timely delivery of authentic messages to vehicles in appropriate regions in the VANET. Many of the approaches proposed in the literature use Road Side Units (RSUs) to collect events (such as accidents, weather conditions, etc.) observed by vehicles in its region, authenticate them, and disseminate them to vehicles in appropriate regions. However, as the number of messages received by RSUs increases in the network, the computation and communication overhead for RSUs related to message authentication and dissemination also increase. We address this issue and present a low-overhead message authentication and dissemination scheme in this dissertation.

On-Board Hardware Implementation in VANET: Design and Experimental Evaluation: Information collected by On Board Units (OBUs) located in vehicles can help in avoiding congestion, provide useful information to drivers, etc. However, not all drivers on the roads can benefit from OBU implementation because OBU is currently not available in all car models. Therefore, in this dissertation, we designed and built a hardware implementation for OBU that allows the dissemination of messages in VANET. This OBU implementation is simple, efficient, and low-cost. In addition, we present an On-Board hardware implementation of Ad hoc On-Demand Distance Vector (AODV) routing protocol for VANETs.

Privacy-preserving approach for collection and dissemination of messages in VANETs: Several existing schemes need to consider safety message collection in areas where the density of vehicles is low and roadside infrastructure is sparse. These areas could also have hazardous road conditions and may have poor connectivity. In this dissertation, we present an improved method for securely collecting and disseminating safety messages in such areas which preserves the privacy of vehicles. We propose installing fixed OBUs along the roadside of dangerous roads (i.e., roads that are likely to have more ice, accidents, etc., but have a low density of vehicles and roadside infrastructure) to help collect data about the surrounding environment. This would help vehicles to be notified about the events on such roads (such as ice, accidents, etc.).Furthermore, to enhance the privacy of vehicles, our scheme allows vehicles to change their pseudo IDs in all traffic conditions. Therefore, regardless of whether the number of vehicles is low in the RSU or Group Leader GL region, it would be hard for an attacker to know the actual number of vehicles in the RSU/GL region.

KEYWORDS: Vehicular Ad Hoc Networks, Vehicular Cloud, Security and Privacy in Vehicular Networks, Arduino microcontroller

Hassan Mistareehi

April 27, 2023

A Secure and Distributed Architecture for Vehicular Cloud and Protocols for Privacy-preserving Message Dissemination in Vehicular Ad Hoc Networks

By
Hassan Mistareehi

Dr. D. Manivannan
Director of Dissertation

Dr. Sherali Zeadally
Codirector of Dissertation

Dr. Simone Silvestri
Director of Graduate Studies

April 27, 2023
Date

Dedicated to the soul of my mother
Farha Mistareehi,

and my father-in-law
Abdelqader Abdeljaber

who taught me how to be a good man!

ACKNOWLEDGMENTS

It is my pleasure to thank my advisor, Dr. D. Manivannan, for his guidance, patience, support, and advice throughout my graduate study. Dr. Manivannan has been a great mentor on every account, and his broad knowledge and constructive suggestions for this dissertation are greatly appreciated.

In addition, I would like to thank other faculty members Dr. Sherali Zeadally, Dr. Zongming Fei, Dr. Tingting Yu, and Dr. John Maddox for serving on my dissertation committee and for their valuable suggestions.

Finally, I thank my beloved family for their continuous support, care, encouragement, and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

Given the enormous interest shown by customers as well as the industry in autonomous vehicles, the concept of an Internet of Vehicles (IoV) has evolved from Vehicular Ad hoc NETworks (VANETs). Thus, VANETs are likely to play an important role in Intelligent Transportation Systems (ITS). According to some estimates, the global market for IoV is likely to exceed USD 200 billion by 2024. Many auto manufacturers have programs in place for developing a platform for connecting to IoV services such as route management and smart parking.

1.1 VANET and its Applications

VANETs are a type of Mobile Ad hoc NETworks (MANETs) that allow vehicles on roads to communicate among themselves and form a self-organized network. VANET communications are mainly classified into two major types. In the first type, vehicles communicate with other vehicles directly forming vehicle-to-vehicle communication (V2V). The second type is called vehicle-to-infrastructure communication (V2I) where vehicles communicate with roadside units (RSUs). Messages in V2V could be forwarded within a vehicle's transmission range to its neighbors and also the messages could be forwarded further through using a multi-hop routing protocol [117, 60, 109]. In V2I, communication takes place between vehicles and RSUs. Vehicles communicate with RSUs by using other vehicles as routers especially when RSUs are not within the transmission range of the vehicles.

VANETs have a large number of applications that can help drivers in many ways. Vehicles can collect, process, and broadcast information about themselves and their environment to other vehicles. For example, modern vehicles are equipped with Emergency Electronic Brake Lights (EEBL) which is a system that aims to warn other vehicles on the road in case there is a need for sudden hard-breaking or in the case of foggy weather where visibility may become low and brake lights are not bright enough to be recognized by other drivers. So, by using only V2V communication, vehicles can broadcast alert messages about the need for hard breaking ahead to other vehicles [6]. Improving intersection collision avoidance systems helps to avoid road accidents; this system is based on V2I communication. The infrastructure gathers, processes, and analyzes the information from the vehicles moving close to the intersection; depending on the analysis of data, if there is a possibility for an accident, a warning message is sent to the vehicles close to the intersection to warn them about the possibility of an accident so that they can take appropriate action to avoid it [84]. A dynamic traffic congestion pricing system for IoV [12] has been proposed. In this system, to alleviate traffic congestion, the participating vehicles are rewarded for taking an alternative path. The proposed system is implemented using VANETs, which eliminate the need for installing a costly electronic toll collection system. The authors in [11] proposed an accident prediction system for VANET. The crash risk in their system can be observed using velocity, driver fatigue, weather conditions, vehicles density, and crash

location. They used a hidden Markov model to model the correlation between these observations and the crash risk. The results of their proposed system show the ability to detect potential crashes [11].

1.2 Clustering Techniques in VANET

Clustering techniques have been used in V2V communication-based VANET architectures, wherein the network is divided into multiple clusters and one node in each cluster is selected as their Cluster Head (CH). The CH is responsible for all local cluster communication. This clustering technique helps with reducing the message overhead because it restricts the communication between CH and the members in its cluster. The CH can collect and also process and aggregate information from its cluster members and then propagate them to other clusters through other CHs [2, 111]. Many researchers proposed schemes [116, 94] for electing CHs in each cluster based on specific parameters, such as vehicle location, vehicle speed, etc. Dividing the network into multiple clusters reduces communication overhead and improves network efficiency. Note that, some clustering schemes do not scale well because frequent CH elections could occur if vehicles move fast. In addition, if CH fails, data aggregated by them may be lost.

1.3 Vehicular Cloud

In the last decade, cloud computing emerged as an economical solution for customers to rent IT infrastructures, platforms, or software, instead of investing money to own and maintain such services. The service providers give such flexible services to customers when they need them, and then they charge them based on their usage [55]. Modern vehicles are equipped with computing, communication, and storage resources, which often remain underutilized. Vehicular networks can also benefit from cloud computing. The *Vehicular Cloud* (VC) architecture which combines VANETs with cloud, was first proposed in [77] to fully capitalize the resources in VANET. In their approach, a VC is a collection of autonomous vehicles in VANET where vehicles contribute their underutilized computing, sensing, and communication resources to the cloud. Vehicle resources and the information shared by the vehicles with the cloud can be used in decision-making [37].

In VANET, multiple vehicles can observe the same phenomena and propagate it to other vehicles which can result in redundant propagation of data and waste the vehicle's resources. A vehicular cloud allows vehicles to exchange their collected data with the cloud where it can be analyzed, verified, organized, aggregated, and then propagated to the relevant vehicles/customers. Various other applications can also benefit from using a VC. Some of these applications include accident alerts, parking management, road conditions alerts, cooperative driving, and traffic management. The planned evacuation system is another application that could benefit from using the VC. When disasters like hurricanes occur, VCs can contribute to organized evacuations. Vehicles can also receive software updates from the cloud when vehicle manufacturers upload a new version of software [55, 108].

1.4 Security and Privacy in VANET

Ensuring security and privacy is an important issue in VANET and vehicular cloud; if information exchanged between entities is modified by a malicious vehicle, serious consequences such as traffic congestion and accidents can occur. In addition, sensitive data could be lost, and human lives also could be in danger. Hence, messages sent by vehicles must be authenticated and securely delivered to vehicles in the appropriate regions. Furthermore, privacy-related information such as the driver's name, position, and traveling route must be preserved. If vehicles cannot communicate anonymously, an attacker could easily trace vehicles by monitoring the messages sent by that vehicle [67].

1.5 Routing protocols in VANET

Routing protocol plays a vital role in extending the range of awareness in VANETs [31, 18]. Routing aims to establish routes from one node to other nodes, forward the packets in the network, and maintain and update the routes. There are two main types of routing protocols: proactive and reactive routing protocols. Reactive protocols are based on an on-demand mechanism in which each node in the network discovers or maintains a route when needed. This makes it suitable for VANETs due to the high mobility and vehicles density such that network traffic and bandwidth are reduced [51]. The works in [34, 88, 4] used simulation models for implementing and testing Ad hoc On-Demand Distance Vector (AODV) in VANET. AODV is one of the well-known reactive routing protocols [81]. In AODV, when a source node wants to send a data packet to a destination node, it first checks the available routes in its routing table. Then, if the route information is already in the table, the packet is sent to its destination. Otherwise, the source node broadcasts a route discovery request to all neighboring nodes. Then, this process continues until the request packet gets an intermediate node with a route to the destination or the destination node itself. When the route reply packet arrives from the destination or the intermediate node, the nodes forward it along the established reverse path and store the forward route entry in their routing table.

1.6 Motivation, Problems Addressed and Solved in the Dissertation

In this dissertation, we address and solve the following problems.

A Secure and Distributed Architecture for Vehicular Cloud: In VC, a malicious vehicle may impersonate to be another vehicle or an RSU to steal other drivers' sensitive information. Moreover, if vehicles cannot communicate anonymously, a malicious vehicle can track vehicles by linking the packets transmitted by that vehicle. Therefore, location privacy is identified as one of the main concerns in VANET [66]. Multiple vehicles may observe the same phenomena and forward these to the cloud which could result in the propagation of redundant messages. However, if information about observed phenomena is aggregated and stored in a cloud, this redundant

propagation of messages can be prevented. Hence, data aggregation is another crucial requirement for building an efficient vehicular cloud. In this dissertation, we present a secure and distributed architecture for vehicular clouds. In our architecture, vehicles collect data and forward it to the cloud where this data can be verified, analyzed, organized, aggregated, and then propagated to the relevant vehicles. In addition, our architecture ensures the anonymity and privacy of vehicles using pseudo IDs. We proposed an RSU-mix zone model where vehicles can change their pseudo IDs to increase their privacy.

A Low-Overhead Message Authentication and Secure Message Dissemination Scheme for VANETs: Efficient, authenticated message dissemination in VANETs are important for the timely delivery of authentic messages to vehicles in appropriate regions in the VANET. Many of the approaches proposed in the literature use RSUs to collect events (such as accidents, weather conditions, etc.) observed by vehicles in its region, authenticate them, and disseminate them to vehicles in appropriate regions. However, if traffic becomes heavy, it may not be possible for RSUs to receive messages about events observed by all vehicles in its region, authenticate them, and disseminate them in a timely manner, especially because the same event will be observed and sent by many vehicles in its region. We present a low-overhead message authentication and dissemination scheme in this dissertation. In our approach, when the density of vehicles in an RSU's region is high, the RSU divides the region within its transmission range into several sub-regions and selects one vehicle in each sub-region as the Group Leader (GL). The GL selected in a sub-region is supposed to collect messages sent by vehicles in its sub-region, authenticate them, aggregate them, and forward them to the RSU. This reduces the overhead related to message authentication for the RSU.

On-Board Hardware Implementation in VANET: Design and Experimental Evaluation: Information collected by On Board Units (OBUs) located in vehicles can help in avoiding congestion, provide useful information to drivers, etc. However, not all drivers on the roads can benefit from OBU implementation because OBU may not be available in all car models. In this dissertation, we design and built a hardware implementation for OBU that allows the dissemination of messages in VANET. This OBU implementation is simple, efficient, and low-cost. In addition, we present an On-Board hardware implementation of Ad hoc On-Demand Distance Vector (AODV) routing protocol for VANETs. Our developed hardware enables both V2V and/or V2I communications. Based on using V2V communication, vehicles can wirelessly communicate while moving along the road. Furthermore, the hardware implementation allows sending alert messages between vehicles in case of accidents or other road alerts.

Privacy-preserving Approach for Collection and Dissemination of Messages in VANETs: Many privacy-preserving approaches were proposed in the literature, but most of them don't consider safety message collection in areas where the density of vehicles is low and roadside infrastructure is sparse. These areas could

also have hazardous road conditions and may have poor connectivity. In this dissertation, we present an improved method for securely collecting and disseminating safety messages in such areas which also preserves the privacy of vehicles. We also present a pseudonym-changing method that reduces the chance of attackers linking two different pseudonyms of the same vehicle.

1.7 Organization of the Dissertation

The remainder of this dissertation is organized as follows.

- In Chapter 2, we present related works, which include the classification of vehicular cloud architectures, and challenges in implementing VANET/vehicular cloud; and we also discuss some solutions proposed in the literature to overcome these challenges and identify the drawbacks of the proposed solutions and discuss some open issues.
- In Chapter 3, we present a distributed, secure vehicular cloud architecture that ensures security and privacy in communication. Our scheme is scalable and has less communication overhead. Furthermore, it is capable of eliminating redundant messages through aggregation. The architecture is attacked resilient and can handle the failure of RSUs.
- In Chapter 4, we present a Low-Overhead message authentication and secure message dissemination scheme for VANETs. We use a clustering technique to reduce the overhead related to message authentication for the RSU.
- In Chapter 5, we present a hardware implementation of OBU which allows the dissemination of messages in VANET. In addition, we present a hardware implementation of Ad hoc On-Demand Distance Vector (AODV) routing protocol for VANETs.
- In Chapter 6, we present an improved method for securely collecting and disseminating safety messages in such areas that have a low density of vehicles and roadside infrastructure which also preserves the privacy of vehicles. In addition, we present a pseudonym-changing method that reduces the chance of attackers linking two different pseudonyms of the same vehicle.
- We conclude the dissertation in Chapter 7.

Chapter 2 Related Works

2.1 Vehicular Cloud Architectures: A Classification

In this section, we classify vehicular cloud architectures into three categories: temporary, permanent, and hybrid, and discuss their merits and demerits.

2.1.1 Temporary Cloud

A temporary cloud consists of vehicles that together form a cloud temporarily which allows them to share their resources (e.g., computing, networking, and storage) for collecting, processing, and disseminating information to other vehicles and other relevant customers as needed or carry out a requested task from other vehicles/entities.

The goal of vehicular cloud networking (VCN) proposed by Lee et al. [52] is to form a temporary vehicular cloud through a collaboration of vehicles in order to provide the needed services. Under VCN, one of the vehicles in the cloud is elected as a cloud leader based on some selected metrics (e.g., connectivity to other vehicles), and the rest of the vehicles cooperate in the cloud formation process. The cloud leader broadcasts a resource request (RREQ) message to vehicles within its range. Vehicles willing to share their resources (e.g., storage, phenomena observed by sensors, and computing resources) send a resource reply (RREP) message back to the cloud leader with information on their resource capabilities. After receiving RREP messages, the cloud leader selects cloud members and constructs a cloud. The cloud leader assigns tasks to cloud members taking into consideration the available resources of the respective members. The cloud members return the results back to the cloud leader after completing their tasks. After collecting the results from the cloud members, the cloud leader processes them and then publishes the final results. Vehicles may leave and join the cloud at any time; the cloud leader is responsible for managing the cloud. For example, when a vehicle leaves the cloud, the cloud leader selects another member in the cloud that has the necessary resources to complete the tasks assigned to the leaving member and assigns those tasks to that member. When the cloud leader no longer uses the cloud or moves out of the cloud, it sends a cloud release message to all the members, so they can join other clouds and contribute their resources [68].

The cluster-based vehicular cloud architecture proposed by Arkin et al. [10] uses a clustering technique to solve the resource allocation problem (e.g., some applications need more storage and computation resources) by grouping the vehicles according to the vehicle's location and velocity and allowing vehicles in the same group to provide resources. In this scheme, vehicles form clusters and cluster heads (CH) are selected using fuzzy logic. They model a cluster head selection algorithm that allows a selection of a set of optimal CHs. CHs are determined based on their FitFactor, defined in the paper. A CH is responsible for the creation, maintenance, and deletion of vehicles in the cluster. All vehicles in the cluster register their resources with the

CH. If a vehicle needs some resources from the vehicular cloud, it asks the CH. The CH is responsible for allocating all resources in the vehicular cloud. This approach is different from VCN [52] in the way in which CHs are selected. Figure. 2.1 shows the proposed cluster-based vehicular cloud architecture [10] and Table 2.1 summarizes the merits and demerits of temporary cloud architectures discussed above.

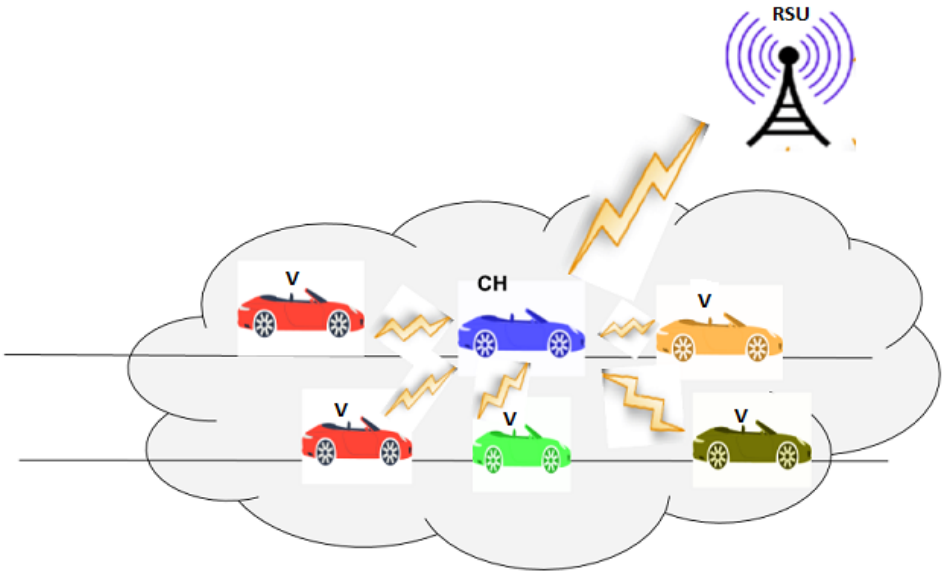

Figure 2.1: Temporary cloud using cluster based VC architecture [10].

Table 2.1: Merits and demerits of temporary cloud architectures discussed above.

Temporary cloud architectures	**Merits**	**Demerits**
VCN [52]	- Provides services to members of VCN. - The cloud leader distributes the tasks to cloud members, taking into account the availability of their resources. - Allows processing of information by using resources of all vehicles in the cloud.	- Computing/storage resources are limited compared to the conventional cloud. - The selection criteria of the cloud members and the format of the exchanged messages are not specified. - The cloud leader is a possible bottleneck because it is responsible for assigning tasks to cloud members, collecting results from cloud members, and managing the cloud. - If a cloud leader fails, data aggregated may be lost.
Cluster-based vehicular cloud [10]	- Uses a clustering technique to solve the resource allocation problem and to improve vehicular networks performance. - Predefined criteria are used for selecting Cluster Head.	- The CH is a possible bottleneck because it is responsible for assigning tasks to vehicle members, collecting results from them, and managing the cluster. - If the CH fails, data aggregated may be lost.

2.1.2 Permanent Cloud

A permanent cloud supports increased computing processing capacity. In a permanent cloud, vehicles send the phenomena collected as well as contribute their hardware resources to the cloud. Then the permanent cloud processes the data gathered from the vehicles and provides the needed services to vehicles/drivers (e.g., parking information, road conditions, accidents, traffic information, etc).

Hussain et al. [44] proposed a Cooperation-aware VANET cloud. In their approach, vehicles and cloud infrastructure cooperate with each other to provide drivers with services such as traffic information and warning messages. Vehicles share observed phenomena with the cloud and the cloud processes the data gathered from the vehicles and shares the information with the vehicles in the cloud. This model has two levels of architecture: The first level is VANET, which consists of vehicles serving both as producers (they send information to the cloud) and consumers (they get information from the cloud). The second level is the permanent cloud, which consists of Authenticator, Cloud Collecting Point(CCP), Cloud Knowledge Base(CKB), and

Cloud Decision Module(CDM). The authenticator is responsible for handling contributions from vehicles and authenticating them. The data are collected at CCP and sent to CKB for processing, and then the processed information is shared with the VANET users. The virtualization layer that works as a mediator between VANET and the cloud, takes inputs from VANET and passes them to the cloud, and disseminates the output from the cloud to vehicles in the VANET.

Wan et al. [100] proposed a context-aware architecture with mobile cloud support for vehicular cyber-physical systems (VCPS). Context awareness allows the adaptation of services according to many factors such as occurred changes in the environment, user preferences, user location, and capabilities of mobile devices. The applications and services in VCPS are divided into three different computational layers: location computational layer, vehicle computational layer, and cloud computational layer. In the location computational layer, RSUs deployed at strategic locations can exchange information with OBUs installed on vehicles. When a vehicle with an OBU passes by an RSU, it can receive updated traffic information from the RSU and share its own information (e.g., destination and vehicle route data) with the RSU. Vehicles that do not have an RSU within their transmission range can connect to RSUs through neighboring vehicles. In the cloud computational layer, there are multiple systems working with each other to share resources and provide a number of services such as vehicle multimedia content and traffic information.

Salahuddin et al. [87] proposed a vehicular cloud architecture, called RSU cloud, which uses RSUs and data centers. The RSU cloud provides services that meet changing demands from vehicles. The RSU cloud architecture exploits the flexibility and deep programmability offered in software-defined networking (SDN). In SDN, there are two communication planes, the physical data plane, and an abstracted control plane. This decoupling of control and data planes enables the flexibility and programmability of the SDN. Drivers register with the RSU cloud and the cloud keeps track of the driver's status such as stability (alcoholic or not) and current location. Users can request service from the RSU cloud. Based on the user's request, RSU cloud will respond to users. In the RSU cloud, virtualization via virtual machines (VMs) and SDN is employed to dynamically instantiate, migrate, replicate services, and reconfigure data forwarding rules in the network to meet the frequently changing service demands.

The cloud-based system proposed by Wang et al. [107] consists of a cloud, several types of radio access networks (RANs), and a set of vehicles. Vehicles are assumed to be equipped with GPS and OBUs. In the cloud, there are various servers, which could be either real physical machines or virtual machines. Vehicles from time to time report their status such as current location and speed through RSU to the cloud. RSUs send updates on their status such as the number of active vehicles and traffic load to the cloud. The cloud then processes the collected data and disseminates the information to vehicles that are in need.

Lim et al. [55] proposed a secure incentive-based architecture for the vehicular cloud to encourage vehicles to participate in the cloud. Tokens are given to the vehicles as a reward to participate in the cloud and vehicles can use the token to get services from the cloud. Their scheme has three phases. In phase 1, the service

provider manager (SPM) sends a message asking vehicles for sharing their resources. When an interested vehicle receives the message and wants to share its resources with the cloud, it sends a message to the SPM through the RSUs. Then, the SPM authenticates the vehicle with the help of a trusted authority (TA). Once the vehicle is authenticated, the SPM signs a contract between the service provider and the vehicle and sends it to the vehicle, so the vehicle can start allowing its resources to be used by the cloud. Every vehicle uses its pseudo ID in all communications to protect its privacy. In phase 2, a vehicle sends a message with proof of the work done to the SPM. The SPM verifies the proof of the work done and sends a reward token request to the reward token system (RTS) so it can send tokens to the vehicle. In phase 3, the reward token earned for participating in the cloud is used as payment for the cloud services obtained. Vehicles can check their token balance with the OBUs to buy services from the cloud. Figure. 2.2 illustrates the secure architecture for the vehicular cloud that encourage vehicles to participate in the cloud [55] and Table 2.2 summarizes the merits and demerits of permanent cloud architectures discussed above.

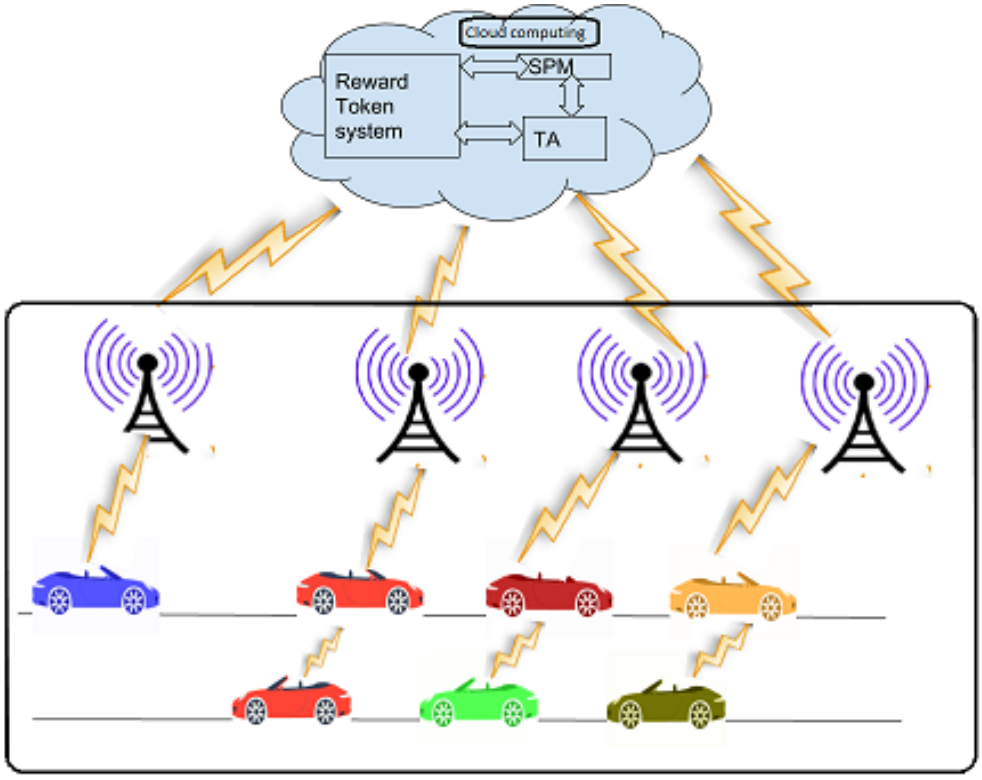

Figure 2.2: Permanent cloud using incentive-based architecture for VC [55].

Table 2.2: Merits and demerits of Permanent cloud architectures discussed in this section.

Permanent cloud architectures	**Merits**	**Demerits**
Cooperation-Aware VANET Cloud [44]	- This scheme provides vehicles with traffic information and warning messages.	- This architecture does not encourage vehicles to participate in the cloud.
Context-aware architecture [100]	- They create a context-aware pervasive system for mobile vehicles, drivers, passengers, and relevant traffic authorities by designing a multi-layered architecture with cloud capability. - Each layer provides multiple context-aware services.	- Security issues are not addressed.
RSU cloud [87]	- The RSU cloud hosts services to meet the demand from the OBUs in the vehicles. - The RSU cloud architecture explores the benefits of the flexibility and deep programmability offered in SDN. - In the RSU cloud, virtual machines (VMs) and SDN are used to dynamically instantiate, migrate, and/or replicate services and dynamically reconfigure data forwarding rules in the network to meet the frequently changing service demands.	- Despite the benefits of the programmability of RSU clouds, service instantiation, migrations, replication, and network reconfiguration will result in large overhead.
Cloud-based system [107]	- This approach helps in overcoming the limitations of vehicles for enabling advanced services. - It supports a larger network of up to eight times in size, compared with the one using the conventional cloud.	- Large communication latency and packet losses caused by connectivity discontinuity will make the information provided by the cloud unusable.

A secure incentive-based architecture for VC [55]	- This architecture encourages vehicles to contribute their underutilized resources to the cloud by issuing tokens that can be used by the vehicles to get services from the cloud. - Token transaction is secure and robust against attacks. - Integrity and authenticity of the messages exchanged between entities are ensured. - Privacy of vehicles is protected.	- If an RSU fails, data aggregated may be lost and will not be delivered in time.

2.1.3 Hybrid Cloud

A hybrid cloud is a combination of a temporary cloud and a permanent cloud. Vehicles can access permanent cloud as well as temporary cloud formed by vehicles to accomplish a specific task. A permanent cloud (stationary cloud) provides support for various software applications, computing, and processing capabilities (e.g., storage devices, processors, servers, etc.) to vehicles. A temporary cloud, using vehicular resources such as OBUs provides sensing information as a service and provides services such as support for communication infrastructure. So, the hybrid cloud benefits from both temporary and permanent clouds and provides users with better services.

Bitam et al. [17] proposed the VANET-cloud model to improve traffic safety and provide services to drivers. Their proposal uses both permanent and temporary clouds. Permanent cloud, which consists of stationary nodes (e.g., servers, workstations, etc.) offers cloud services such as software as a service (SaaS), infrastructure as a service (IaaS), and platform as a service (PaaS) to vehicles. In the VANET cloud, the temporary cloud consists of vehicles that have computing resources (e.g., OBUs) installed on them and these vehicles together form a cloud. Their temporary VANET-cloud model consists of three layers. The client layer, formed by end users (an end user might be a general customer) use communication and computing devices such as smartphones, laptops, OBUs, and GPS and the end user can initiate his/her service request through a service access point (SAP). The second layer is a communication layer that connects the client layer with the cloud layer. This layer consists of several communication devices and networks such as VANETs, 3G/4G networks, cellular base stations, RSUs, and so on. The third layer is the cloud layer, which consists of a stationary cloud and a temporary cloud. The interconnection between permanent and temporary VANET clouds is enabled by a network consisting of all data centers of both VANET clouds. Therefore, the provider is responsible for managing and controlling the merged network using different networking techniques and protocols. For example, each vehicle in the temporary VANET cloud can access the permanent VANET cloud, and each node in the permanent VANET cloud can establish a connection with temporary VANET cloud nodes leading to a global network

controlled by the service provider. As a result, a large and flexible vehicular cloud can be formed to serve many types of end users.

The three-layer architecture (V-Cloud) proposed by Abid et al. [1] combines in-car vehicular cyber-physical systems, vehicle-to-vehicle network (V2V), and vehicle-to-infrastructure network (V2I) layers to help improve the safety and comfort of the drivers. The in-car layer consists of two types of sensors, which are the vehicle's internal physical sensors and smartphone-embedded sensors. Smartphones monitor the health and mood conditions of the driver through embedded sensors and send the collected information to the cloud. The cloud will store this information, which can help predict the mood of drivers early. Vehicles in the V2V network organize themselves into clusters. Each cluster has a cluster head to send all the information to other vehicles in the cluster as well as to neighboring cluster heads. Each cluster head will identify whether it is near any access point or not in order to transmit its cluster-needed information to the cloud computing environment. In V2I, vehicles connect with the cloud through RSUs.

Chaqfeh et al. [27] presented a model for vehicular cloud data collection for Intelligent Transportation Systems (ITSs) to provide route guidance and navigation alternatives based on information about road conditions. This model consists of three phases: In phase 1, a requesting vehicle broadcasts a route request (RREQ) to its desired region of interest (ROI) through one-hop neighbors. In phase 2, when the vehicles at the desired ROI receive RREQ, they cooperate to collect the desired data like environment condition data and sensed vehicular data (e.g., speed or distance). Vehicles at ROI form a vehicular cloud, and every vehicle competes to be a broker and the roadside unit (RSU) manages the process of broker election based on the connectivity criteria. When a broker is elected, the broker will collect the desired data from the members and then the broker sends it to a server in the internet cloud if further processing is required. In the case of a simple request, VC resources may be sufficient to send a response. In complex cases, the broker communicates with the Internet cloud to allocate the required computing resources. In phase 3, a route reply message (RREP) is created by the broker and then sent to the requesting vehicle. In case of simple requests like traffic conditions, the request would be processed in the vehicular cloud. But in complex cases like finding alternate routes to avoid traffic congestion, the server from the Internet cloud would send a response to the broker. Fig 2.3 presents an architecture of this Hybrid Cloud. Table 2.3 summarizes the merits and demerits of hybrid cloud architectures discussed in this section.

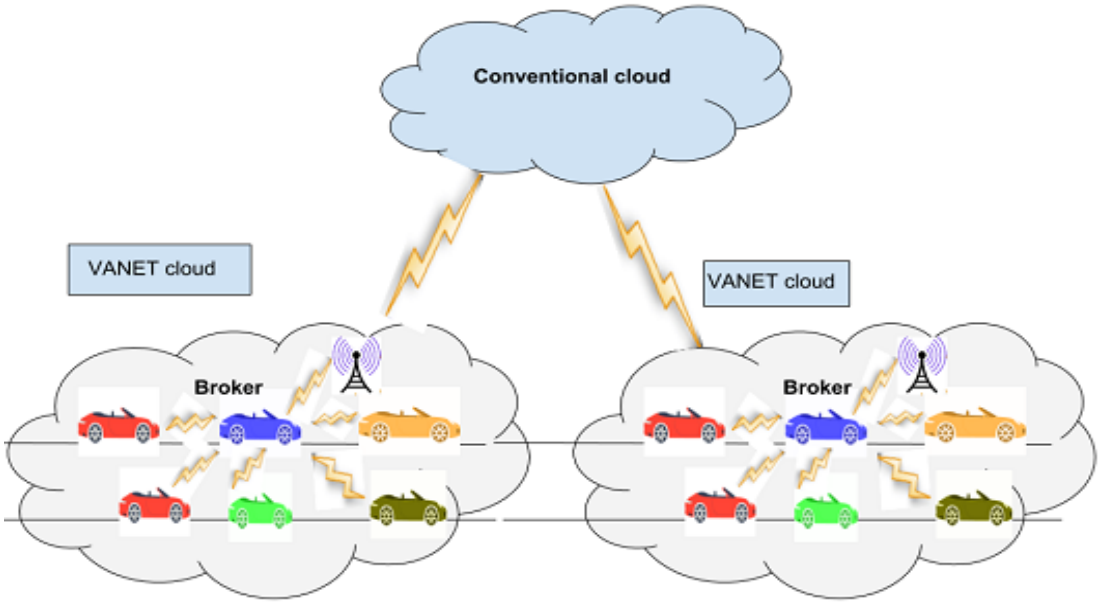

Figure 2.3: Hybrid cloud VANET cloud and conventional cloud [45].

Table 2.3: Merits and demerits of hybrid cloud architectures presented in this section.

Hybrid cloud architectures	**Merits**	**Demerits**
VANET-CLOUD [17]	- This model benefits from the computing capabilities of vehicles that support processing, storage, as well as sensing to extend traditional cloud computing capabilities.	- Mobility of vehicles in the temporary cloud can affect the performance of VANET-Cloud applications. - They did not address the security and privacy issues.
V-cloud [1]	- Combines the concept of VANET, CPS, and Cloud Computing to provide safety and comfort for drivers.	- Security and privacy issues are not addressed.
Merging VANET with cloud computing [45]	- Authors suggest a concrete VANET cloud architecture to use underutilized onboard computing and communication units of vehicles.	- They did not address authentication, security, and privacy issues.

VC data collection for ITSs [27]	- Provides route guidance and navigation alternatives based on information about road conditions.	- If the broker fails, data aggregated may be lost, and the cloud will not be able to do the intended functionalities.

2.2 Challenges and Some Proposed Solutions

In this section, we highlight some of the challenges in implementing VANET/VC. These challenges include the authenticity and integrity of messages, preserving the privacy of vehicles/drivers, handling selfish nodes, routing, data processing, etc. Next, we describe these challenges and present a critical comparison of the solutions proposed in the literature addressing these challenges.

2.2.1 Authentication, Integrity and Privacy Preservation

Authentication is one of the important requirements in the vehicular cloud. A receiver should be able to verify that a transmitted message has been sent by an authentic member. For example, a single vehicle can claim to be one of hundred vehicles in order to give wrong information about the congested road. Sharma et al. [92] proposed a dynamic key-based authentication scheme for vehicular cloud computing for mutual authentication of senders and receivers.

Message integrity ensures that an intruder is not able to modify a message. Some of the driver information that needs to be protected from intruders is driver identity, trip path, and speed [89]. So when messages are disseminated, the authenticity of the vehicles that are disseminating messages should be verified, the integrity of the messages should be guaranteed and the privacy of vehicles should be preserved. Raya et al. [85] used a set of anonymous keys to ensure privacy; these keys are changed frequently and each key can be used once only and expires after its usage. These keys are stored in the vehicle's tamper-proof device (TPD). The TPD is responsible for all the operations related to key management and usage. Each key is certified by the issuing Certificate Authority (CA) and has a short lifetime.

Lin et al. [57] proposed a secure and privacy-preserving protocol for VANETs. This scheme provides privacy to ensure the safety of the drivers. Group signatures are used to secure communication between vehicles and ID-based signatures are used to secure communication with roadside units (RSUs). There are two types of managers, the membership managers which provide security and system parameters to RSUs and send private and group keys to these units, and the traffic managers who are responsible for collecting information when the identities of vehicles need to be revealed. When a vehicle is determined as malicious, that vehicle will be excluded from the system and new group and private keys are generated for the remaining safe vehicles and are sent out.

Lim et al. [55] proposed secure incentive-based architecture for the vehicular cloud. They used hash-based digital signature along with public key encryption to ensure the

integrity and authenticity of messages. When a message is sent, the sender attaches the digital signature to the message. The digital signature is made by encrypting the hash of the message using the sender's private key. The service provider manager (SPM) is connected to the trusted authority (TA) in the cloud. The TA helps the SPM in authenticating the sender. If the hash in the digital signature matches with the hash of the message calculated by the receiver, then the receiver is able to verify the authenticity and integrity of the message, but if the hash of the message does not match, the message is discarded upon arrival, so integrity is guaranteed. Also, the privacy of vehicles is protected by assigning unique pseudo IDs for each vehicle. These pseudo-IDs are used in all communication to protect the real identities of vehicles. If a malicious node is detected, the real ID of the malicious vehicle is revealed by the TA to the authorities for legal investigation.

2.2.2 Selfish Nodes Problem

In a VC, some vehicles, called selfish nodes, may not contribute their resources or collected phenomena to the cloud but only would like to exploit the resources of other vehicles; some schemes were designed to encourage selfish nodes to join the VC and contribute their resources. Lim et al. [55] proposed a secure architecture for the vehicular cloud to encourage vehicles to participate in the cloud by contributing their underutilized resources to the cloud. In this method, tokens are given to the vehicles as a reward to participate in the cloud and they can use it to get services from the cloud.

An incentive framework for the vehicular cloud on the road was proposed by Kong et al. [49] to encourage vehicles to contribute their under-utilized on-board resources to VC. It consists of the following types of entities: task server, RSU, leader vehicle, and vehicles. The task server selects a leader for vehicles and the leader works as the controller of VC. The task server is responsible for searching for on-board resources for the intelligent vehicles, managing the registration of vehicles, distributing the keys to the vehicles, and also maintaining accounts for the registered vehicles. RSUs serve as a gateway between the task server and vehicles and collect and transmit the related information generated by the vehicles to the task server and also send the messages from the task server to vehicles within their transmission range. The leader vehicle selects vehicles to collaborate for completing tasks and is responsible for organizing the on-board resources of the vehicles, publishing tasks to the vehicles, assigning tasks to each vehicle, calculating payments for the participating vehicles, and collecting results after vehicles complete the tasks allocated to them. After vehicles complete the tasks, the leader vehicle sends the vehicles' rewards to the task server through its closest RSU. Then the task server updates the rewards for each vehicle in its account.

2.2.3 Vehicular Cloud Management

Change in the number of resource providers (vehicles) over time affects cloud management. For example, some vehicles are parked in the parking lot for several days,

and if their owners agree to rent their resources, these vehicles need to be plugged into a power outlet to share their resources. The available resources can change depending on the arrival and departure of vehicles. A vehicle can leave the parking lot while its resources are being used by some applications. So, the issue is how to take into account the unpredictable nature of the vehicles arriving and departing parking lots to schedule resources and assign computational tasks to the various vehicles in the vehicular cloud. Other issues that may affect the stability of the cloud are the vehicle's velocity, and broken V2V and V2I communications due to interferences and obstacles [64].

In [10, 24], the authors propose several clustering methods for electing a cloud controller or a set of leaders to address the above challenges in VANET clouds. Authors in [10] have proposed a clustering technique to solve the resource limitation problem. Their method groups vehicles and vehicles in each group cooperate with each other to contribute their resources. Since some applications require large amounts of data to upload, download, and store, these applications need more storage and computation resources. Furthermore, they assume that all vehicles are equipped with a positioning system like GPS to get information about their location. In this method, vehicles form clusters and the ones that are more appropriate become cluster heads (CH). The CH is responsible for the creation, maintenance, and deletion of a vehicular cloud. All vehicles will register their resources with the CH and cloud resources are scheduled by CH. If a vehicle needs some resources from the vehicular cloud, it asks CH and CH selects the best cluster member (vehicle with sufficient unutilized resources) from the cluster to complete the requested service.

2.2.4 Routing and Data Processing in Cloud

Route selection is an essential factor to avoid congested routes and transfer the data to the destination in a reasonable amount of time. In the vehicular cloud, the selection of paths is challenging due to the high mobility of vehicles. The selection of an entity for data processing is also challenging. Processing data can be done inside vehicles, infrastructure equipment (e.g., RSUs), or conventional cloud.

Kumar et al. [50] proposed a cloud-assisted design for autonomous driving, which allows the cloud to access sensor data from autonomous vehicles as well as RSUs to assist autonomous cars in planning their routes. Moreover, it assists vehicles with determining efficient routes and avoiding obstacles such as road work, accidents, traffic jams, etc. The cloud records the current route of all the vehicles and collects information about all obstacles. Then, the cloud sends alternate routes avoiding the obstacles. Thus, the cloud can assist vehicles in determining efficient routes. Authors in [82] proposed VehiCloud architecture to provide routing service for vehicular networks. Vehicles in VehiCloud monitor certain conditions in certain areas as well as predict their future locations. This information is sent to the Cloud decision module through terminals (e.g., RSUs), which are responsible for making the routing decision. As a result, the cloud can predict future traffic information by collecting the trajectory information of vehicles.

Wang et al. [101] proposed a body area sensor network (BASN) formed by vehicular devices and sensors attached to the drivers for measuring bio-medical information. It consists of a set of layers that process the collected data before making a suggestion to the driver. In the repository layer, data can be classified into low-level context (e.g., temperature, blood pressure) and high-level context (e.g., gesture, activity) based on pre-processing techniques. In the knowledge processing layer, low-level and high-level context will be processed using techniques such as data mining, reasoning, k-means clustering, etc. Finally, some outputs will be delivered through the context-aware middle layer to the upper actuator layer and some actions will be taken like stopping the car, turning slow, and sending alarm signals to the driver.

2.2.5 Virtualization in Vehicular Cloud

A vehicular cloud is a dynamic environment due to the mobility of vehicles; moreover, vehicular clouds have limited computing and storage capacity compared to traditional clouds. Thus, virtual machine (VM) management seems to be a challenge in vehicular clouds. For example, when the vehicles in a parking lot are used as data centers, it is possible to store the data of the customers in a vehicle temporarily. This data needs to be moved from the devices before the vehicle leaves the parking lot. Therefore, the VC requires a virtual machine to manage the physical devices used to process or store data [103].

Yu et al. [114] studied cloud resource allocation and VM migration for effective resource management in cloud-based vehicular networks. They presented different scenarios of VM migration due to vehicle movement. Figure. 2.4 illustrates different VM migration scenarios. In the first case, when vehicle A moves from the coverage area of RSU-1 to RSU-2, a VM migration is needed. Since RSU-1 and RSU-2 connect to different cloudlets, guest VM-A should be transferred from roadside cloudlet-1 to roadside cloudlet-2. After that, A will access cloudlet-2 via RSU-2 to resume its service. In the second case, when vehicle A moves from the coverage area of RSU-1 to RSU-2. Since these two RSUs connect to the same roadside cloudlet, there is no need for VM migration. However, radio handoff from RSU-1 to RSU-2 may still take a short period. In the third case, vehicle A moves from the coverage area of RSU-2 to RSU-1. Before A's movement, nodes A, C, and D have connections in an ad hoc manner. Vehicle C access the roadside cloud through vehicle A. The movement of A will cause the disconnection of C from the roadside cloud. In this case, guest VM-C will be transferred from the roadside cloud to the vehicle cloud in D. Then, vehicle C can continue its service through D. The last case is similar to that of case three, except that there is no direct link between vehicles C and D. In this case, guest VM-C has to be migrated from the roadside cloud to the central cloud. After that, C will access the central cloud to resume its service using long-distance communications such as 3G/4G cellular networks.

The VM migration scheme proposed by Reffat et al. [86] for vehicular cloud works as follows: The source node chooses a destination node depending on the search criteria. If the destination node does not have enough resources to host a VM or if the VM cannot be migrated to the destination node in a pre-defined time window,

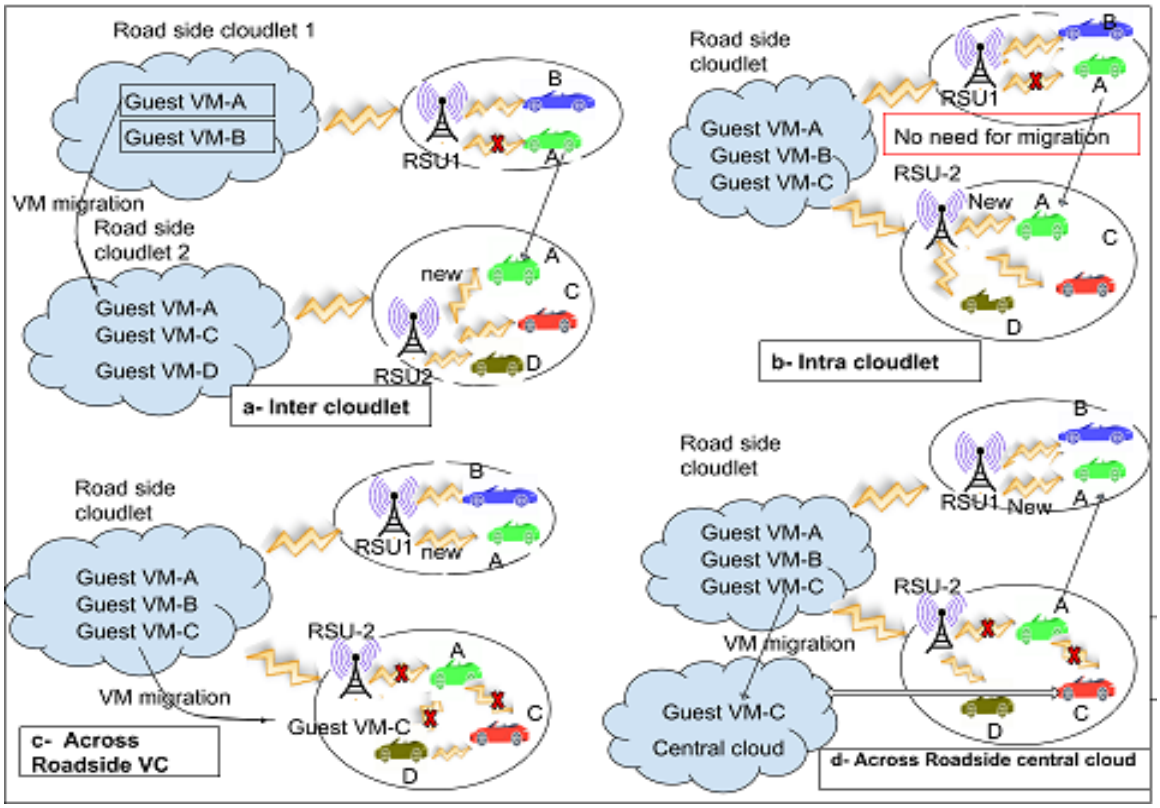

Figure 2.4: Virtual machine migration scenarios [114].

migration is retried by excluding that destination. Otherwise, the VM is migrated to the destination node. After a certain number of migration attempts fail, migration is marked as unsuccessful. If migration is unsuccessful, the VM is directed to the RSU. The authors proposed two methods against the random selection of the destination node. The first one is the vehicular virtual machine migration with the least workload (VVMM-LW) and the second one is vehicular virtual machine migration with mobility awareness (VVMM-MA). The first approach selects the vehicle with the lightest workload among the vehicles that are predicted to remain in the network as the destination node to migrate the virtual machine. The second approach uses the vehicle's routes and approximations of the future locations of all vehicles in the vehicular cloud and excludes the vehicles that are predicted to go off the network.

2.2.6 Context Awareness in Vehicular Cloud

Context-aware information can provide more convenience and safety for drivers and passengers. For example, a context-aware service could be a live video of a planned route for the driver or a real-time traffic update [98]. Sultan et al. [5] proposed a context-aware driver behavior detection system to detect irregular behavior of drivers

and notify other drivers on the road to prevent accidents from happening. This architecture is divided into three phases, namely, sensing, reasoning, and acting. In the sensing phase, the system collects information about the driver, the vehicle's state, and environmental changes. The reasoning phase involves reasoning about uncertain contextual information to get the behavior of the driver. They designed a Dynamic Bayesian Network (DBN) model to perform probabilistic reasoning to infer the behavior of the driver. This model combines information collected from different sensors capturing the driver's behavior and uses probabilistic inference to get the driver's current driving style. The driver and other vehicles are then alerted by triggering an in-vehicle alarm and by sending warning messages containing corrective actions to other vehicles in the VANET.

The multi-layer context-aware vehicular cloud architecture proposed by Wan et al. [100] has the following three layers: Vehicular computational layer, location computational layer, and cloud computational layer. In the vehicular computational layer, a context-aware driver behavior detection system is implemented. This system communicates with other vehicles to share the context-aware road and safety information. The location computational layer uses the RSUs deployed at specific locations on the road to exchange information with OBUs. The cloud computational layer provides context-aware cloud services through interconnected clouds of automotive multimedia content cloud, traffic authority cloud, location-based service cloud, automotive manufacturer cloud, and other application clouds. The authors describe cloud-assisted parking services that address traditional parking garage scenarios for drivers. The context information of each parking space detected by sensors is forwarded to the traffic cloud through wireless sensor networks (WSNs), 3G communication networks, and the Internet. The collected data are processed in the cloud and then selectively transmitted to the drivers. This is helpful for providing more convenient services and evaluating the utilization levels of the parking garage. Table 2.4 summarizes the merits and demerits of the solutions discussed above.

Table 2.4: Merits and Demerits of proposed solutions

Proposed solutions	Issues addressed	**Merits**	**Demerits**
GSIS [57]	Authentication, integrity and privacy preservation	- This scheme ensures the privacy of the vehicles	- Cryptographic operations can cause large overhead.
Secure incentive-based architecture [55]	Authentication, integrity and privacy preservation	- Their scheme ensures source authentication, message integrity, and privacy preservation.	- This scheme assumes service providers are trustworthy.

Secure incentive-based architecture [55]	Selfish nodes problem	- Encourages vehicles to participate in the cloud by giving them tokens as a reward which they can use to get services from the cloud.	- If the RSU fails, the received and collected data by the RSU can be lost.
A secure and privacy-preserving incentive framework for VC [49]	Selfish nodes problem	- Vehicles can earn payments for participating in and completing the accepted tasks.	- This proposed method didn't take into account the security issues related to guaranteeing the availability of the incentive mechanism. - The leader vehicle is a bottleneck because it is responsible for all tasks. - If the leader vehicle fails, data aggregated could be lost.
A cluster-based VC with learning-based resource management [10]	Vehicular cloud management	- This scheme uses clustering technique to provide resources cooperatively. - CH chooses a vehicle that has sufficient resources to complete a requested service.	- The resource allocation algorithm may cause service delays. - CH is a bottleneck because it is responsible for all tasks. - If the CH fails, data aggregated is lost.
A cloud-assisted design for autonomous driving [50]	Routing and data processing	- This scheme enables autonomous cars to plan safer and more efficient paths by sharing their sensor information with the cloud.	- Autonomous vehicles often require accurate localization. This is not addressed.
Cloud computing facilitated routing in vehicular networks [82]	Routing and data processing	- Provide routing service for vehicles in the network.	- Security issues are not addressed.

BASN [101]	Routing and data processing	- Provide some real-time services based on cloud computing techniques. - BASN, with context-aware reasoning and knowledge processing techniques, can improve drivers' safety and comfort.	- Security issues are not addressed.
Toward cloud-based vehicular networks with efficient resource management [114]	Virtualization in VC	- They studied different scenarios to migrate VMs to provide services to vehicles.	- Central cloud has sufficient cloud resources but end-to-end communications delays can be large.
Dynamic VM migration in a VC [86]	Virtualization in VC	- Aims to handle frequent changes in VC topology efficiently. - Increased fairness in vehicle capacity utilization across VC.	- Does not migrate the workload to multiple destinations simultaneously to maximize chances of success.
Context-aware driver behavior detection system [5]	Context awareness in VC	- Supports improving road safety. - Helps detect irregular behaviors of drivers and notify other drivers on the road to prevent accidents from happening.	- Does not suggest appropriate corrective actions for other vehicles on the road.
Context-aware vehicular cyber-physical systems [100]	Context awareness in VC	- Helps in improving road safety and traffic management.	- Security issues are not addressed.

2.2.7 Drawbacks of the above Proposed Solutions and Open Issues

In this section, we present some of the drawbacks of the solutions presented in this section and discussed some open issues that need further investigation in the vehicular cloud environment.

Cluster-based vehicular cloud architectures have been proposed in [10] and [36], by grouping vehicles according to their location and speed. Both of these schemes rely on a cluster head (CH) which is elected by the vehicles in the cluster, and this CH performs the creation, maintenance, and deletion of all the vehicles in that cluster. A similar approach is proposed by Chaqfeh et al. [27], where vehicles in a specific region form a vehicular cloud and elect a broker among them. The broker collects the

desired data from the vehicles and then sends it to a cloud server if further processing is required. None of these schemes scale well as the number of vehicles increases. When vehicles are moving fast, frequent CH and broker elections occur which result in large message overhead. In addition, if CH or broker fails, data aggregated by them may be lost.

In other architectures [44, 107] which combine VANET and cloud, vehicles collect data and send them to the cloud through a mediator or an RSU. The vehicles in the same area could collect the same data, so this leads to redundancy and results in a large message overhead. However, if information about observed phenomena is aggregated and stored in a cloud, this redundant propagation of messages can be prevented. Hence, data aggregation is another crucial requirement for building an efficient vehicular cloud. Like [10, 36], they also suffer from a single point of failure– if the mediator or the RSU fails, data aggregated could be lost.

Many of the solutions proposed [100, 45, 1, 17] do not address the security and privacy issues. Ensuring security and privacy is an important issue in VANET/VC; if information exchanged between entities is modified by a malicious vehicle, serious consequences such as traffic congestion and accidents can occur. In addition, sensitive data could be lost, and human lives also could be in danger. Hence, messages sent by vehicles must be authenticated and securely delivered to vehicles in the appropriate regions. Furthermore, privacy-related information such as the driver's name, position, and traveling route must be protected. If vehicles cannot communicate anonymously, an attacker could easily trace vehicles by monitoring the messages sent by that vehicle.

Several privacy-preserving authentication schemes such as cooperative authentication [47], anonymous authentication [13], dual authentication [58], cloud secure communication [61], and RSU-aided message authentication [115] have been proposed. In these schemes, vehicles communicate not only with each other but also with the RSUs or the Trusted Authority (TA) to verify the authenticity of the messages. Although they support authentication and privacy, these schemes suffer from communication overhead and do not scale well. If network traffic becomes heavy, it might not be possible for vehicles to authenticate, process, and forward messages in a timely manner. This could also result in message loss and redundant message propagation. Several other schemes [47, 13, 58, 56, 61] exist in the literature for solving authentication and privacy issues in communication; however, many of them [47, 13, 56] do not guarantee the confidentiality of exchanged messages and therefore, are vulnerable to attacks.

As we have seen, some schemes lead to redundancy and result in large message overhead, and some other schemes suffer from a single point of failure. In addition, many of the other schemes do not address security and privacy issues. Some only address authentication and privacy, but they suffer from communication overhead and do not scale well. Furthermore, some of the other schemes do not guarantee the confidentiality of exchanged messages. Therefore, we proposed architecture to address these issues. In the next chapter, we present a distributed architecture for the vehicular cloud that ensures security and privacy in communication. Our scheme has less communication overhead and is more scalable. Furthermore, it is capable of eliminating redundant messages through aggregation. Last but not least, the architecture is attacked resilient and handles the failure of RSUs.

Chapter 3 A Secure and Distributed Architecture for Vehicular Cloud

Marriage between cloud computing and VANETs would help solve many of the needs of drivers, law enforcement agencies, traffic management, etc. In this chapter, we propose a secure and distributed architecture for the vehicular cloud which uses the capabilities of vehicles to provide various services such as parking management, accident alert, traffic updates, cooperative driving, etc. Our architecture ensures the privacy of vehicles and supports scalable and secure message dissemination using the vehicular infrastructure.

3.1 Introduction

Modern vehicles are equipped with computing, communication, and storage resources, which often remain underutilized. Vehicular networks can also benefit from cloud computing. The *Vehicular Cloud* (VC) architecture which combines VANETs with cloud, was first proposed in [77] to fully capitalize the resources in VANET. In their approach, a VC is a collection of autonomous vehicles in VANET where vehicles contribute their underutilized computing, sensing, and communication resources to the cloud. Vehicle resources and the information shared by the vehicles with the cloud can be used in decision-making [37]. A vehicular cloud allows vehicles to exchange their collected data with the cloud where it can be analyzed, verified, organized, aggregated, and then propagated to the relevant vehicles/customers. Various other applications can also benefit from using a VC. Some of these applications include accident alerts, parking management, road conditions alerts, cooperative driving, and traffic management.

In many of the existing vehicular cloud architectures, vehicles either communicate with each other directly or use Road Side Units (RSUs) to form a vehicular cloud. In some schemes such as [47, 56], vehicles authenticate themselves and some other schemes use *RSU*-aided message authentication [115, 61]. These schemes suffer from communication overhead and do not scale well. If network traffic becomes heavy, it might not be possible for vehicles to authenticate, process, and forward messages in a timely manner. This could also result in message loss and redundant message propagation.

Ensuring security and privacy is an important issue in the vehicular cloud; if information exchanged between entities is modified by a malicious vehicle, serious consequences such as traffic congestion and accidents can occur. In addition, sensitive data could be lost, and human lives also could be in danger. Hence, messages sent by vehicles must be authenticated and securely delivered to vehicles in the appropriate regions. Furthermore, privacy-related information such as the driver's name, position, and traveling route must be preserved. If vehicles cannot communicate anonymously, an attacker could easily trace vehicles by monitoring the messages sent by that vehicle. Several schemes [47, 61, 56, 13, 58] exist in the literature for solving authentication

and privacy issues in communication; however, many of them do not guarantee the confidentiality of exchanged messages and therefore, are vulnerable to attacks.

In our proposed architecture, vehicles collect data and forward it to the cloud where this data can be verified, analyzed, organized, aggregated, and then propagated to the relevant vehicles. Multiple vehicles may observe the same phenomena and forward these to the cloud which could result in the propagation of redundant messages. However, if information about observed phenomena is aggregated and stored in a cloud, this redundant propagation of messages can be prevented. Hence, data aggregation is another crucial requirement for building an efficient vehicular cloud. Many of the existing schemes do not address this issue and therefore, suffer from computation and communication overhead.

Objectives. The purpose of our work is to design a distributed, secure vehicular cloud architecture that ensures security and privacy in communication. Our scheme should also be scalable and should have less communication overhead. Furthermore, it should be capable of eliminating redundant messages through aggregation. Last but not the least, the architecture is attacked resilient and can handle the failure of RSUs.

Contributions. Following are the main contributions in this direction:

- *Confidentiality.* We propose a *Vehicular Cloud* architecture that ensures the confidentiality of sensitive messages by encrypting messages using *Symmetric Key Cryptography.*
- *Authentication.* We use digital signature based on *Public Key Cryptography* to ensure the authenticity and integrity of messages.
- *Aggregation.* Our architecture supports data aggregation based on the type and location of the message to eliminate redundant messages.
- *Scalability.* In our scheme, vehicles do not exchange messages between themselves; they only forward them to the nearest *Road Side Units*, which in turn propagate the messages further in a hierarchical manner. This reduces communication overhead and makes the architecture scalable.
- *Privacy.* Our scheme ensures the anonymity and privacy of vehicles using pseudo IDs. We propose an RSU-mix zone model where vehicles can change their pseudo IDs to improve their privacy.

The rest of the chapter is organized as follows. In Section 3.2, we describe our proposed architecture. In Section 3.4, we present the security and overhead analysis of the architecture. We present and compare some related works in Section 3.5. Finally, Section 3.6 summarizes the chapter.

3.2 Proposed Model

In this section, we present our system model and then describe the proposed architecture in detail.

3.2.1 System Model

Fig.1 illustrates the proposed architecture which consists of vehicles, Road Side Units (RSUs), Regional Clouds (RC), and a Central Cloud (CC).

Vehicle: Vehicles are assumed to be equipped with an on-board unit (OBU) for computation and communication. Vehicles can communicate with RSUs through the radio defined under the IEEE Standard 1609.2 [97], which is the proposed standard for wireless access in vehicular environments (WAVE). We assume that vehicles obtain their public/private key pairs and the public keys of the RSUs and a set of pseudonyms when they register with their local RC such as the Department of Motor Vehicles (DMV) that administers vehicle registration and driver licensing.

RSU: RSUs are distributed on the roadsides. In our architecture, RSUs are assumed to be not compromised. They collect the information sent by vehicles as well as authenticate and aggregate the received messages and forward them to the regional cloud.

RC: We assume the geographical area (for example, a country) is divided into regions and each region is controlled by an RC. An RC store, analyzes, processes, and aggregates the relevant messages received from RSUs in its region and forwards the information to CC if necessary. It manages all private information about vehicles in its region and shares them securely with RSUs upon request. The RC and the RSUs within its region are able to communicate with each other through a wired or wireless network. When a vehicle sends a message, the RSU can verify the authenticity of the message and the RC can also help RSUs to identify the real identity of vehicles when investigations are required. RCs are assumed to be trustworthy and not compromised and have a large computation and storage capacity. RCs are assumed to be connected to all RSUs in its region, possibly through the Internet.

CC: The CC is assumed to have more storage and computational power than the RCs. CC and RCs can communicate with each other securely via a wired or wireless network. RCs provide the CC with services that may be needed by other vehicles in other regions. In addition, CC can provide services to other departments such as law enforcement, traffic management, etc. The CC is assumed to be trustworthy and not compromised.

3.2.2 Proposed Architecture

In this section, we describe our architecture in detail. The notations used in this section are listed in Table 3.1.

In our scheme, information collected by vehicles in an area is sent to the RC which covers that area (e.g., city or state) through the nearby RSUs which authenticate and aggregate the received messages and then forward them to the RC for storage. RC analyzes, processes, and further aggregates the relevant messages and sends them to vehicles in appropriate regions so the drivers can take appropriate action. The RCs also communicate securely via wired/wireless networks with the CC which has more storage and computational power. The CC provides services to the RCs and to different departments (e.g., police department, traffic department, health department,

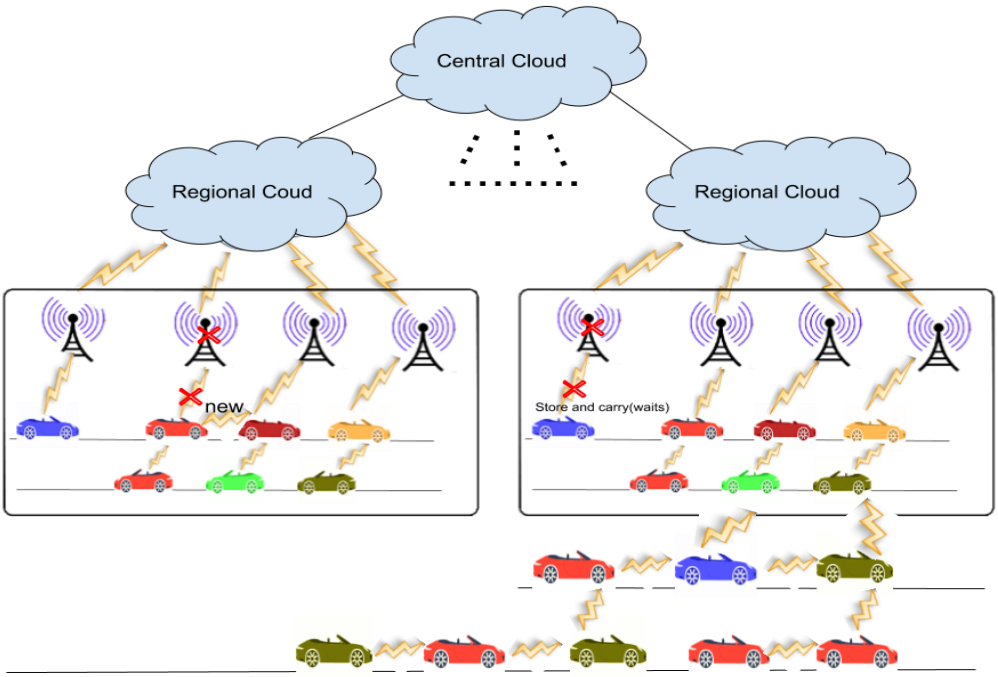

Figure 3.1: Secure and Distributed Architecture for Vehicular Cloud.

Table 3.1: Notations.

Notation	**D**escription
RC_i	Regional Cloud i
RSU_i	Road Side Unit i
CC	Central Cloud
ID_A	Identity of Entity A
PID_A	Pseudo Identity of Entity A
M	A Message
V_x	Vehicle x
$Type$	Type of Message
Loc	Location of the Phenomena
ts	Timestamp
SK_A	Private Key of Entity A
PK_A	Public Key of Entity A
K	Secret Shared Key
$SIG_A(M)$	Signature of M Signed using A's Private Key
$H()$	Hash Function
$E(M, K)$	Encryption of M with Key K

etc.) and these departments can also provide the CC with services that the drivers might need. For example, if someone's car gets stolen, he/she will call the police department, the police will notify the CC which will forward it to the RCs and the RCs will forward it to the vehicles in their region through $RSUs$. When a vehicle sees the stolen car using a camera that captures the plate number of the stolen car, it will send a message that contains the location of the stolen car back to the RC which will send to the police department.

Fixed infrastructure (e.g., $RSUs$) may not exist in some areas or nearby RSU could have failed. In such cases, messages have to be routed to another nearby RSU through intermediate vehicles. If there are no vehicles or RSUs within the transmission range of a vehicle, the vehicle stores and carries the message until it gets closer to the next RSU or a vehicle. $RSUs$ are responsible for verifying the authenticity and integrity of messages sent by vehicles before forwarding them to RC. In addition, all driver information should be protected and attackers should not be able to trace the routes of the vehicles. We propose an RSU-mix zone model where vehicles can change their pseudo IDs and use them in all communications instead of their real identities and changing them frequently will improve privacy.

3.2.3 Key Generation and Distribution

We assume that the $RSUs$, RCs, and the CC are trusted and not compromised. When a vehicle v is registered or renewed with RC, it gets the public keys of the RSUs and stores them in the vehicles and each vehicle is preloaded with a set of pseudonyms $(PID_1, PID_2, ..., PID_n)$.

In our architecture, all messages are authenticated using digital signatures. When a vehicle sends a message, the sender vehicle attaches its digital signature to the message. The digital signature is made by encrypting the hash of the message using the vehicle's private key. Moreover, not all messages need to be encrypted; a vehicle can decide if the message needs to be encrypted or not depending on the type of the message. For example, if a vehicle has to notify about ice on the road, it need not encrypt the message. On the other hand, if a vehicle wants to send a message about a crime scene, this message needs to be encrypted. If the vehicle decides to encrypt a message, it generates a secret key K and encrypts the message using K and this key also is encrypted using the public key of RSU PK_{RSU} and sent to the nearby RSU. This ensures confidentiality. Now, when RSU receives the message, it gets the secret key K first by decrypting it using $RSU's$ private key SK_{RSU} and then it decrypts the message using the secret key K and authenticates the message using the signature.

The following two subsections describe how vehicles send messages about observed phenomena and how they request a service from the RC.

3.2.4 Vehicles Sending Messages about Observed Phenomena

We assume events are classified into various types such as traffic congestion, ice on the road, accidents, etc. When vehicles sense events, they send messages about the sensed events to the RC through $RSUs$ and the RC determines if the message needs to be

forwarded to all the vehicles in its region. At the same time, the RC also forwards the messages to the CC, if necessary. The CC processes and stores the received messages and forwards the messages to other RCs if they need them. For example, if a vehicle observes abnormal road conditions (e.g., work zone), it will send a message to RC which will send it to the vehicles in its region and to the CC which will forward it to the relevant RCs. So the vehicles that are going toward that work zone area can avoid the area. So, the driver can have information about traffic conditions which can increase driver safety and reduce the number of traffic accidents. Figure 3.2 shows the flow chart of communication from vehicle to RC. This scheme works as follows.

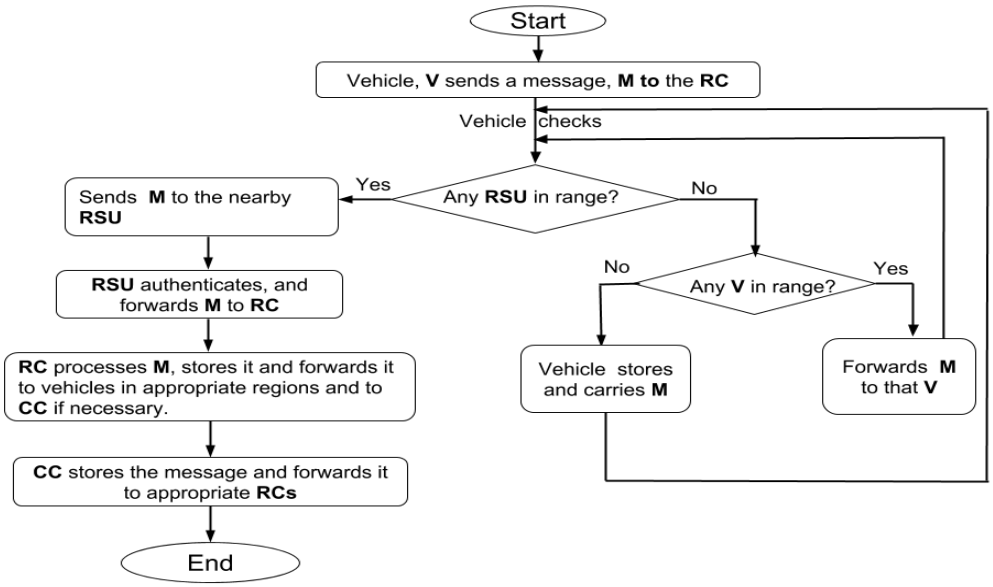

Figure 3.2: Vehicle Sending a Message to RC.

When a vehicle v_x senses an event, it decides whether it is sensitive information or not. If it is not sensitive, v_x assembles the message M_1 without encrypting it, and then sends it to the nearby RSU_k. If it is sensitive information, it assembles M_2 and then sends it to the nearby RSU_k, where M_1 and M_2 are defined as follows:

$M_1 = ID_{RSU_k}, (PID_{v_x}, Type, Loc, ts), SIG_{v_x}(M_1)$
$(where, SIG_{v_x}(M_1) = E(H(PID_{v_x}, Type, Loc, ts), SK_{v_x}).$

$M_2 = ID_{RSU_k}, E((PID_{v_x}, Type, Loc, ts), K), E(K, PK_{RSU_k}), SIG_{v_x}(M_2)$
$(where,\ SIG_{v_x}(M_2) = E(H(PID_{v_x}, Type, Loc, ts), SK_{v_x}).$

The message M_1 includes the pseudo ID of vehicle v_x, the type of the message $Type$, the location of phenomena Loc, and the timestamp ts. v_x attaches its digital signature that is obtained by computing the hash of the message and encrypting it using its private key SK_{v_x}. In message M_2, the symmetric key K, generated by v_x is used to encrypt the message and this key is encrypted with the public key of RSU_k. When the nearby RSU_k receives the message, it verifies the authenticity and integrity of the message using the signature and processes the message. In case, no nearby RSU

exists or nearby RSU becomes unavailable due to failure, our architecture handles this situation as follows.

Case 1. When a vehicle finds that the RSU within its transmission range has failed (or there is no RSU within its transmission range), it computes the message using the ID and the public key of the nearest RSU and then forwards the message to that RSU through other vehicles using an underlying routing algorithm. We assume that all vehicles know the location as well as the public keys of all RSUs. For example, in AODV protocol [73], a node requests a route to a destination by broadcasting a route request (RREQ) message to all its neighbors. When a node receives an RREQ message, it, in turn, broadcasts the RREQ message. This process repeats until the RREQ reaches a destination (RSU). Then, the destination responds with a route reply (RREP) message. Therefore, if a vehicle doesn't get a reply message from RSU, the route to that RSU failed. Next, the vehicle will look for another RSU and go through the same process until it gets an RREP message. Once the vehicle gets an RREP message, it will forward the sensed message to that RSU.

Case 2. If there is no nearby RSU and no nearby vehicles, v_x waits (stores and carries the message) until it finds an RSU or a vehicle within its transmission range. If v_x finds any of them it forwards the message to that RSU or to that vehicle.

Message Aggregation by *RSU*s and *RC*. When an RSU receives multiple messages from different vehicles that describe the same event, it aggregates them (based on *type*, *location*, and *time*) to a single message M_{agg} and sends it to the RC. When RC receives the message M_{agg}, RC stores M_{agg} and performs more aggregation if it gets similar messages from other RSUs. M_{agg} is defined as follows:

$$M_{agg} = ID_{RC_i}, (ID_{RSU_i}, M, ts), SIG_{RSU_i}(M_{agg})$$
$$(where\ SIG_{RSU_i}(M_{agg}) = E(H(ID_{RSU_i}, M, ts), SK_{RSU_i}).$$

3.2.5 *RC* Sending Messages to *CC* or to *RSUs*

When an RC decides to send a message to the RSUs in the area covered by its region (e.g., city or state), it computes and disseminates M_3 to the appropriate RSUs and the RSUs broadcast the message to the vehicles within their transmission ranges. In addition, RC decides whether or not to send M_3 to CC. If RC thinks that M_3 is an important message, then it sends M_3 to CC as well. M_3 is given by,

$$M_3 = ID_{RC}, (Type, Loc, ts), SIG_{RC}(M_3)$$
$$(where\ SIG_{RC}(M_3) = E(H(Type, Loc, ts), SK_{RC}).$$

Upon receiving the message M_3, CC stores and processes M_3. Then it computes M_4 which could be encrypted if necessary and then disseminates M_4 to all appropriate RCs if they need them. When RC_i receives the message M_4, it broadcasts M_4 to all vehicles within its region through the RSUs and the intended vehicles then consume it. M_4 is given by,

$M_4 = ID_{RC_i}, ID_{CC}, (Type, Loc, ts), SIG_{CC}(M_4)$
$(where \;\; SIG_{CC}(M_4) = E(H(Type, Loc, ts), SK_{CC}).$

Algorithm 1 shows the algorithm for vehicle sending message to RC.

Algorithm 1 Vehicle v_i Sending Messages to RC

```
When v_i wants to send a message:
  if there is an RSU within its transmission range // step 1.
      Send(M) to RSU;
  else if there is a vehicle v_j within its transmission range
      Send (M) to v_j;
  else
      Store (M) and go to step 1.

When RSU_i receives M from v_i:
      Authenticate (M);
      Aggregate (M) with other messages if possible;
      Send(M) to RC;

When an RC receives M from an RSU:
      Authenticate(M);
      Aggregate(M) with other messages if possible;
      Process(M);
      Store(M);
      Send(M) to CC if necessary;
      Send(M) to other RSUs in its region if necessary;

When CC receives the message M:
      Authenticate(M);
      Aggregate(M) with other messages if possible;
      Process(M);
      Store(M);
      Send(M) to appropriate RCs;
```

3.2.6 Vehicles Requesting Service from an RC

Figure 3.3 gives a flow chart of actions taken when a vehicle requests a service from its RC. Next, we present the algorithm for requesting service in detail.

When vehicle v_x wants to request a service from the RC, it computes and sends the message M_1 to the nearby RSU which then forwards M_2 to the RC where M_1

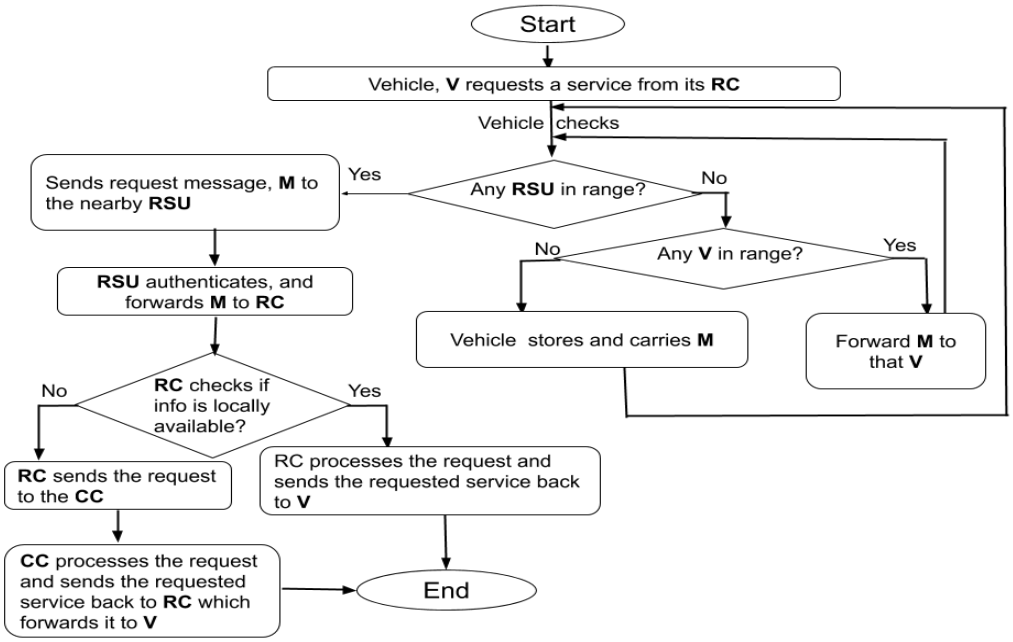

Figure 3.3: Vehicle Requesting a Service from RSU.

and M_2 are given by,

$M_1 = ID_{RSU}, (PID_{v_x}, Type, Loc, ts), SIG_{v_x}(M_1)$
$(where \;\; SIG_{v_x}(M_1) = E(H(PID_{v_x}, Type, Loc, ts), SK_{v_x}).$
$M_2 = ID_{RC}, (ID_{RSU}, Type, Loc, ts), SIG_{RSU}(M_2)$
$(where\ SIG_{RSU}(M_2) = E(H(ID_{RSU}, Type, Loc, ts), SK_{RSU}).$

The message M_1 includes the pseudo ID of vehicle PID_{v_x}, the type of event $Type$, the location of event Loc, and the timestamp ts. The vehicle v_x attaches its digital signature SIG_{v_x} so that the RSU can authenticate the request message. If the vehicle v_x wants to request sensitive information (e.g., a vehicle from a police department), it generates a symmetric key K and encrypts the message using K and then encrypts K with RSU's public key PK_{RSU}. In this case v_x sends M_3 where M_3 is defined as follows:

$M_3 = ID_{RSU}, E((PID_{v_x}, Type, Loc, ts), K), E(K, PK_{RSU}), SIG_{v_x}(M_3)$
$(where\ SIG_{v_x}(M_3) = E(H(PID_{v_x}, Type, Loc, ts), SK_{v_x}).$

If the nearby RSU is not within the transmission range of v_x, it will forward the request message to the nearby RSU through intermediate vehicles using an underlying routing protocol.

When an RC receives the request message, it checks if it has the requested information. If so, it computes and sends a service message M_4 to the v_x through RSU. Note that if the vehicle is not within the transmission range of RSU, M_4 will be forwarded through intermediate vehicles, where,

$M_4 = ID_{RSU}, (ID_{RC}, Type, Loc, ts), SIG_{RC}(M_4)$
$(where\ SIG_{RC}(M_4) = E(H(ID_{RC}, Type, Loc, ts), SK_{RC})$.

If RC does not have the requested information, it forwards the request to the CC. When CC receives the request, it checks if it has the requested information. If it has it, CC computes and sends the service message to the RC. When RC gets the service message, it forwards the service message to the relevant RSU which forwards it to the vehicle v_x directly if v_x is within $RSU's$ transmission range or through intermediate vehicles if it is not. Then v_x authenticates the message and consumes it. Algorithm 2 gives the algorithm for requesting information.

3.2.7 When vehicle v moves to new RC

When a vehicle enters the area covered by a new regional cloud, it sends a join message (M_j) to the nearby RSU.
$M_j = ID_{RSU_k}, (PID_{v_x}, Join, ts), SIG_{v_x}(M_j)$
$(where, SIG_{v_x}(M_j) = E(H(PID_{v_x}, join, ts), SK_{v_x})$.

The message M_j includes the pseudo ID PID_{v_x} of the vehicle, the join request message $Join$, and the timestamp ts. We use the timestamp to ensure the freshness of the join message and the vehicle v_x attaches its digital signature SIG_{v_x}.

After receiving the join message of the vehicle, RSU sends it to the regional cloud. The regional cloud verifies the validity of the vehicle which has information about all the vehicles in its region. If it is a valid vehicle, RC sends $PIDs_{v_x}$ and PK_{v_x} to all RSUs within its region. Then RSUs start receiving and authenticating messages from vehicle v_x. In addition, if vehicles travel from one RC to other RCs, their information will be kept in RSUs. In case vehicles travel rarely in other RCs, their information will not be kept in RSUs for a long time. For example, if a vehicle doesn't visit an RC thirty days after its first visit, that RC will inform the RSUs in its region to remove all information pertaining to that vehicle.

3.3 Achieving Privacy in Our Architecture

Vehicles in VANETs use pseudonyms (pseudo IDs) to communicate with each other and RSUs to ensure privacy. Researchers recommend changing the pseudonyms frequently so that messages cannot be linked. Changing pseudonyms will not be useful if previous and current pseudonyms can be linked to a vehicle. Therefore, different schemes have been proposed to hide the pseudonym changes to make it difficult to link pseudonyms. One of the strategies to change pseudonyms is to establish mix zones [39, 59, 25, 96, 78, 35, 42, 112], where several vehicles in an area change pseudonyms at the same time to confuse the attacker from linking old and new pseudonyms. Most of these schemes do not fully guarantee privacy when changing pseudonyms in some situations (e.g., vehicles density is low). Hence, we design a scheme for changing pseudonyms that ensures privacy in all traffic conditions. Our scheme depends on mix zones, which are regions that allow vehicles to change their pseudonyms to make

Algorithm 2 Vehicle v_i Requesting Information from RC

When v_i sends a request message M to RC:
  if there is an RSU within its transmission range
    Send(M) to RSU;
  else if there is a vehicle v_j within the transmission range of v_x
    Send (M) to v_j;
  else
    Store (M) and go to the first step.

When an RSU_i receives M from v_x:
    Authenticate (M);
    Send(M) to RC;

When an RC receives M from an RSU:
    Authenticate(M);
    if RC has the requested information
      Send(SM) to v_x through RSU;
    else
      Send (M) to CC

When CC receives M from RC_k:
    Authenticate(M);
    if CC finds the requested information
      Send(SM) to RC_k; // SM is the *Service Message*
    else
      Send (NSM) to RC_k; // NSM is *No Service Message*

When RC_k receives SM or NSM from CC:
    if the message is SM
      Authenticate(SM);
      Store(SM);
      Send(SM) to the requested v_x through RSU;
    else
      Authenticate(NSM);
      Send (NSM) to the requested v_x through RSU;

it difficult for an attacker to link pseudonyms to a vehicle.

3.3.1 Proposed model for Privacy

In this section, we present our scheme for changing pseudonyms in a mix zone environment.

In our scheme, we assume that each vehicle has a set of pseudonyms generated by the RC, and the vehicles change their pseudonyms in a predefined RSU region. As an example, when a vehicle v_i runs out of pseudonyms, v_i sends a request message M_p to RC for getting a new set of pseudonyms. The message M_p is defined as follows:

$M_p = ID_{RSU_k}, E((PID_{v_i}, type, ts), K), E(K, PK_{RSU_k}), SIG_{v_i}(M_p)$
$(where,\ SIG_{v_i}(M_p) = E(H(PID_{v_i}, type, ts), SK_{v_i})$.

The message M_p includes the pseudo ID of vehicle v_i, the type of message *type*, and the timestamp ts. The symmetric key K, generated by v_i is used to encrypt the message and this key is further encrypted with the public key of RSU_k. Finally, v_i attaches its digital signature that is obtained by computing the hash of the message and encrypting it using its private key SK_{v_i}. When the nearby RSU_k receives the message, it verifies the authenticity and integrity of the message and forwards it to RC. After receiving the request message, RC generates a set of pseudonyms for the v_i and sends the message M'_{p_1} to all the $RSUs$ in its region. The message M'_{p_1} is defined as follows:

$M'_{p_1} = ID_{RSU_i}, E((ID_{RC}, (PID_1, ..., PID_n), type, ts), K), E(K, PK_{RSU_i}), SIG_{RC}(M'_{p_1})$
$(where,\ SIG_{RC}(M'_{p_1}) = E(H(ID_{RC}, PID_1, ..., PID_n, type, ts), SK_{RC})$ and $i = 1, 2, ..., m$
(assuming there are m $RSUs$).

The message M'_{p_1} includes the ID of the RC ID_{RC}, a set of pseudonyms $PID_1, ..., PID_n$ assigned for the vehicle v_i, the type of message *type*, and the timestamp ts. The symmetric key K, generated by the RC is used to encrypt the message and this key is further encrypted with the public key of RSU_i (where $i = 1, 2, ..., m$). Finally, RC attaches its digital signature that is obtained by computing the hash of the message and encrypting it using its private key SK_{RC}. When the nearby RSU_i receives this message, it verifies the authenticity and integrity of the message and forwards it to the v_i that requested the pseudonyms.

The message forwarded to the v_i is M'_{p_2} and defined as follows:

$M'_{p_2} = PID_{v_i}, E((ID_{RSU_i}, M'_{p_1}, type, ts, K), E(K, PK_{v_i}), SIG_{RSU_i}(M'_{p_2})$
$(where,\ SIG_{RSU_i}(M'_{p_2}) = E(H(ID_{RSU_i}, M'_{p_1}, type, ts), SK_{RSU_i})$ and $i = 1, 2, ..., m$
(assuming there are m $RSUs$).

Algorithm 3 gives the algorithm to get a new set of pseudonyms when a vehicle runs out of pseudonyms.

Algorithm 3 Getting New Set of Pseudonyms

When vehicle v_i runs out of pseudonyms:

v_i sends a request message M_p to RC to get a new set of pseudonyms;

When an RSU_i receives M_p from v_i:

`Authenticate` (M_p);
`Send(`M_p`)` to RC;

When an RC receives M_p from an RSU:

`authenticate(`M_p`)`;
`compute(`M'_{p_1}`)`;
// M'_{p_1} contains a list of pseudonyms $(PID_1, ..., PID_n)$ for v_i
`send` (M'_{p_1}) to all the $RSUs$;

When an RSU_i receives M'_{p_1} from RC:

`authenticate(`M'_{p_1}`)`;
`compute(`M'_{p_2}`)`;
`send(`M'_{p_2}`)` to the v_i that requested for pseudonyms;
v_i `authenticate(`M'_{p_2}`)`;
v_i and RSUs store pseudonyms;
v_i starts using a new pseudonym from that list;

In our model, vehicles can change pseudonyms even if the density of vehicles is low. This will be done with the help of RSU. In case there is a low density of vehicles in the RSU mix zone, the RSU will send a message to some vehicles and tell them to act as k vehicles to increase their density and make it hard for the attacker to link pseudonyms to them. In this case, the attacker will think that there are actually k vehicles participating in pseudonyms changes. Furthermore, the pseudonym can be traced to its identity with the collaboration of RSU and RC if a vehicle misbehaves. Figure. 3.4 shows an RSU mix zone where vehicles change their pseudonyms.

3.3.2 RSU-Mix Zone Establishment

Beacon messages are periodically broadcasted by vehicles. These messages contain the location, speed, and other information of vehicles so that RSUs can determine the number of vehicles within their transmission range. When it is time for vehicles to change pseudonyms if there are a sufficient number of vehicles in the mixed zone (Figure 3.4-a), vehicles will change their pseudonyms.

If the number of vehicles (x) is less than the predefined threshold (k) (Figure 3.4-b), RSU sends a secure message M''_p to all vehicles in the mixed zone to act as more than one vehicle. The message M''_p is defined as follows:

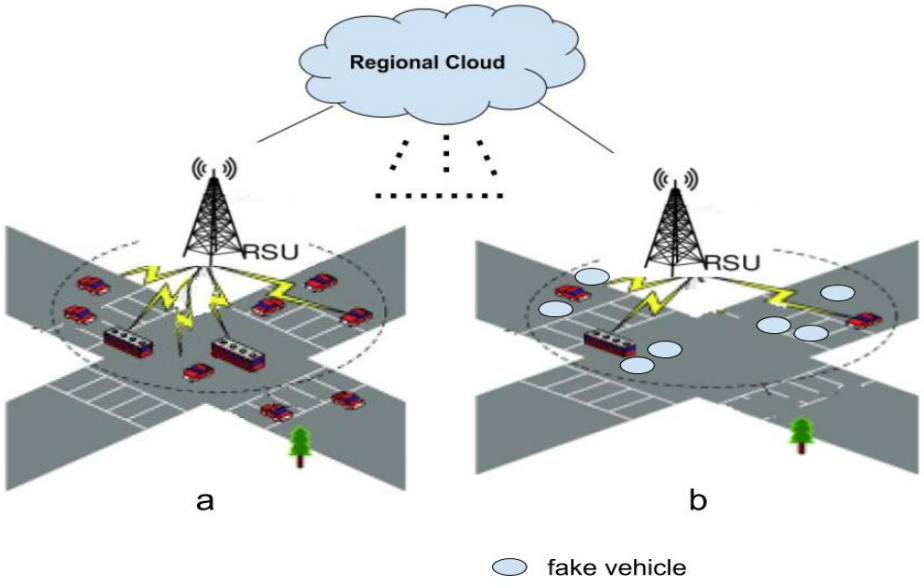

Figure 3.4: Vehicles Changing Pseudonyms in RSU Mix Zone.

$M_p^{''} = PID_{v_i}, E((ID_{RSU_k}, type, \lceil k/x \rceil, ts), K), E(K, PK_{v_i}), SIG_{RSU_k}(M_p^{''})$
$(where,\ SIG_{RSU_k}(M_p^{''}) = E(H(ID_{RSU_k}, type, \lceil k/x \rceil, ts), SK_{RSU_k})$. and $i = 1, 2, ..., n$ (assuming there are n vehicles).

The message $M_p^{''}$ includes PID of the vehicle v_i (where $i = 1, 2, ..., n$), ID of the RSU_k, the type of message $type$, number of pseudonyms ($\lceil k/x \rceil$) each vehicle v_i will use, and the timestamp ts. The symmetric key K, generated by RSU_k is used to encrypt the message and this key is further encrypted with the public key of v_i. Finally, RSU_k attaches its digital signature that is obtained by computing the hash of the message and encrypting it using it's private key SK_{RSU_k}.

As mentioned above, each vehicle v_i will receive a copy of the message $M_p^{''}$ from the RSU which is in charge of the mix zone. Now, each v_i inside the RSU-mix zone will verify the authenticity and integrity of the message and then generates $\lceil k/x \rceil$ messages for changing pseudonyms. As a result, a mix zone is established with at least k indistinguishable new pseudonym-changing messages. Therefore, the attacker will have less chance of linking messages with different pseudonyms to a vehicle. For example, if the predefined threshold on the number of vehicles in a mix zone is 10 (k = 10) and there are 3 vehicles ($x = 3$) present there, each of the vehicles will send 4 ($\lceil 10/3 \rceil$) messages to reach the threshold. Following are the four messages each v_i will send to the RSU in the RSU-mix zone to change their pseudonyms.

$M_{p_1}^{''} = ID_{RSU_k}, (PID1_{v_i}, type, ts), SIG_{v_i}(M_{p_1}^{''})$

$(where,\ SIG_{v_i}(M''_{p_1}) = E(H(PID1_{v_i}, type, ts), SK_{v_i}).$

$M''_{p_2} = ID_{RSU_k}, (PID2_{v_i}, type, ts), SIG_{v_i}(M''_{p_2})$
$(where,\ SIG_{v_i}(M''_{p_2}) = E(H(PID2_{v_i}, type, ts), SK_{v_i}).$

$M''_{p_3} = ID_{RSU_k}, (PID3_{v_i}, type, ts), SIG_{v_i}(M''_{p_3})$
$(where,\ SIG_{v_i}(M''_{p_3}) = E(H(PID3_{v_i}, type, ts), SK_{v_i}).$

$M''_{p_4} = ID_{RSU_k}, (PID4_{v_i}, type, ts), SIG_{v_i}(M''_{p_4})$
$(where,\ SIG_{v_i}(M''_{p_4}) = E(H(PID4_{v_i}, type, ts), SK_{v_i}).$

When these three physical vehicles send twelve messages using twelve different pseudonyms to the RSU, inside the mix zone, the attacker will be deceived into thinking that there are twelve vehicles sending messages at any given time. If we look at the messages above, we see that all four of them are sent in plaintext. However, the signatures generated by each of them are different since each of them uses a different pseudonym. Thus, the attacker will have less chance of linking these messages with different pseudonyms to a vehicle. We present our pseudonym-changing algorithm in algorithm 4.

Algorithm 4 Changing Pseudonyms

When vehicles attempt to change their pseudonyms within an RSU-mix zone:
RSU_i `determines the number of vehicles (`x`) in the mix-zone:`

`if` x `> predetermined threshold (`k`)`
vehicles automatically change their current set of pseudonyms to a new set of pseudonyms without involving the RSU
`else`
RSU_i `generates` a message M''_p and `broadcasts` it to all the vehicles within its transmission range;
// M''_p contains no. of pseudonyms each vehicle will use
Each vehicle v_i `authenticates (`M''_p`)`;
Each v_i `produces` $\lceil k/x \rceil$ messages and sends them to the RSU_i in the mix zone;
// This results in $x * \lceil k/x \rceil$ indistinguishable new pseudonyms in the region

3.4 Security and Overhead Analysis

3.4.1 Ensuring Security and Privacy

Confidentiality. All the sensitive messages sent by the vehicles, RSUs, RCs, and CC are encrypted using a symmetric key which has low overhead.

Message Authentication and Non-repudiation. We used digital signatures based on public key cryptography to ensure the authenticity and integrity of messages. In our architecture, a digital signature is attached to every message, whether it is encrypted or not. When the receiver receives the message, the receiver authenticates the message based on the digital signature and processes it. Since no one can forge the digital signature of the sender because no one knows his/her private key, the authenticity of the message and non-repudiation are guaranteed.

Privacy Preservation. Vehicles are assigned pseudo IDs. A vehicle never uses its real ID in any communication. So, the privacy of vehicles is preserved. In order to maximize privacy, we also proposed a new pseudonym-changing strategy. When a malicious node is detected, the real ID of the malicious vehicle can be revealed by the RC to the authorities for legal investigation.

3.4.2 Attack Resilience

Man-in-the-middle Attack. In this attack, the adversary intercepts and alters the messages transferred between two vehicles or between a vehicle and an RSU. In our scheme, a sender attaches the digital signature $SIG(M) = E(H(M), SK)$ to the message. A signature cannot be forged without knowing the sender's private key. Thus, our scheme prevents man-in-the-middle attack.

Replay Attack. In this attack, the attacker resends or delays a previously transmitted message. To detect such attacks, all messages carry the time stamp ts. We assume clocks are loosely synchronized. Vehicles can use Global Positioning System (GPS) for synchronizing clocks.

Message Modification Attack. In order to protect the integrity of the messages from attackers, each message carries the signature of the sender. Moreover, it also guarantees unforgeability under chosen message attacks– adversary who knows the public key, and gets to see signatures on messages of his/her choice, cannot modify the message and produce a verifiable signature on the modified message.

3.4.3 Communication and Computation Overhead

Our architecture is scalable since vehicles do not store any keys of other vehicles and do not authenticate any message sent by other vehicles. The RSUs which have more storage and computational power than vehicles are responsible for authenticating, aggregating, and forwarding messages. Vehicles only need to verify messages received from RSUs, not from other vehicles; so communication and computation overhead is low for vehicles. In addition, not all messages need to be encrypted in our architecture, just the sensitive messages need to be encrypted. If any vehicle decides to encrypt a message, it generates a symmetric key to encrypt the message. Vehicles do not have

to request a symmetric key from RSUs, which further reduces the communication and computation overhead.

We analyzed the overhead involved using a Toshiba computer running windows 8.1 operating system for vehicle side. The machine is equipped with an Intel i3 quad-core processor, a 2.50 GHZ clock frequency, and 6 gigabytes of memory. For the RSU side, we used Asus computer running windows 10 operating system which is equipped with an Intel i7, 2.21 GHz processor, and 12 gigabytes of memory. The public key authentication and encryption scheme is based on RSA (Rivest–Shamir–Adleman). Following are some notations used for presenting our results: time for computing signature (T_{sign}), time for signature verification (T_{verify}), time for encrypting the symmetric key using a public key (T_{Epk}), time for decrypting the symmetric key using a private key (T_{Dsk}), time for encrypting the message using a symmetric key (T_{Ek}), and time for decrypting the message using a symmetric key (T_{Dk}). We used AES (Advanced Encryption Standard) to encrypt and decrypt the messages. The execution time of the above operations is listed in Table 3.2.

Table 3.2: Execution time for different operations (Milliseconds).

Operation	**V**ehicle side	**R**SU side
T_{sign}	0.06	0.01
T_{verify}	0.005	0.0008
T_{Epk}	0.0046	0.0009
T_{Dsk}	0.02	0.002
T_{Ek}	1.166	0.41
T_{Dk}	2.128	1.02

Computation Overhead when messages are sent from vehicles to RC: We focus on the computation overhead of the RSU because it does most of the work such as forwarding messages for vehicle members and to the RC, encrypting and decrypting messages, and authenticating vehicles in its region. Figure 3.5 shows computation overhead due to encryption-decryption of messages. It also shows overhead due to creating and verifying signatures at the RSU side and the vehicle side when the messages are sent from vehicles to the RC. The relationship between the computation overhead and the number of messages is plotted where the number of messages varies from 10 to 100.

Computation Overhead when messages are sent from RC to vehicles: In Figure 3.6, we present a comparison of the computation overhead at the RSU side and the vehicle side when messages are sent from the RC to vehicles. In this case, the RSU (as depicted in our proposed scheme) only transmits messages to vehicles in its region. The message verification and decryption operations are performed only at the vehicle side (no such operations are performed at the RSU side); hence, no computation overhead at the RSU side. The proposed scheme also does not require much computation time to decrypt and verify messages on the vehicle side.

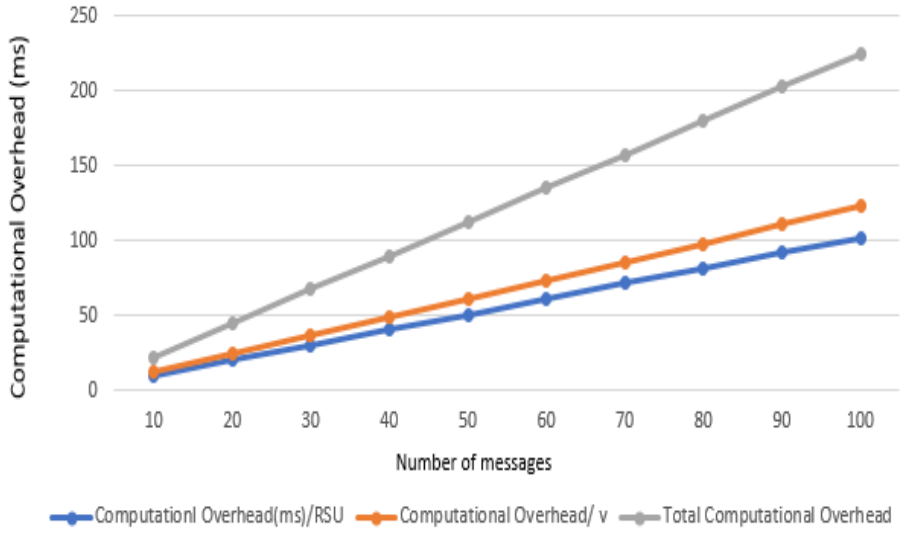

Figure 3.5: Computation time at the RSU side and at the vehicle side.

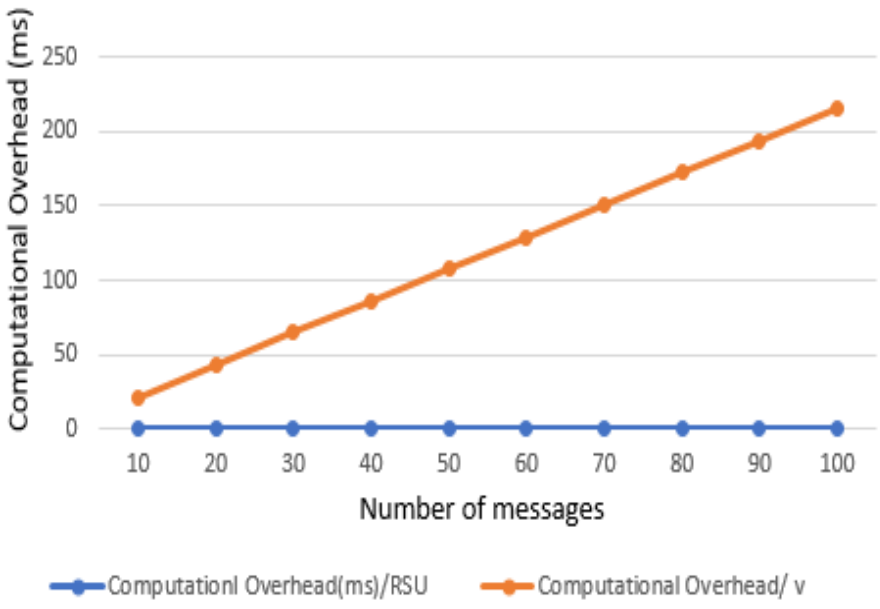

Figure 3.6: Computation time at the RSU side and at the vehicle side.

Computation Overhead due to pseudonym changes: In our model, if there are a sufficient number of vehicles present at any given time in the RSU-mix zone, RSU does not need to broadcast messages to vehicles. However, in case, there is a low density of vehicles, RSU broadcasts messages to vehicles and instructs them to act as $\lceil k/x \rceil$ vehicles to facilitate the pseudonym-changing process, where k is a predefined threshold and x is the number of vehicles present within RSU-mix zone at that particular time. In this case, the total number of broadcast messages depends on the number of vehicles within the RSU-mix zone, which could be at least one and at most x. On the vehicle side, the total number of broadcast messages is at least one in case there is a sufficient number of vehicles and at most k in case there is only one vehicle within the RSU-mix zone.

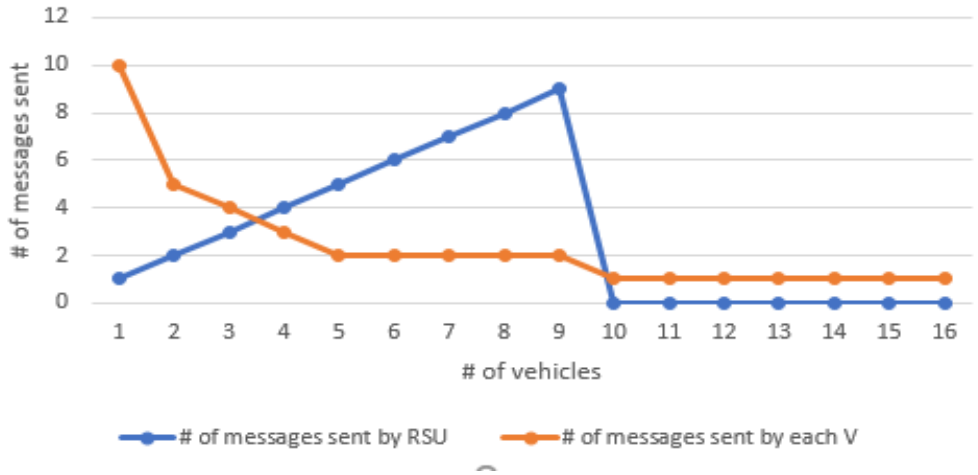

Figure 3.7: Broadcast messages cost.

Figure 3.7 shows the cost of broadcast messages. We assumed the predefined threshold $k = 10$. In Figure 3.7, if the number of vehicles is less than 10, RSU sends messages to all vehicles within its transmission range. In case, there are 10 vehicles or more within the RSU's transmission range, RSU does not need to exchange any messages since there is already a sufficient number of vehicles present in the mix zone to execute the pseudonym-changing process. As a result, in our scheme, the communication cost is lower when there are 10 vehicles or more. As presented in Table 4, the number of messages sent by a vehicle varies based on the number of vehicles within the RSU transmission range. In case there is only one vehicle ($x = 1$) present in the RSU-mix zone, this vehicle will act as 10 ($\lceil 10/1 \rceil$) vehicles and will send a total of 10 ($1 * \lceil 10/1 \rceil$) messages at one instance. Similarly, when there are 3 vehicles ($x = 3$), each of them will send 4 ($\lceil 10/3 \rceil$) messages to reach the threshold. If there are either 5 or 9 vehicles ($x = 5$ or $x = 9$), each vehicle will send 2 messages ($\lceil 10/5 \rceil$ or $\lceil 10/9 \rceil$).

A Comparison Between Our Scheme and the Existing Pseudonym Changing Schemes:

A number of pseudonym-changing schemes have been proposed in the literature. Some schemes work well in a dense traffic environment while others are applicable only in sparse environments (e.g., freeways). In VANET, ensuring location privacy has been identified as one of the key concerns. Various schemes are proposed for preserving privacy but they either incur computational overhead or compromise safety to some extent. In this subsection, we present some pseudonym-changing strategies presented in the literature and compare our scheme with them. We classified the pseudonym-changing schemes into two main categories namely, mix zone schemes and mix content schemes.

Mix Zone Schemes: Freudiger et al. [39] presented the first implementation of mix zone in VANET where road intersections are used as mix zones. In their scheme, they used RSU to execute the pseudonym-changing task. Whenever vehicles are in a mix zone, they exchange safety messages using a shared symmetric key issued by the RSU. Hence, their scheme is also known as the CMIX (Cryptographic Mix Zone) protocol. The tracking of vehicles depends mostly on the traffic density and their delay characteristics. The anonymity of vehicles increases linearly with the number of vehicles in the intersection. This scheme is not suitable for sparse networks and is also not scalable. Moreover, it is vulnerable to internal adversary attacks. To overcome the possible internal adversary attacks in the CMIX protocol, Carianha et al. [26] proposed a status forwarding scheme where the status information in the periodic beacon messages transmitted by a vehicle is delivered only to its neighbors. Like the CMIX protocol, mix zones are managed by RSU in this scheme. When a vehicle is within the transmission range of an RSU, it initiates the key establishment process as defined in the CMIX protocol. However, in this scheme, alongside the shared symmetric key, RSU also issues a private key for each participating vehicle in the mix zone. That private key is used to encrypt the status information. When RSU receives such messages, it decrypts the status information and forwards it only to the neighbors of that vehicle. Since an internal adversary only has access to its neighbors' status information, it is not possible for the adversary to track all other vehicles in the mix zone. This scheme increases location privacy compared to the CMIX protocol, but the forwarding scheme introduces communication overhead. Like CMIX, it can not ensure privacy in a sparse network as anonymity is proportional to the number of vehicles in the mix zone.

Scheuer et al. [91] presented the concept of ProMix Zone (PMZ) by introducing a communication proxy inside the mix zone. The crossroads and highway intersections are considered as mix zones in their scheme. The mix zones basically contain a set of infrastructure units referred to as proxy which is interconnected, have a computational unit, and cover the whole PMZ with wireless transceivers. PMZ allows vehicles to send beacon messages through the proxy while changing their pseudonyms. All these messages are encrypted and signed using asymmetric key cryptography to provide confidentiality and accountability. Similar to the previous two mix zone schemes, the anonymity, and untraceability increase linearly with a growing number of vehicles in the mix zone. However, linking pseudonyms is possible when the traffic density

is too low. The use of asymmetric key cryptography with an X.509 certificate (in PMZ) makes the beacon size 3.5 times larger than the regular beacon size and thus introduces overhead. Boualouache et al. [19] presented another mix zone scheme named S2Si (Silence and Swap at Signalized Intersection). They used the radio silence technique to perform pseudonym changes. The silence protocol in their scheme first creates a silent mix zone and then the swap protocol ensures that vehicles exchange their pseudonyms in the mix zone controlled by an RSU. Unlike other mix zone schemes, the exchange of pseudonyms among vehicles creates confusion in the attackers' minds. However, as the communication stack parameters are not changed, tracking might still be possible. Additionally, the radio silence period may pose some risks as the safety applications get interrupted for a certain period of time due to the changing of pseudonyms. Another mix zone scheme named Vehicular Location Privacy Zone (VLPZ) [21, 22] is proposed where places with high traffic density (e.g., toll booths, gas stations, rest areas, etc.) are used as mix zones to change pseudonyms. There is a moderate probability that some vehicles may not pass through such zones which prevents those vehicles to change pseudonyms. The use of radio silence period may jeopardize communication and safety messages in the network. Moreover, if the traffic density is low this scheme does not perform well.

Mix Content Schemes: Gerlach et al. [40] proposed the first implementation of mix context for VANET. The mix context is defined as the triggering condition for the simultaneous change of pseudonyms between vehicles. Only when mix context is satisfied, vehicles change their pseudonyms irrespective of their physical location which removes the certainty of change at a particular fixed place. Location privacy significantly increases as the number of vehicles increases. This scheme is not suitable for a sparse traffic environment. Pan et al. [79] presented a cooperative pseudonym-changing scheme considering the number of neighbors. In this scheme, a flag is inserted into beacons to indicate if a vehicle is ready to change a pseudonym. If a vehicle finds that at least k neighbors are preparing to change their pseudonyms, or if its neighborhood has at least k vehicles which are preparing to change their pseudonyms, then it changes its pseudonym with its neighbors. Since the neighboring density is considered as the triggering condition, this scheme does not perform well in sparse networks and gives the attacker a chance to link pseudonyms. Boualouache et al. [20] proposed a scheme named Traffic-Aware Pseudonym Changing Strategy (TAPCS). In TAPCS, vehicles continuously monitor road traffic conditions based on traffic congestion detection protocols to find a suitable place where mix zone can be created. Vehicles change pseudonyms (using the radio silence technique) only when traffic congestion is found. However, in a practical scenario, if some of the vehicles do not pass through such congested areas, they do not get any opportunity to change their pseudonyms.

Bouksani et al. [23] proposed a dynamic pseudonym-changing scheme for addressing privacy in VANET. In this scheme, the vehicle first requests a real pseudonym (Rpseud) from a Trusted Authority (TA) which then sends a response containing Rpseud to the vehicle. The vehicle communicates with the RSU and asks it for an

initial pseudonym (Ipseud). Once the verification and validation process between the RSU and the TA is performed, RSU sends the initial pseudonym (Ipseud) to the vehicle. A vehicle continuously looks for a trusted neighbor by communicating with the RSU and the TA; if it finds one then it changes its pseudonym (Npseud). As soon as a vehicle discovers another RSU, the life of its current pseudonym ends and another cycle begins. The communication between entities is secure in this scheme and location privacy is ensured. However, since a vehicle generates and changes its pseudonym each time it identifies a new trusted neighbor, this scheme incurs heavy computation overhead on the network. Benarous et al. [16] proposed a pseudonym-changing strategy that attempts to confuse any attacker at the pseudonym update phase to thwart linkability. Their scheme prevents linkability and tracking both in high and low-density roads. When a vehicle is in a low-density neighborhood, it broadcasts beacons with altered location and the speed information for a specific period of time to confuse the attacker and his predictions. When the next time period for pseudonym change is reached, the vehicle's pseudonym is updated and beacons with their real position and speed are sent. This scheme has a drawback, if there is an emergent event, the vehicle switches back to normal mode (i.e., broadcast beacons with real position and speed) to report the event which opens the door for an attacker to track the vehicle in the low-density traffic environment.

In summary, most of the above-mentioned pseudonym-changing strategies have some drawbacks. As we noticed, many of the mix zone schemes use radio silence periods and thus compromise safety to increase the unlinkability of pseudonyms. In our scheme, we did not use any silence period and thus does not disrupt communication between entities. Our goal is to create a balance between safety and privacy so that our proposed scheme does not pose any risk due to a pause in the transmission of safety messages, hence the chances of accidents are reduced substantially. To overcome the limitations of mix zone schemes, various mix content schemes are proposed which use techniques such as the exchange of pseudonyms between vehicles, implicit triggering techniques, etc. The pseudonym-swapping technique significantly impacts the overall working of pseudonym authentication and lacks accountability. In our scheme, vehicles change their pseudonym automatically from a pre-assigned pool of pseudonyms. A possible drawback associated with the triggering technique is that if there are not a sufficient number of vehicles in a mix zone, the adversary may trace the vehicle. Therefore, the triggering technique is bound to the anonymity set size or the number of neighboring vehicles. Our scheme overcomes this limitation– it is applicable in both dense and sparse networks and allows every vehicle in all conditions to change its pseudonym.

3.4.4 Simulation

We simulate the proposed architecture in the MATLAB environment. The detailed simulation parameter settings are shown in Table 3.3.

We used the RSA algorithm for security in our proposed model, which is a widely used algorithm for the secure transmission of data and AODV [63] as a routing

Table 3.3: Simulation parameter settings.

Parameters	**V**alues
Simulation area	1000 m x 1000 m
Number of $RSUs$	4
Number of RCs	1
Number of vehicles	50, 150, 200, 250, 300
Vehicle transmission range	100m
RSU transmission range	300m
Vehicles speed	30m/h
Routing protocol	AODV
Packet size	512 bytes

protocol. Figure 3.8 shows the communication between nodes (vehicles), RSUs, and RC. For example in Figure 3.8, when the vehicle (e.g., Vs) senses an event, it sends a message to a nearby RSU (e.g., RSU1). Next, RSU1 authenticates the message and then sends it to the RC, which in turn forwards it to the intended vehicles through RSUs.

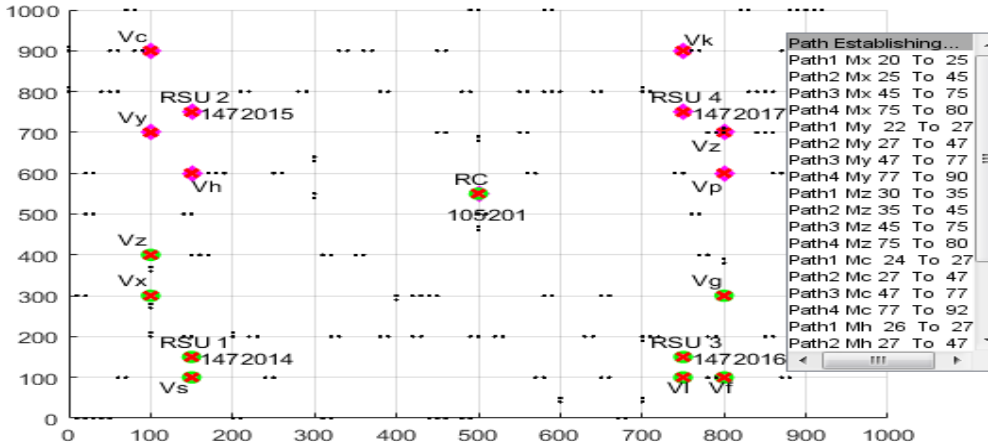

Figure 3.8: Simulation Entities.

We evaluate the effect of different numbers of vehicles on performance. The number of vehicles used is 50, 100, 150, 200, 250, and 300. Figure 3.9 illustrates the comparison of the energy consumption with the number of vehicles in different data rates (packets sent per unit of time during the communication process). As shown in this figure, initially the energy consumption for different data rates is approximately the same. Additionally, the total energy consumption of the network increases sharply as the number of vehicles in the network increases, therefore, an increase in vehicles' density results in an increase in the number of communications and more packets

exchanged between sources and destinations. For example, when the number of vehicles is 300, the 4 packets per second consume less energy than the 8, 10, 12, and 14 packets per second.

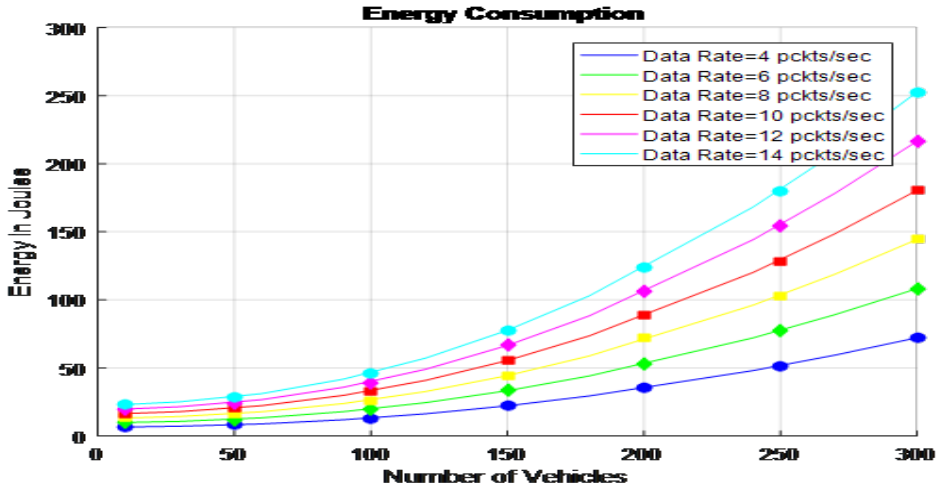

Figure 3.9: Energy consumption.

Figure 3.10 shows the comparison of throughput (ratio of the total amount of successfully transferred data to the total amount of time required to transfer data) with security (encryption and decryption process) and without security. The throughput of the network increases with an increase in packet rate, which means an increase in the amount of data transmission. It is also clear that the throughput without security performs better than the throughput with security. This is due to the process used in the encryption and decryption of sensitive messages which needs more time to transmit packets. In addition, when vehicles are dense, the overhead due to all vehicles transmitting messages about the same phenomena will be high. RSUs aggregate those messages to eliminate redundant messages.

3.5 Related Work

Cluster-based vehicular cloud architectures proposed in [10] and [36], group vehicles according to their location and speed. Both of these schemes rely on a cluster head (CH) which is elected by the vehicles in the cluster, and this CH performs the creation, maintenance, and deletion of vehicles in that cluster. A similar approach is proposed by Chaqfeh et al. [27], where vehicles in a specific region form a vehicular cloud and elect a broker among them. The broker collects the desired data from the vehicles and then sends it to a cloud server if further processing is required. None of these schemes scale well as the number of vehicles increases. When vehicles are moving fast, frequent CH and broker elections occur which result in large message overhead. In addition, if CH or broker fails, data aggregated by them may be lost.

Architecture combining VANET and cloud has been proposed by researchers in [44, 107]. In these schemes, vehicles collect data and send them to the cloud

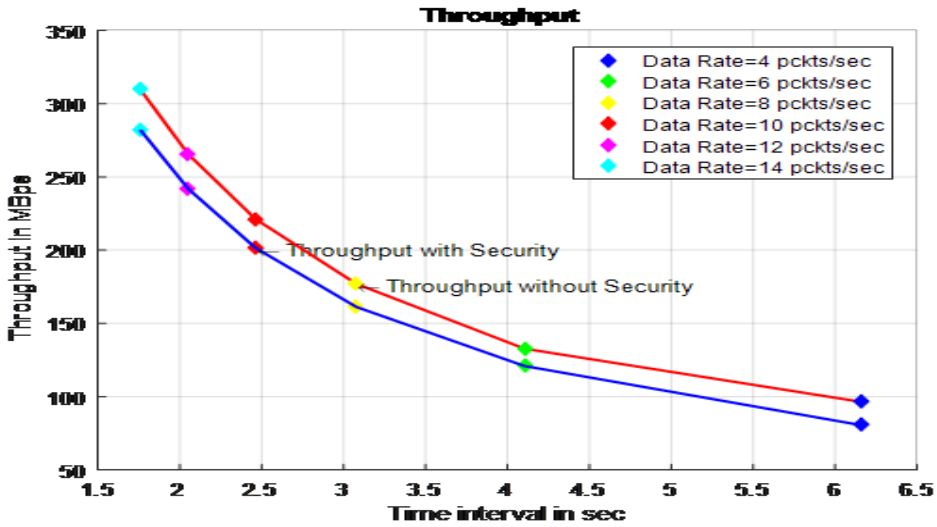

Figure 3.10: The average throughput for the simulated VC.

through a mediator or an RSU. Vehicles in the same area could collect the same data, so this leads to redundancy and results in message overhead. Like [10, 36], they also suffer from a single point of failure– if the mediator or the *RSU* fails, data aggregated could be lost.

Many of the solutions proposed [100, 45, 1, 17] do not address the security and privacy issues. Ensuring security and privacy is an important issue in the vehicular cloud; if information exchanged between entities is modified by a malicious vehicle, serious consequences such as traffic congestion and accidents can occur. In addition, sensitive data could be lost, and human lives also could be in danger. Hence, messages sent by vehicles must be authenticated and securely delivered to vehicles in the appropriate regions. Furthermore, privacy-related information such as the driver's name, position, and traveling route must be protected. If vehicles cannot communicate anonymously, an attacker could easily trace vehicles by monitoring the messages sent by that vehicle.

Several privacy-preserving authentication schemes such as cooperative authentication [47], anonymous authentication [13], and dual authentication [58] have been proposed. In these schemes, vehicles communicate not only with each other but also with the *RSU*s or the *TA* (Trusted Authority) to verify the authenticity of the messages. Although they ensure authentication and privacy, these schemes do not scale well when traffic becomes heavy– vehicles may not be able to verify all the messages sent by their neighbor vehicles in a timely manner, which could result in message loss.

3.6 Summary

In this chapter, we proposed a secure and distributed architecture for the vehicular cloud. This architecture is hierarchical and consists of vehicles, roadside units, re-

gional clouds, and a central cloud. Each regional cloud covers a region (e.g., city, state) and processes the information collected from vehicles through the *RSU*s, and provides on-demand services to vehicles in its region. These regional clouds further communicate with the central cloud and exchange information between themselves to provide a wide range of services to vehicles. Our architecture also copes with *RSU* failures. In addition, we designed a scheme for changing pseudonyms that ensures privacy in all traffic conditions. Our scheme depends on mix zones for changing pseudonyms to make it difficult for an attacker to link pseudonyms. Our scheme also ensures the confidentiality and authenticity of messages.

We analyzed the computation overhead for ensuring security at the RSU side and the vehicle side when the messages are sent from vehicles to the RC and vice versa. Results show RSU has less computation overhead than vehicles because it has more computation power. In addition, there is no computation overhead at RSU when messages are sent from RC to vehicles because RSU doesn't encrypt and/or authenticate messages, it's the vehicle's responsibility to do so. Also, we evaluated our scheme with respect to energy consumption and throughput. We observed that when messages are encrypted, it incurs very little overhead.

Chapter 4 A Low-Overhead Message Authentication and Secure Message Dissemination Scheme for VANETs

4.1 Introduction

In infrastructure-based architectures for VANETs, vehicles use *RSU*s to form a VANET. In some schemes [47, 56], vehicles authenticate each other, while in other schemes [29, 115], vehicles use *RSU*s for authenticating disseminating messages sent by vehicles in its region. If traffic becomes heavy, it may not be possible for *RSU*s to receive messages about events observed by all vehicles in its region, authenticate them, and disseminate them in a timely manner, especially because the same event will be observed and sent by many vehicles in its region. In this chapter, we address this problem and propose a solution.

In our approach, when the density of vehicles in an *RSU*'s region is high, the *RSU* divides its region within its transmission range into several sub-regions and selects one vehicle in each sub-region as the Group Leader (GL). The GL selected in a sub-region is supposed to collect messages sent by vehicles in its sub-region, authenticate them, aggregate them, and forward them to the *RSU*. This reduces the overhead related to message authentication for the *RSU*.

Following are the contributions in this direction:

- We propose a low-overhead message authentication and secure message dissemination scheme for VANETs. Vehicles themselves do not authenticate messages. *RSU*s are responsible for collecting, aggregating, authenticating, and disseminating messages to vehicles.
- To reduce the message authentication overhead, *RSU*s can select some vehicles in its region as group leaders (GLs) to collect/aggregate messages from vehicles in their sub-regions and send them to the *RSU* for further aggregation and dissemination.
- Our scheme ensures the authenticity and integrity of messages using digital signature based on public key cryptography.

The rest of the chapter is organized as follows. We discuss some related works in Section 4.2. In Section 4.3, we describe our proposed approach. In Section 4.4, we present the security and privacy analysis of our approach. Finally, Section 4.5 summarizes the chapter.

4.2 Related Works

Many privacy-preserving authentication schemes, such as anonymous authentication [13], cooperative authentication [47], and dual authentication [58] have been proposed. For example, Azees et al. [13] proposed a PKI-based efficient anonymous authentication scheme with a conditional privacy-preserving (EAAP) scheme

for VANETs. The vehicles and RSUs communicate anonymously to provide privacy and anonymity during the authentication process, and the TA can revoke a misbehaving vehicle and find out its real identity in case of a dispute. This scheme is secure against different attacks (e.g., impersonation attacks, message modification attacks, etc). However, in the above schemes [47, 13, 58], vehicles communicate not only with each other but also with the RSUs to verify the authenticity of the messages.

Schemes presented in [115, 67, 110] used RSUs for authenticating, processing, and disseminating messages received from vehicles in its region. In [110], a safety warning system in fog-cloud-based VANETs using a Certificateless Aggregation Signcryption Scheme (CASS) has been proposed. Vehicles send traffic messages to the RSUs, which act as fog nodes. These fog nodes process and aggregate the received messages. These schemes [115, 67, 110] address the security and privacy issues of VANETs. However, they do not consider heavy densities of vehicles, which may cause increased computation and communication overhead.

In our scheme, vehicles do not form clusters among themselves. Each RSU can decide when and where to form clusters in its region, based on the density of vehicles and other parameters such as the region from which the RSU receives a large number of messages. In addition, the RSU assigns a Group Leader GL (the Group Leader is not elected) for each cluster and the GL is responsible for collecting, authenticating, and aggregating the messages received from its cluster/group and forwarding them to the RSU. The RSU is responsible for collecting the messages sent by the GLs in its region, authenticating them, aggregating them, and forwarding them to the vehicles in its region and/or other RSUs for further dissemination. This approach reduces the computation and communication overhead for the RSUs.

4.3 Proposed Approach

In this section, we present our system model and describe the proposed method for authenticated message dissemination in detail. The acronyms used in this chapter are listed in Table 4.1.

4.3.1 System Model

The system model for our scheme is shown in Figure 4.1. It consists of Department of Motor Vehicles (DMV), Road Side Units (RSUs), On-Board Units (OBUs), and Group Leaders (GLs). We describe the functions of these entities next.

- **DMV:** We assume that all vehicles are registered with a trusted authority (TA), such as the Department of Motor Vehicles (DMV), that administers the registration of the vehicles. The DMV is assumed to be trusted and cannot be compromised. The DMV generates its public and private keys (PU_{DMV}, PR_{DMV}) and distributes a PU_{DMV} to all RSUs and vehicles securely. In addition, the DMV generates pseudo-IDs (PID_v) for each vehicle, certificates corresponding to each pseudo-ID of a vehicle ($Cert_v$) where

Table 4.1: acronyms used in this chapter.

acronyms	**D**escription
ID_A	Identity of Entity A
PID_A	Pseudo Identity of Entity A
M	A Message
v	Vehicle v
ts	Timestamp
PR_A	Private Key of Entity A
PU_A	Public Key of Entity A
K	Symmetric Key established between two communicating parties
$SIG_A(M)$	Signature of M Signed using A's Private Key
$H()$	Hash Function
$E(M, K)$	Encryption of M with Key K
RSU	Roadside unit
GL	Group Leader
DMV	Department of Motor Vehicles
$Cert_v$	Certificate issued to vehicle v by the DMV
$Cert_{RSU}$	Certificate issued to RSU by the DMV

$Cert_v = E((PID_v, PU_v, ts), PR_{DMV})$, and certificates of RSUs ($Cert_{RSU}$) where $Cert_{RSU} = E((ID_{RSU}, PU_{RSU}, ts), PR_{DMV})$.

- **Vehicle:** Each vehicle is assumed to be equipped with an On-Board Unit (OBU) for computation and communication with RSUs as well as with other vehicles. The OBU stores the vehicle's public and private key pair (PU_v, PR_v), its pseudo-IDs, certificates corresponding to each pseudo-ID of the vehicle ($Cert_v$ signed by the DMV), and the public key of the DMV (PU_{DMV}).

- **RSU:** The Road Side Units (RSUs) are fixed entities along the roadside which facilitate V2V and V2I communication. RSUs are connected to each other and to the DMV, possibly through the Internet. In our scheme, a RSU collects the messages sent by the vehicles in its region, authenticates the messages, aggregates the messages, and forwards them to vehicles within its region, as well as to vehicles in other regions as needed.

- **Group Leader (*GL*):** Each *RSU* divides its region into sub-regions based on the density of vehicles in the region. Then, the *RSU* selects one vehicle in each sub-region as a *GL*. The *GL* is responsible for collecting, authenticating, and aggregating messages sent by vehicles in its sub-region and for sending them to the *RSU*. The *GL* is also responsible for receiving messages from the *RSU*, authenticating them, and disseminating them to vehicles in its sub-region.

We describe the proposed method in detail next.

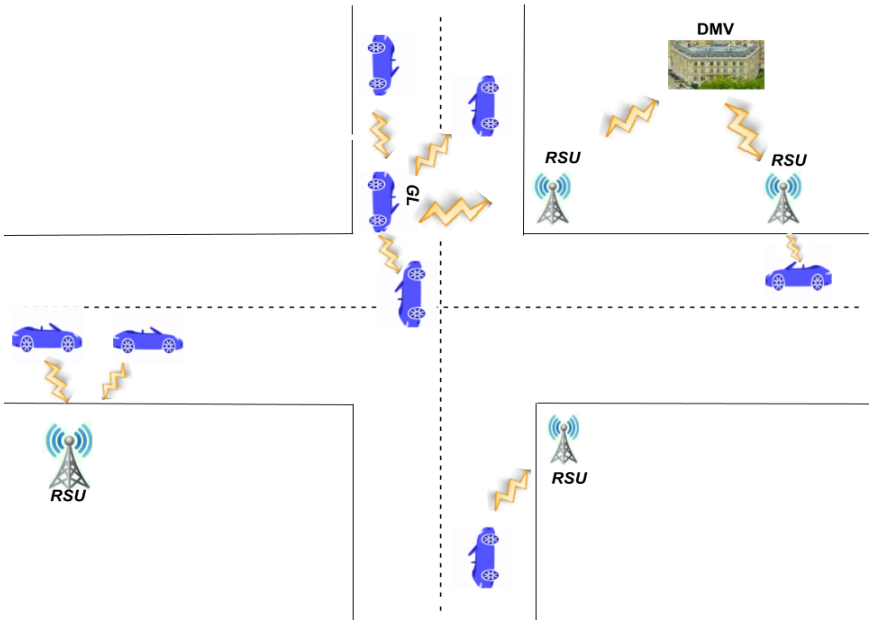

Figure 4.1: System model for VANETs.

4.3.2 Proposed Method

In our scheme, *RSU*s are responsible for verifying the authenticity and integrity of messages sent by vehicles before disseminating them to other vehicles or *RSU*s. If traffic is heavy in the region of an *RSU*, the *RSU* may not be able to receive messages from all vehicles in its region, process them, and disseminate them in a timely manner due to the authentication, aggregation, and communication overhead involved. To help *RSU* minimize this overhead, the *RSU* divides its region into sub-regions and selects one vehicle in each sub-region as the Group Leader (*GL*). These Group Leaders help the *RSU* with receiving, authenticating, and aggregating messages from vehicles in its sub-regions and forward them to the *RSU*. The *RSU*, in turn, is responsible for collecting, authenticating, and further aggregating the messages received from all the *GL*s in its region, and disseminating them to all vehicles in its region through the *GL*s or to vehicles in other regions through other *RSU*s, as necessary. Thus, *RSU*s incur less computation and communication overhead for collecting, authenticating, and disseminating messages. Following is the list of assumptions made in this chapter:

1. We assume that the clocks of RSUs, the DMV, and the vehicles are loosely synchronized. This can be achieved using time received from a GPS. Messages are time-stamped using the local clock time to verify the freshness of the messages.

2. Certificates issued by the DMV for the vehicles and RSU are used for the authentication of vehicles and RSUs.

3. We do not address the issue of determining malicious vehicles or RSUs. Several approaches have been proposed in the literature to identify malicious entities in VANETs. Any of those approaches can be used for determining malicious vehicles. Once a vehicle is determined to be malicious, the DMV revokes its certificate and includes the certificate in the Certificate Revocation List (CRL). The DMV broadcasts the CRL to all RSUs when it changes. The RSUs, in turn, broadcast the CRL to vehicles in its region.

4. When a vehicle v enters the region of an RSU (i.e., v is within the transmission range of an RSU), even though v will be able to receive messages sent by the RSU, v may not be able to send messages directly to the RSU because the RSU may not be within the transmission range of v. In this case, v uses an underlying routing algorithm to send messages to the RSU through other vehicles. Any of the many routing algorithms proposed in the literature can be used for that purpose.

Next, we describe our approach in detail.

When a vehicle v enters the region of an RSU: Each RSU periodically broadcasts its $Cert_{RSU}$. When a vehicle v enters an area covered by an RSU, v retrieves the public key of the RSU from $Cert_{RSU}$ and checks its CRL to see if this RSU's certificate has been revoked (the certificate of an RSU could be revoked if it is removed from the system). If not, then v sends a join request message M to the RSU. The join request message M contains its currently used PID_v, the corresponding certificate $Cert_v$, and a timestamp (ts). After receiving this message, the RSU checks the freshness of the message using the ts. Then, the RSU retrieves the public key PU_v and pseudo-ID PID_v of the vehicle from $Cert_v$ and checks the CRL to determine if the vehicle's certificate has been revoked. If not, then the RSU sends an accept message to v. The accept message contains a symmetric key K to be used for secure communication between the RSU and v, and a timestamp ts, encrypted using the public key PU_v of v; it also attaches the certificate of the RSU, signed by the DMV ($Cert_{RSU}$), and the signature of the RSU (SIG_{RSU}) to the message as follows:

$$M_1 = (RSU, PID_v, (E(\text{"}Accept\text{"}, K, ts), PU_v), Cert_{RSU}, SIG_{RSU}),$$

where

$$SIG_{RSU} = E(H(\text{"}Accept\text{"}, K, ts), PR_{RSU}).$$

Upon receiving the above accept message from the RSU, the vehicle uses the received ts to verify the freshness of the accept message. After that, it verifies the

$Cert_{RSU}$ and the signature of the RSU. Algorithm 5 contains the algorithm illustrating the joining process of a vehicle v when v enters the region of an RSU.

Algorithm 5 When a vehicle v enters the region covered by an RSU

`When a vehicle` v `enters the region covered by an` RSU`:`

Verifies $Cert_{RSU}$ received in the broadcasted message using PU_{DMV};
Retrieves PU_{RSU} from the $Cert_{RSU}$;
Computes $M_1 = (\text{"}Join\text{"}, ts)$;
Encrypts M_1 using public key PU_{RSU} of RSU;
Sends $M_1' = (PID_v, RSU, E(M_1, PU_{RSU}), Cert_v, SIG_v)$ to the RSU, where $SIG_v = E(H(M_1), PR_v)$

`When the` RSU `receives` M_1' `from` v`:`

Decrypts M_1' using PR_{RSU};
Verifies $Cert_v$ using PU_{DMV};
Retrieves PU_v from $Cert_v$;
Verifies the signature using PU_v;

If verification succeeds {

Computes $M_2 = (\text{"}Accept\text{"}, K, ts)$;
// M_2 contains the acceptance message
// for the joining message from v;
// K is the symmetric key to be used between v and RSU;
Encrypts M_2 using public key PU_v of v;
Sends $M_2' = (RSU, PID_v, E(M_2, PU_v), Cert_{RSU}, SIG_{RSU})$ to v, where $SIG_{RSU} = E(H(M_2), PR_{RSU})$};
Else { Discards $M2$; }

`When a vehicle` v `receives` M_2' `from` RSU`:`

Decrypts M_2' using its private key PR_v to obtain M_2;
Verifies SIG_{RSU} using PU_{RSU};
If verification succeeds {
Stores (M_2);}
Else { Discards $M2$. }

Next, we describe how an RSU selects Group Leaders in its region and informs them about being selected.

Informing selected vehicles as Group Leaders: When a vehicle v enters the region covered by an RSU, it sends a join message to the RSU after authenticating the RSU. Then, the RSU authenticates v and sends an "Accept" message, which includes a symmetric key K to be used between v and the RSU. Afterward, the

vehicle can send messages about sensed events to the RSU, encrypting them using K. If the RSU is not within the vehicle's transmission range, the messages are sent to the RSU using an underlying routing algorithm, as we mentioned earlier. Upon receiving "join" messages from vehicles in its region, an RSU can determine the number of vehicles in its region and their location. If the density of vehicles in the region of an RSU is low, the RSU does not need to select a GL. If the density of vehicles in an RSU's region is high, it divides its region into sub-regions and selects one vehicle from each sub-region as the Group Leader (GL). After selecting GLs, the RSU informs the selected vehicles (GLs) of their leadership and sends a proof-of-leadership message $M_1 = E(("Leader", PU_{GL_i}, ts), PR_{RSU})$. The RSU encrypts the M_1 using a symmetric key K, established between v and RSU when v entered the RSU's region, attaches its signature (SIG_{RSU}) to the message, and sends the M_1', where $M_1' = (RSU, PID_v, E(M_1, K), SIG_{RSU})$, and $SIG_{RSU} = E(H(M_1), PR_{RSU})$.

When a GL receives the above message M_1' from the RSU, it decrypts the message using a symmetric key K and uses the received ts to verify the freshness of the message. After that, it verifies the signature of the RSU and stores M_1 as proof of leadership, so it can present it to the vehicles in its sub-region as proof that it is a leader. Algorithm 6 illustrates how an RSU informs the selected vehicles of their leadership (GLs). The GLs are responsible for authenticating, aggregating, and forwarding messages collected from vehicles in their sub-regions. Thus, the RSU only needs to authenticate and process messages that come from GLs. Therefore, the communication and computation overhead for RSUs will be reduced. Moreover, when an RSU needs to send some message to all vehicles in its region or only to vehicles in some sub-regions, it will send that message only to the GLs in those sub-regions, which, in turn, will send it to all the vehicles in its sub-region.

Next, we describe how a vehicle in a sub-region establishes a connection with its Group Leader and communicates with its Group Leader.

When a vehicle v enters the sub-region of a GL: Each GL periodically broadcasts its public key PU_{GL} and the proof of leadership received from the RSU, namely,
$E(("Leader", PU_{GL}, ts), PR_{RSU})$. When a vehicle v enters a sub-region covered by a GL, it retrieves PU_{GL} from the proof of leadership. Then, v sends a join request message M to the GL; M contains a PID_v, $Cert_v$, and timestamp (ts). Upon receiving M, the GL checks the freshness of the message using ts. Then, the GL retrieves PID_v and public key PU_v of the vehicle from $Cert_v$ and checks the CRL to determine if the vehicle's certificate has been revoked. After verification, GL sends an acceptance message and a symmetric key K to be used for secure communication between the vehicle v and the GL. The acceptance message M_1' contains the certificate of the GL, signed by the DMV ($Cert_{GL}$), a K, and a ts, encrypted using the public key PU_v of v as follows: $M_1' = GL, PID_v, (E("Accept", K, ts), PU_v), Cert_{GL}, SIG_{GL})$

Upon receiving the above acceptance message from the GL, v uses the received ts to verify the freshness of the message. After that, it verifies the signatures of the DMV and GL. Note that if v does not receive proof of leadership from a GL (this

Algorithm 6 Assigning Group Leaders (GLs) for selected vehicles by RSU

RSU `determines the number of vehicles and their locations in its region:`
  Based on the density of vehicles in the RSU's region,
  If Density is high {
    RSU selects a set of vehicles as Group Leaders (GLs);

    For each vehicle selected as a GL {
      Computes $M_1 = (E(\text{"}Leader\text{"}, PU_{GL}, ts), PR_{RSU})$;
      Encrypts M_1 using symmetric key K;
      // K is the symmetric key established between v and
      // the RSU when v joined RSU's region;
      $M_1' = (RSU, PID_v, E(M_1, K), SIG_{RSU})$,
      where $SIG_{RSU} = E(H(M_1), PR_{RSU})$;
      Sends M_1' to GL; } }
  `else`{
    No GLs are selected;
    RSU authenticates and process messages from all vehicles; }

`When a` GL `receives` M_1' `from` RSU`:`
  Decrypts M_1' using K;
  Verifies the signature using PU_{RSU};

  If verification succeeds{
    Stores (M_1) as proof of leadership;}
  Else {Discards M_1.}

happens when the RSU has not determined leaders due to the low density of vehicles in its region), after entering an RSU's region, v sends/receives messages to/from the RSU directly, using an underlying routing protocol. Algorithm 7 illustrates the joining process when v is in the sub-region of a GL.

When a vehicle v wants to send a message M to its GL: When v wants to send a message M about an observed event to its GL, it signs and encrypts M and sends M_1 to the GL, where $M_1 = (PID_v, GL, E((M, ts), K), SIG_v)$; here, ts is the timestamp, K is the symmetric key established between v and GL, and PID_v is the pseudo-ID of v.

When GL receives M_1, it decrypts the message using the symmetric key K and checks the freshness of the message using the ts. It uses a signature SIG_v to verify the authenticity and integrity of the message. Then, the GL aggregates the received message with the messages received from other vehicles in its sub-region and forwards the aggregated message to the RSU, and the RSU can further aggregate messages

Algorithm 7 When vehicle v enters a sub-region covered by a Group Leader GL

`When` v `enters the region covered by a` GL`:`

Receives proof of leadership message
$E(("Leader", PU_{GL}, ts), PR_{RSU})$ from the GL;
Retrieves PU_{GL} from the encrypted message using PU_{RSU};
Computes $M_1 = ("Join", ts)$;
Encrypts M_1 using public key of Group Leader PU_{GL}
Sends $M_1' = (PID_v, GL, E(M_1, PU_{GL}), Cert_v, SIG_v)$ to
GL, where $SIG_v = E(H(M_1), PR_v)$;

`When a` GL `receives` M_1' `from` v`:`

Decrypts M_1' using PR_{GL}
Verifies $Cert_v$ using PU_{DMV};
Verifies the signature using PU_v;

If verification succeeds{
Computes $M_2 = ("Accept", K, ts)$;
// M_2 contains the acceptance of GL for v;
// K is a symmetric key between v and GL for further
// communication;
Encrypts M_2 using public key PU_v of v ;
Sends $M_2' = (GL, PID_v, E(M_2, PU_v), SIG_{GL})$ to v,
where $SIG_{GL} = E(H(M_2), PR_{GL})$};
Else { Discards M_2;}

`When` v `receives` M_2' `from the` GL`:`

Decrypts M_2' to obtain M_2;
Verifies SIG_{GL} using PU_{GL};
If verification succeeds{
Stores (M_2); }
Else { Discards M_2; }

received from other GLs in its region and disseminate them to the appropriate sub-regions of its region or regions covered by other RSUs. Algorithm 8 shows this message collection and dissemination process.

Algorithm 8 Vehicle v sending a Message M to its Group Leader GL

`When a vehicle` v `wants to send a message` M `about an observed event:`

Computes $M_1 = (PID_v, GL, E((M, ts), K), SIG_v)$;
Sends M_1 to GL;
// K is the symmetric key established in the
// Algorithm 7.

`When the` GL `receives` M_1 `from` v`:`
Decrypts M_1 using the symmetric key K and retrieves
the message M;
Checks the timestamp ts;
Verifies the signature using public key PU_v of v;
Aggregates (M) with other messages sent by other vehicles;
Computes $M_2 = (GL, RSU, E((M, ts), K), SIG_{GL})$;
Sends M_2 to RSU;
// K is the symmetric key established between the GL and
// the RSU when it entered the RSU's region.

`When the` RSU `receives` M_2 `from` GL`:`
Decrypts M_2 using the symmetric key K and retrieves
the message M;
Checks the timestamp ts;
Verifies the signature using public key PU_{GL} of GL;
Aggregates (M) with other messages sent by other GLs;
Disseminates the message to the appropriate regions through
other RSUs as well as vehicles in its region through the GLs.

Certificate Revocation List (CRL) distribution and certificate revocation process. Misbehaving vehicles can send malicious messages to other vehicles; these misbehaving vehicles should be detected and punished. IEEE 1609.2, the standard for Wireless Access in Vehicular Environments (WAVE)—Security Services for Applications and Management Messages [97], has specified that the vehicle must be authenticated using certificates issued by the TA and defined the CRL that contains the list of the revoked certificates that are updated timely and disseminated in the vehicular network. Once the CRL is distributed to the vehicles, it can compare the certificate of a vehicle with the list and determine if it has been revoked.

In our scheme, the DMV will manage and maintain the updated CRL. The DMV will distribute the CRL to the RSUs, which, in turn, will distribute them to all vehicles in their region directly or through the GLs, if the GLs have been selected. The RSUs and GLs always check the authenticity of the vehicles using the CRL. If a vehicle is found to be malicious, the RSU sends the certificate information of the vehicle to the DMV. Then, the DMV adds the certificate to the CRL and distributes the updated CRL to all RSUs. Note that vehicles only communicate either with the RSU or the GL and that no communication between themselves occurs, which reduces the communication and computation overhead. We do not address the problem of detecting malicious vehicles. Many researchers have addressed the malicious vehicle detection problem in VANETs [76, 43]. Any of those schemes can be used to detect malicious vehicles.

4.3.3 Some Optimizations for Our Approach

In our scheme, when a vehicle v enters the region of an RSU, it obtains a symmetric key K through the *Accept* message $M_2 = (\text{"}Accept\text{"}, K, ts)$ from the RSU for establishing secure communication between v and the RSU (please see Algorithm 5). This key K is used by v to encrypt messages and send them to the RSU in the absence of GLs; this key is also used by the RSU to send messages, as well as $CRLs$, securely to v, in the absence of GLs. To reduce this overhead caused by sending unicast messages, the RSU can attach a group key GK to the accept message as $M_2 = (\text{"}Accept\text{"}, GK, K, ts)$; then, GK can be used by the RSU to broadcast (instead of unicasting) securely the $CRLs$ as well as other messages to all vehicles in its region. Similar optimizations can be performed in Algorithm 7 when a GL assigns a symmetric key K to a vehicle v through the message $M_2 = (\text{"}Accept\text{"}, K, ts)$.

4.4 Results

In our scheme, encryption, and signature are fundamental security mechanisms used to resist impersonation, eavesdropping, replay, and modification attacks. The message that is sent by a vehicle v to its GL to be modified must be decrypted, modified, and then encrypted by an attacker using the $v's$ shared symmetric key. To decrypt the message, the attacker needs the symmetric key shared between the v and GL, which is not available to the attacker, thus making it impossible to modify the message. Replay attacks are prevented using timestamps. In our scheme, an attacker cannot generate a valid signature of other vehicles because the attacker does not know the private key of the vehicle. As a result, an attacker cannot send a malicious signed message without being detected.

Our scheme is secure against impersonation attacks: To perform an impersonation attack, the attacker should be able to obtain the private key PR_v of a legitimate vehicle v, which the attacker does not possess. In addition, an attacker cannot impersonate a vehicle v, as the message encrypted using a shared symmetric key K between v and GL (or between v and the RSU) cannot be decrypted without using K, which the attacker does not possess.

Our scheme preserves privacy—an attacker cannot discover the vehicle's identity: Vehicles are assigned pseudo-IDs. A vehicle never uses its real ID in any communication. This prevents discovering the real identity of the vehicle and prevents attackers from linking messages from the same vehicle using multiple pseudonyms. During registration, a vehicle is assigned a set of pseudonyms and associated certificates. Vehicles can use any of the pseudonym-changing strategies presented in the literature [67, 112] to change pseudonyms. Therefore, the privacy of vehicles is preserved.

Communication and Computation Overhead: In our scheme, if the density of vehicles present in an *RSU*'s region is low, it does not select *GL*s. If the density of vehicles in its region is high, then the *RSU* selects *GL*s from the vehicles to help the *RSU* with authenticating messages. The *GL*s are responsible for authenticating, aggregating, and forwarding messages received from vehicles from their sub-regions. Thus, an *RSU* only needs to authenticate and process messages that come from the *GL*s. Therefore, the communication and computation overhead for an *RSU* are reduced. Note that an *RSU* sends messages to vehicles in its region through *GL*s; vehicles only need to authenticate messages received from its *GL* if the density of vehicles is high, and not from other vehicles, so the communication and computation overhead is low for the vehicles as well.

Figure 4.2 shows a comparison of the total communication cost of our scheme and that of the SEMA scheme [102], in terms of the number of messages exchanged between an *RSU* and the vehicles in its region. For the purpose of comparison, vehicle density within the region of an *RSU* is assumed to be high when the number of vehicles in its region is 1000 or more, and the average number of messages exchanged between a vehicle v and *RSU* is 2; otherwise, we assume that the density is low. Figure 4.2 shows the average number of messages exchanged between an *RSU* and vehicles in its region with this assumption; if the number of vehicles is less than 1000 in its region, the *RSU* authenticates and processes messages received from all vehicles within its region; if there are more than 1000 vehicles in its region, the *RSU* needs to authenticate messages that come from the *GL*s only. As a result, in our scheme, the communication cost is lower on the *RSU* side. For example, if there are only 400 vehicles present in the region of an *RSU*, the *RSU* will authenticate the same number of messages (which is $400*2 = 800$ messages) in our scheme and in the SEMA scheme [102]. For comparison purposes, to compute the number of *GL*s needed in an *RSU*'s region, we assume that a predefined threshold is 100 for each *GL*; i.e., if there are 1000 vehicles, the number of *GL*s needed is ($\lceil(1000/100)\rceil = 10$) and the number of messages exchanged between the *GL*s and the *RSU* would be ($\lceil(1000/100)\rceil * 2 = 20$) under our scheme, whereas under SEMA [102], the number of messages exchanged would be ($1000 * 2 = 2000$). Therefore, the total communication cost increases significantly with the increase in the number of vehicles under SEMA [102]. On the contrary, under our scheme, the communication cost is significantly lower. This is primarily because message collection overhead is shared by selected vehicles (GLs) in the RSU's region.

We analyzed the computation overhead associated with encryption and authentication using a Toshiba computer with an Intel i3 quad-core processor with 2.50-GHZ

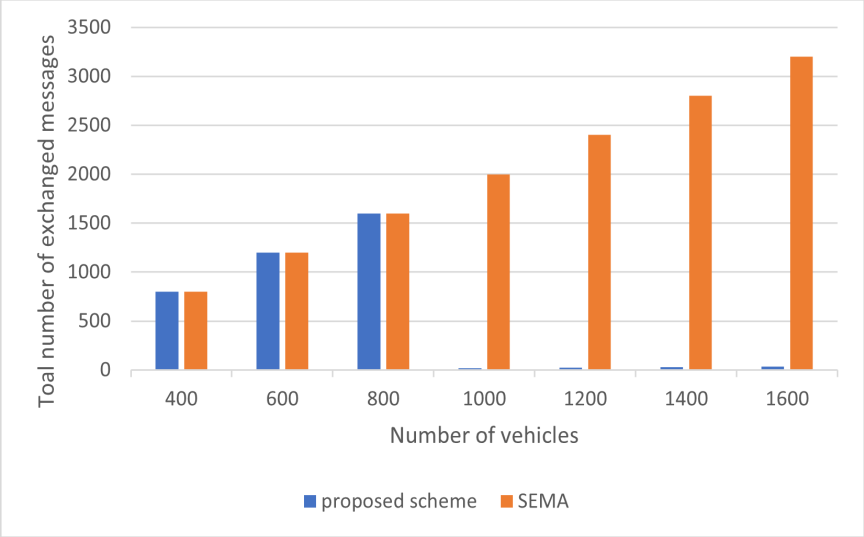

Figure 4.2: Communication overhead comparison.

clock frequency and 6 gigabytes of memory, running Windows 8.1 operating system. The public key cryptography-based signature and encryption scheme are based on RSA (Rivest–Shamir–Adleman) cryptography. Following are some notations used for presenting our results: time for computing RSA-based signatures (T_{sign}); time for signature verification (T_{verify}); time for encrypting a message using a public key (T_{EPU}); time for decrypting the message using a private key (T_{DPR}); time for encrypting a message using a symmetric key (T_{EK}); time for decrypting a message using a symmetric key (T_{DK}). We used the AES (Advanced Encryption Standard) to encrypt and decrypt the messages using a symmetric key. The execution time of the above operations is presented in Table 4.2. We used a message size of 39 bytes, as specified in the IEEE 1609.2 standard, for the encryption and the corresponding decryption operations.

Computation Overhead on *GL*: The *GL* is responsible for collecting, authenticating, and aggregating messages received from vehicles in its sub-region and forwarding them to the *RSU*. Figure 4.3 shows the computation overhead incurred by a *GL* for decrypting and verifying the signature of messages received from the vehicles in its sub-region as well encrypting and signing those messages for sending them to the *RSU* for a number of messages ranging from 50 to 500.

Computation Overhead for *RSU*: Figure 4.4 shows a comparison of the computation overhead between our scheme and SEMA [102] at an *RSU* for a varying number of signature verifications. Our scheme incurs significantly lower overhead compared to SEMA [102]. This is due to the use of the *GL*s, which help the *RSU* with the authentication and aggregation process of the messages sent by vehicles. For example, when the number of signatures reaches 1400, the overall cost is approximately 7 ms for the scheme in [102], whereas it is only 0.7 ms for our scheme.

Table 4.2: Execution time for different operations (milliseconds).

Operation	**T**ime
T_{sign}	0.06
T_{verify}	0.005
T_{EPU}	1.274
T_{DPR}	2.654
T_{EK}	1.166
T_{DK}	2.128

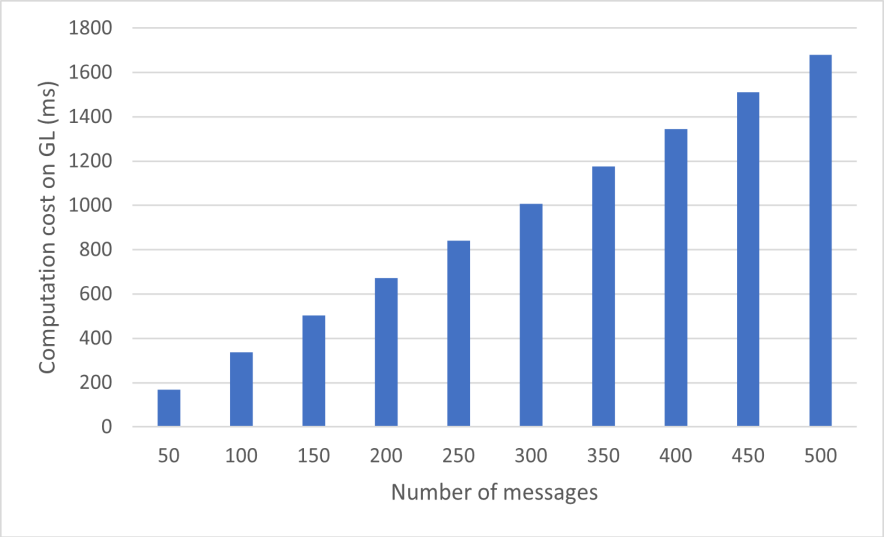

Figure 4.3: Total computation time at a GL for various numbers of messages.

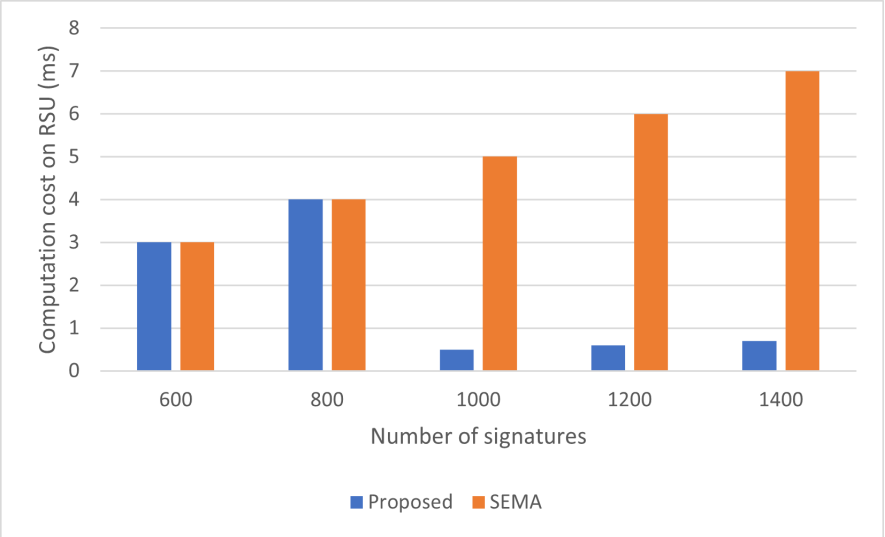

Figure 4.4: Computation time at the RSU.

4.5 Summary

In this chapter, we presented a low-overhead *RSU*-aided message authentication and dissemination scheme. In this scheme, when the overhead for collecting, authenticating, aggregating, and disseminating messages increases for an *RSU*, the *RSU* can designate some of the vehicles in its region as Group Leaders and make them share the overhead involved in authenticating, aggregating, and disseminating messages. Thus, this scheme helps the *RSU*s by reducing the computation and communication overhead related to collecting, authenticating, aggregating, and disseminating messages. We have also shown that our scheme is privacy-preserving and secure and resilient to various attacks. We also analyzed and compared the communication and computation overheads of our scheme with an *RSU*-aided approach for authentication and message dissemination.

Chapter 5 Hardware Implementation of On-Board Unit in VANET: Design and Experimental Evaluation

5.1 Hardware Implementation of On-Board Unit in VANET

Information collected by On Board Units (OBUs) located in vehicles can help in avoiding congestion, provide useful information to drivers, etc. However, not all drivers on the roads can benefit from OBU implementation because OBU is not available in all vehicles. Therefore, in this chapter, we design and build a hardware implementation for OBU. This OBU implementation is simple, efficient, and incurs low cost. Evaluation results show that our proposed model can transmit and receive messages (e.g., safety messages) to nearby vehicles, Access Points (APs), and destination with acceptable delay.

5.1.1 Introduction

Nowadays, smart devices can provide applications like mobile health (m-health), location-based services, etc. In such applications, smart devices could perform data sensing and processing. Vehicles are not likely to be equipped with OBUs in the near future. Therefore, smart phones could play a key role in vehicular networking as they provide a set of embedded sensors (e.g., accelerometer), computation, and communication capabilities that could be used in the deployment of VANET applications. Some schemes have been proposed to minimize the risk of accidents using smartphones [41, 106, 113, 75, 38]. In [113], authors proposed a smartphone application called CarSafe, which collects information from both front and back cameras to identify unsafe driving conditions. This application can track and predict whether the driver is disturbed or tired using the front camera. Also, the back camera is used for monitoring road conditions. However, these schemes only provide driver behavior services and not other services. In addition, since they used sensors in their schemes, not all events can be detected by sensors. A number of VANET hardware implementation schemes have been proposed in the literature [7, 83, 93, 95]. These schemes proposed collision-detecting systems to improve traffic efficiency. However, since they used sensors in their schemes and not all events can be detected by sensors.

Our proposed scheme depends on humans and their ability to communicate and it doesn't depend on sensors because not all vehicles are equipped with smart sensors and not all events can be detected by sensors. Users could receive information through our application about their surrounding traffic conditions. For example, if someone sees an incident, they can report it by sending a warning message to *other vehicles, access points, and to different departments* such as police, hospital, fire department, etc., to take proper actions. The OBU in our scheme consists of an Arduino microcontroller [8]- which acts as the brain of our model, a Radio Frequency (RF) module, and a Bluetooth module. The driver will send the message (e.g., obstacle on the road) using his/her phone through the Bluetooth module to the OBU. Then

the OBU will send the message wirelessly to other vehicles (which have OBUs as well) using Radio Frequency (RF) module. When OBUs receive the message, they will send the message to other drivers' cell phones through the Bluetooth module, so other drivers can take proper action [65]. The details of the proposed model are explained in detail in the next section of the chapter.

The rest of the chapter is organized as follows. In Section 5.1.2, we describe our proposed model. In Section 5.1.3, we present the hardware implementation and performance evaluation. We discuss and compare some related work in Section 5.1.4. Finally, Section 5.1.5 summarizes the chapter.

5.1.2 Proposed Model

In this section, we present our system model and describe the proposed architecture in detail.

5.1.2.1 System Model

Figure. 5.6 illustrates the proposed architecture which consists of On Board Units (OBUs), Cell Phones, Access Points (APs), and Destinations.

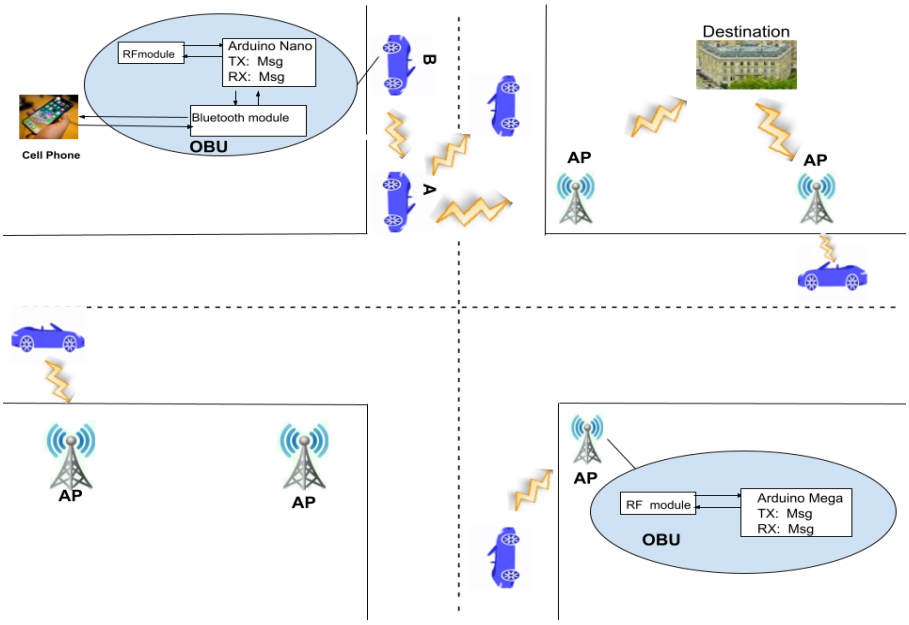

Figure 5.1: Message Dissemination Scheme for Rural Areas.

- **On Board Unit (OBU):**
  OBU consists of the following components:
  - **Arduino Microcontroller:**
    Arduino microcontroller is a special-purpose mini computer. It has a dedicated input and output device and ports to control the device components.

The microcontroller is attached with an RF module and Bluetooth to send and receive data either from the vehicle or infrastructure [33]. The purpose of the Arduino in our architecture is to transmit and/or receive messages and then forward them to other Arduinos located in other vehicles. We used two types of Arduinos, Arduino nano for vehicles and Arduino mega for access points. The Arduino mega has more storage and computational power than Arduino nano. Table 5.1 compares the specification of the two types of Arduinos.

Table 5.1: Comparison of Arduino nano and Arduino mega [8]

	Arduino Nano	**A**rduino Mega
Micro-controller	ATmega328	ATmega2560
Digital I/O Pins	14	54
Analog Input Pins	8	16
Flash Memory	32 KB	256 KB
SRAM	2 KB	8 KB
Clock Speed	16 MHz	16MHz

  - **Radio Frequency (RF) module:**
    An RF module is a small electronic device used to transmit and/or receive radio signals between two devices [104]. The transmission range of the RF module is 100 m. In our architecture, an RF module is used to enable wireless communication between Arduino devices.
  - **Bluetooth module:**
    Bluetooth is used for communication between Arduinos and cellphones wirelessly. We used Bluetooth module HC-05. We created an Arduino-Bluetooth interface for exchanging messages between Arduino and the cellphone of a driver. Figure 5.2 shows the components of the OBU of a vehicle as well as an access point.

- **Cell Phone:**
  We use the serial Bluetooth application downloaded on the cellphone that allows us to write a message and send it to Arduino through the Bluetooth module. In addition, the serial Bluetooth application reads messages that come from Arduino microcontrollers.

- **Access Point (AP):**
  Access points are fixed units that can be deployed along the roadside (e.g., at major road intersections, gas stations, etc.). The AP collects the messages sent by vehicles and forwards them to the Destination. In addition, it can send messages to vehicles within its transmission range. APs are assumed to be connected to destinations, possibly through the Internet.

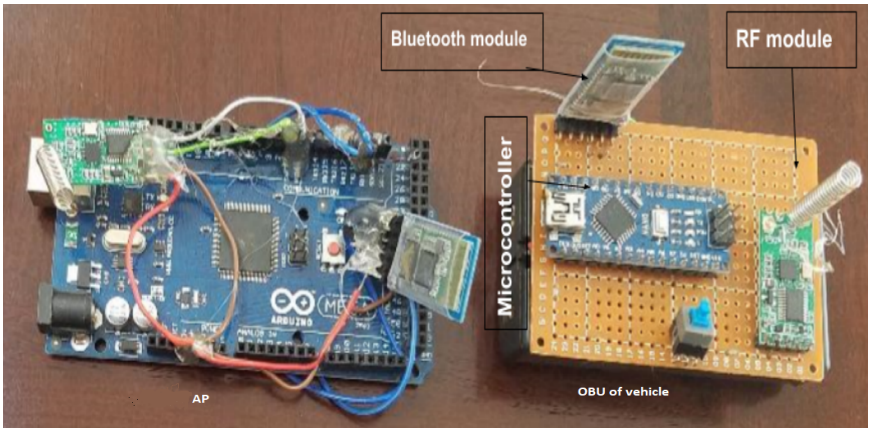

Figure 5.2: Components of OBU and Access Point.

- **Destination:**
  Destinations in our model could be the traffic department, police department, hospital, etc.

5.1.2.2 Proposed Architecture

In this section, we describe our architecture in detail.

In the proposed architecture, information collected by drivers in an area (e.g., driver A sees an accident) is sent to other vehicles or nearby access points through the OBU. For example, as shown in Figure 5.1, the driver in vehicle A sends a message (e.g., notifying about an accident) from the driver's cellphone via Bluetooth to the OBU of vehicle A. Then, the OBU forwards the message to the nearest vehicle (e.g., vehicle B) using its RF module. When the OBU of vehicle B receives the message, it forwards it to cellphone B using its Bluetooth module, and then driver B can take appropriate action. The messages also could be forwarded to the access point which will forward them to the destination.

We built OBU hardware that can be attached to vehicles and can also be installed in some areas to increase connectivity; These are called Access Points (APs). These APs will send the messages received from vehicles to the destination. If there are no vehicles or APs within the transmission range of a vehicle, the vehicle stores and carries the message until it gets closer to the next AP or another vehicle.

APs are responsible to forward the messages to nearby vehicles and/or the destinations (e.g., police department, traffic department, health department, etc.). The destination is responsible for processing the messages and sending the services that the original source vehicle might need.

5.1.3 Implementation and Evaluation

A general overview of our implementation is shown in Figure 5.3. We built OBU hardware for vehicles and AP. We used 3 OBUs- vehicles, 1 access point, and a Personal Computer (PC)- Destination.

Messages in our model are transferred from a cellphone of the driver via serial Bluetooth terminal of the android application by pairing two Bluetooth devices (Bluetooth of cell phone and HC-05 Bluetooth module of OBU). After pairing the devices, the Arduino nano located in the OBU broadcasts the message to neighbor vehicles (OBUs) or nearby AP via RF module. Figure 5.3 illustrates the process of sending messages between vehicles, AP, and Destination. In our scheme, any vehicle (e.g., vehicle 1) can broadcast a message to neighboring vehicles within its transmission range (e.g., vehicle 2, vehicle 3), as well as AP. Also, the AP could broadcast the message to vehicles within its transmission range (vehicle 1, vehicle 2, and vehicle 3) as well as the Destination (PC). The messages sent/received to/from the PC is shown using open terminal software (e.g., putty). For example, if the Destination wants to inform about an incident, the Destination (PC) can send a message to AP which in return forwards it to vehicles within its transmission range.

Figure 5.3: Sending messages between vehicles, AP and PC.

We connected AP with a personal computer (PC) to measure the end-to-end delay and the packet delivery ratio using MATLAB environment. The end-to-end delay is the time taken for a packet to reach the destination (see Figure 5.4). We used AES Encrypter/Decrypter to encrypt and decrypt the messages. We used 6 data packet sizes: 32, 64, 128, 256, 512, and 1024 bytes. The packets were forwarded wirelessly through Bluetooth module HC-05 of the vehicle from a smartphone then the vehicle (OBU) received it through the RF module and transmitted it to AP, which transmitted it to the PC. In analyzing the end-to-end delay results (Figure 5.4), we have observed that the transmission times grow with the size of the data packet transmitted. For instance, the average delay of a data packet size of 64 bytes is longer than that of 32 bytes for plaintext and encrypted messages. In addition, encrypted messages take more time than plaintext ones due to their size which is bigger than plaintext messages.

Figure 5.5 shows the experimental results in terms of Packet Delivery Ratio (PDR), which refers to the ratio between the packets successfully delivered to the number of packets sent by a source vehicle. Transferring the data of the packets were

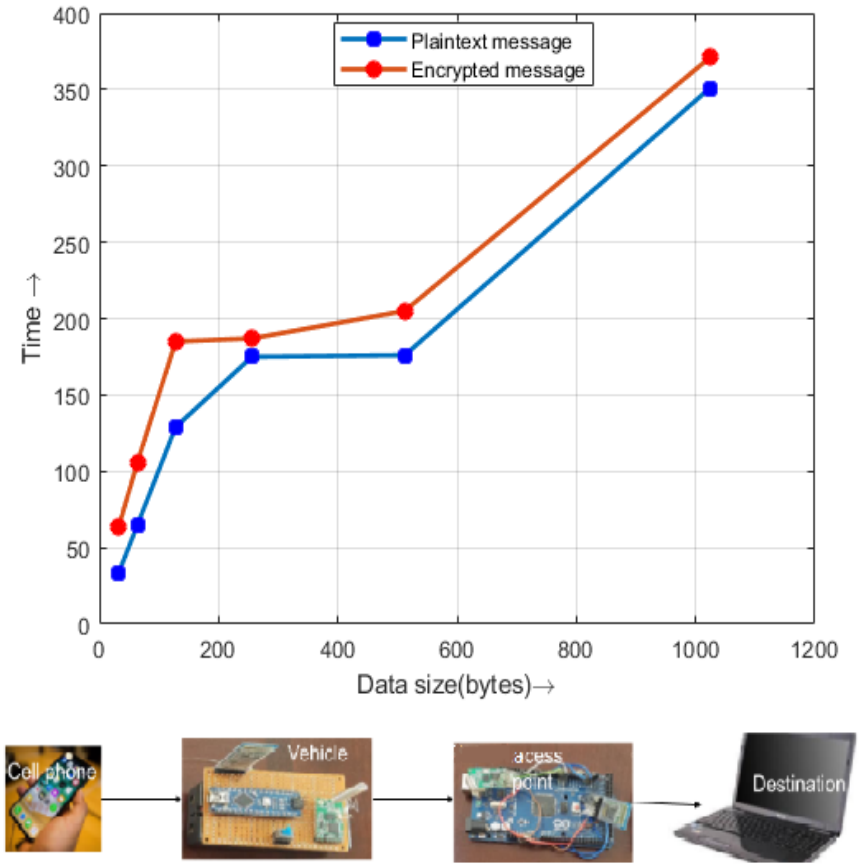

Figure 5.4: End-to-end delay (milliseconds) for different data packet sizes.

successfully 100% delivered when the source and destination were separated by 25m and 50m, and then its PDR decreased to 82% at 75m and below 40% at 100m. When the distance increased to 125m, none of the data packets that were sent reached the destination (PDR=0%) due to the loss of connection - transmission range, which in our scheme is only up to 100m.

5.1.4 Comparison with Related Work

A number of VANET hardware implementation schemes have been proposed in the literature [7, 83, 93, 95]. Anadu et al. [7] proposed a collision detection system to improve traffic efficiency. Their model consists of a microcontroller, an RGB LCD screen for data display, a Mpu6050 accelerometer, a transmitter on one car and a receiver on the other one. Various parameters of the vehicles such as position and speed are used to create messages for collision detection. The above schemes [7, 83,

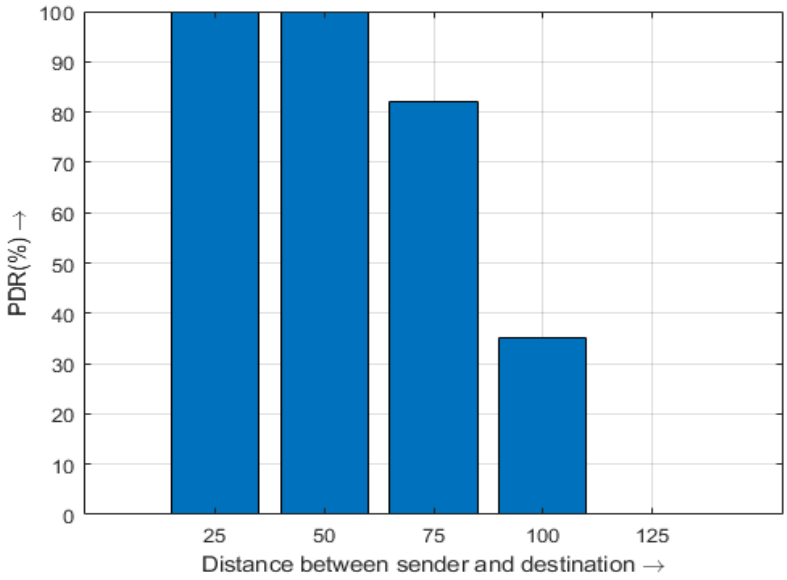

Figure 5.5: Packet delivery ratio vs distance between sender and destination.

93, 95] provide collision detection and avoidance service only and do not provide other services. In addition, they used sensors in their schemes, and not all events can be identified by sensors. Our scheme provides different services and we rely on humans instead of sensors for detecting events.

As we noticed, many of them use sensors to detect events, and thus in some cases the data that come from sensors may not be sufficient to take a decision or it may lead to a wrong decision. For example, the sensor of the camera cannot capture the obstacle on the other side of the road. In addition, not all events can be identified by sensors. Our scheme provides different services and we use humans instead of sensors for detecting events. Our goal is to combine smartphones and OBU of vehicles so that our scheme gets benefits from smartphone applications (e.g., GPS, AES Encrypter/Decrypter, etc) and integrated with OBU of vehicles.

5.1.5 Summary

In this chapter, we presented a hardware implementation of On-Board for VANET. Our scheme consists of cell phones, vehicles, access points, and destination. The driver in our scheme can send messages to nearby vehicles, *AP*s, and destination. We evaluated our scheme with respect to end-to-end delay and packet delivery ratio. We have observed that the transmission times grow with the size of the data packet transmitted. It also shows the packet delivery ratio decreases when the distance between the source and the destination increases.

5.2 A Hardware Implementation of AODV Routing Protocol in VANET: Design and Experimental Evaluation

In VANET, On-Board Units (OBUs) in each vehicle can collect information about accidents, road conditions, traffic updates, etc., and disseminate it to other drivers that can help other drivers. However, most of the proposed VANET schemes were based on simulation models; only a few of them have built hardware implementations. To our knowledge, there is no hardware implementation for routing protocols in VANET. In this chapter, we present a hardware implementation of Ad hoc On-Demand Distance Vector (AODV) routing protocol for VANETs. Our implementation allows messages to be disseminated to and from vehicles and roadside units (RSUs). This implementation is simple and incurs low cost. Experimental results show that the implemented method can transmit and receive messages between the source and destination in a timely manner.

5.2.1 Introduction

VANETs allow vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. The Onboard Units (OBUs) located in the vehicles allow the vehicles to collect data from their environment, process, and send information to other vehicles and/or RSUs through wireless communication. However, not all existing vehicles are equipped with the needed hardware (sensors, transceivers), and some rural areas have no network coverage. Therefore, there is a need to use cheap and simple hardware technology for vehicles onboard and roadside units. Several hardware implementations of VANET have been developed in the literature [7, 83, 93, 95]. These implementations were proposed for collision detection to improve traffic safety, assuming single-hop transmission between vehicles. However, to our knowledge, none implemented a routing protocol in hardware to enable multi-hop V2V communications.

Routing protocol plays a vital role in extending the range of awareness in VANETs [31, 18]. Routing aims to establish routes from one node to other nodes, forward the packets in the network, and maintain and update the routes. There are two main types of routing protocols: proactive and reactive routing protocols. Reactive protocols are based on an on-demand mechanism in which each node in the network discovers or maintains a route when needed. This makes it suitable for VANETs due to the high mobility and vehicles density such that network traffic and bandwidth are reduced [51]. AODV is one of the well-known reactive routing protocols [81]. In AODV, when a source node wants to send a data packet to a destination node, it first checks the available routes in its routing table. Then, if the route information is already in the table, the packet is sent to its destination. Otherwise, the source node broadcasts a route discovery request to all neighboring nodes. Then, this process continues until the request packet gets an intermediate node with a route to the destination or the destination node itself. When the route reply packet arrives from the destination or the intermediate node, the nodes forward it along the established reverse path and store the forward route entry in their routing table. Then the

packet is forwarded to the destination. The works in [34, 88, 4] used simulation models for implementing and testing AODV in VANET. However, to our knowledge, no hardware implementations for the AODV protocol in VANET have been proposed. In this chapter, we present a hardware implementation of AODV for VANETs using less expensive, but efficient devices.

Our developed hardware enables both V2V and/or V2I communications. Based on using V2V communication, vehicles can wirelessly communicate while moving along the road. This the implementation uses an Arduino Nano module, an Arduino MEGA module, an RF ZigBee module, a Bluetooth module, and a mobile phone with Android applications. The Arduino module plays a central role in our implementation; it receives the alert messages (Arduino connected to an Android device through Bluetooth) from the driver's phone via the Bluetooth module, and then sends the message using the RF ZigBee module to the intended vehicle. The warning messages can be delivered to the intended user using V2V and/or V2I communication. As a result, the packet delay is reduced, and transportation efficiency improves. In addition, we note that our implementation can extend network connectivity in areas with the poor network coverage. The performance evaluation results show that the destination successfully received the data messages on time [71].

The rest of the chapter is organized as follows. In Section 5.2.2, we present our proposed model and describe the hardware implementation. We present the results and analysis in Section 5.2.5. In Section 5.2.6, we discuss some related work. Finally, Section 5.2.7 summarizes the chapter.

5.2.2 Proposed Model

In this section, we present our system model and describe the proposed hardware implementation.

5.2.2.1 System Model

Figure. 5.6 shows the system model of our implementation. It consists of OBU in every vehicle, RSU, cell phone-LCD, and destination. The function of each of these entities is described as follows:

- **The On-Board Unit (OBU):**
  The OBU contains the following hardware components: the Arduino Microcontroller, the RF ZigBee module, the Bluetooth module, and an ultrasonic sensor. Figure 5.7 shows the OBU components.

  - **The Arduino Microcontroller:**
    The Arduino boards are an open-source electronics platform containing both hardware and software. Our implementation uses Arduino Nano and Arduino MEGA2560 microcontrollers [9] because they are cost-efficient, handy, and highly reliable. The Arduino MEGA2560 has 54 digital input and output pins that run on 5V and has 8KB of memory. The Arduino

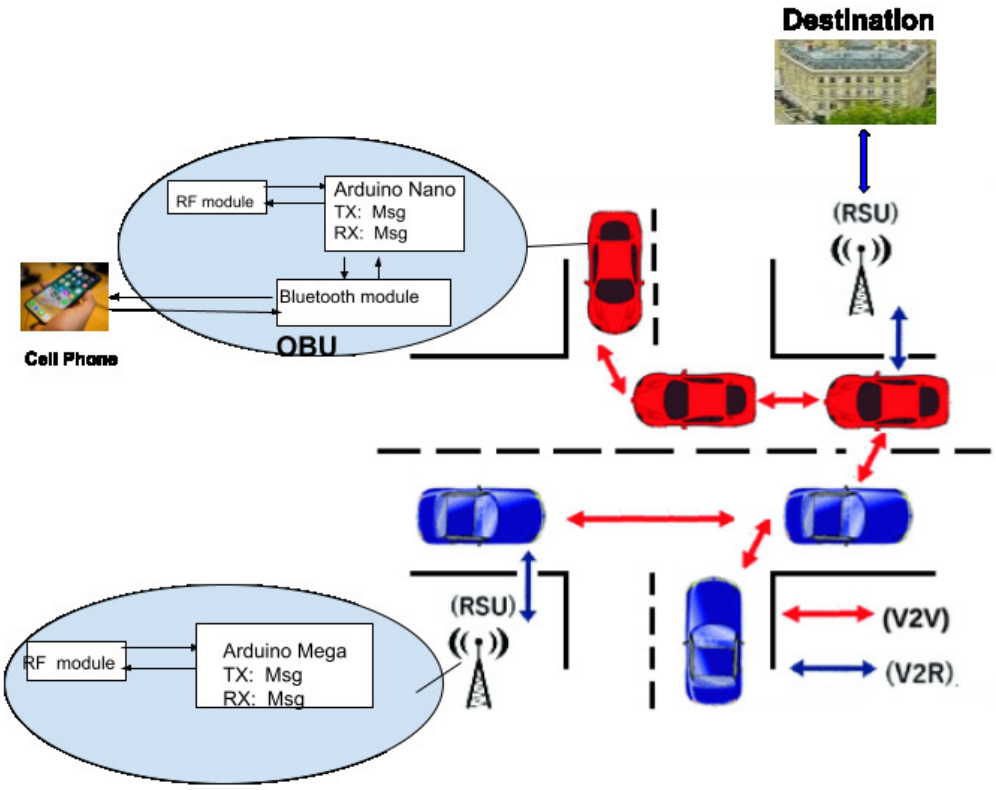

Figure 5.6: System Model for VANETs.

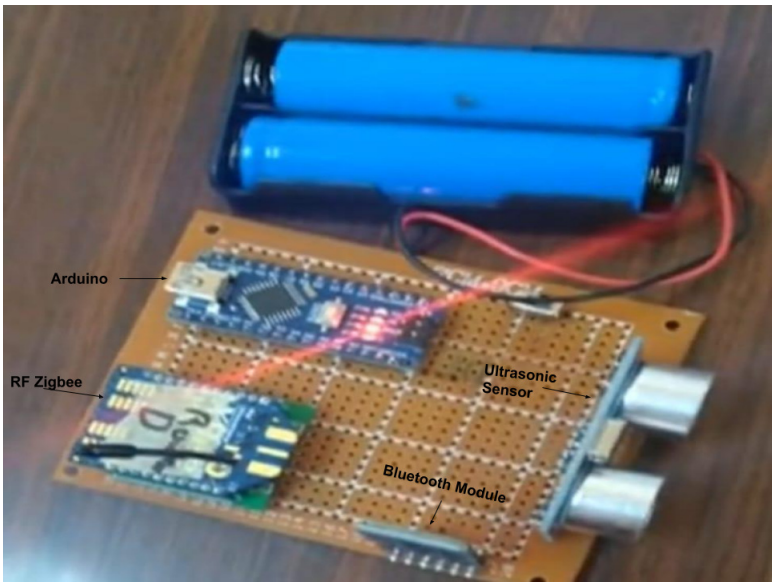

Figure 5.7: The implementation of the OBU part.

Nano is based on the ATmega328, powered by a 12V supply, and consists of 14 digital input-output pins.

- **The RF ZigBee Module:**
  For communication, the XBee Series 2 module is used. This module is based on ZigBee technology, an IEEE 802.15.4 standard protocol [105], [3], which has a mesh networking functionality to allow communication between the XBee devices. Each XBee device can act as a coordinator, router, and/or end device. The Coordinator establishes the network service and also selects the operating channel.

  The transmission range of the XBee Series 2 Pro module is 140 feet for indoor and 4000 feet for outdoor scenarios. The RF ZigBee module is used to enable wireless communication between Arduino devices. The Coordinator, routers, and end devices are the main devices used in a ZigBee network. The coordinator is responsible for creating a network; therefore, every network must have at least one coordinator. In addition to its own tasks, the coordinator does everything a router does, such as choosing a Personal Area Network (PAN) ID, security mode, etc. The router serves as a transmitter and/or receiver and is responsible for routing data. Finally, the end device sends and/or receives data to/from the routers or the coordinator and can send and/or receive information from other end devices [80].

  Figure. 5.8 shows the network topology with XBee S2B wireless modules, configured with Configuration and Test Utility (XCTU) software. Each ZigBee network creates a virtual network and labels it with a 16-bit PAN address. All the communicating radios must be turned on to the same frequency to receive messages. When the ZigBee coordinator picks a network PAN address, it checks all the available channels and selects one channel for that network's connectivity. All the radios in that network must use the selected channel. By default, XBee radios automatically handle channel selection. Figure. 5.8 also shows the Link Quality Indicator (LQI). The LQI describes the connection strength between XBee devices. The range of values to determine the link quality is between (0 and 255). For example, the value 255 refers to the best connection between the communicating XBee devices.

- **The Bluetooth Module:**
  The Bluetooth module HC-05 is used in our implementation. This module allows all serial-enabled devices to communicate with each other wirelessly. We create an Arduino-Bluetooth interface for exchanging messages between Arduino and the cell phone of a driver.

- **Ultrasonic Sensor:**
  We use ultrasonic sensors to determine the distance between vehicles and their destination. Ultrasonic sensors provide high-accuracy distance detection and stable readings.

- **Cell Phone:**
  We use the driver's cell phone as an LCD screen through a serial Bluetooth

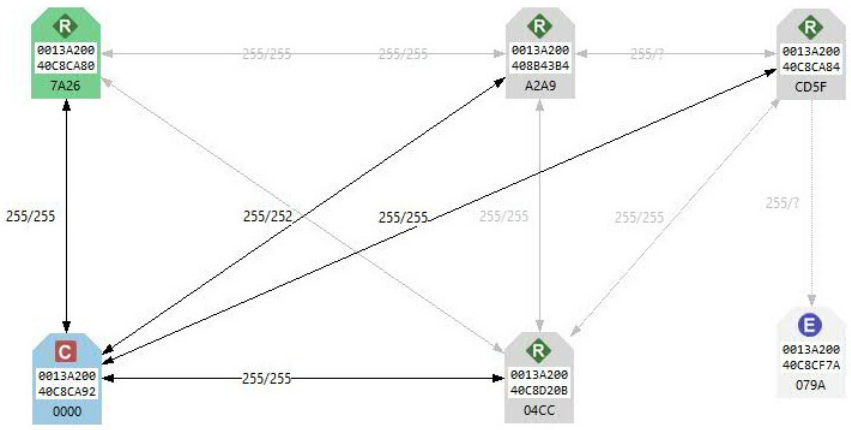

Figure 5.8: ZigBee Network Topology.

application. This application allows the driver to write and send messages to the Arduino through the Bluetooth module. Also, this application can display messages that are received from Arduino.

- **Road Side Unit (RSU):**
  The RSUs are fixed units deployed along the roadside (e.g., gas stations, road intersections, etc.). They collect the messages sent by the vehicles and forward them to other vehicles within their transmission range, or to the destination.

- **Destination:**
  The destination can be the department of motor vehicles, hospital, police station, etc.

5.2.3 Architecture of the Proposed Hardware Implementation

In the following section, we describe the implemented hardware architecture in detail. The Arduino microcontrollers are the main data processing units in our system. In our implementation, we define two main types of nodes, the transmitter, and the receiver. At a transmitter node, the vehicle transmits the data message (e.g., observed accident on the road) using a Bluetooth module to the Arduino unit. Then, the message is wirelessly transmitted using an RF Zigbee module to an RF Zigbee module at the receiving node. Then, the Arduino at the receiver sends the received message to the driver in the receiving vehicle via Bluetooth. The received message appears on the driver's phone, which enables the driver to take action.

In our implementation, the information collected by each driver in the V2V network (e.g., observed accident) is sent to a nearby RSU through the OBUs using the AODV routing protocol. Figure 5.9 shows the flow chart of the message forwarding process in our proposed system. Under AODV, when a source node wants to send a data packet to a destination node, it checks its routing table to look for a candidate route. If a route exists, the packet is forwarded to the next hop along the found path

to the destination, if there is no available route in the table, the source node starts a route discovery process by sending a route request message [81].

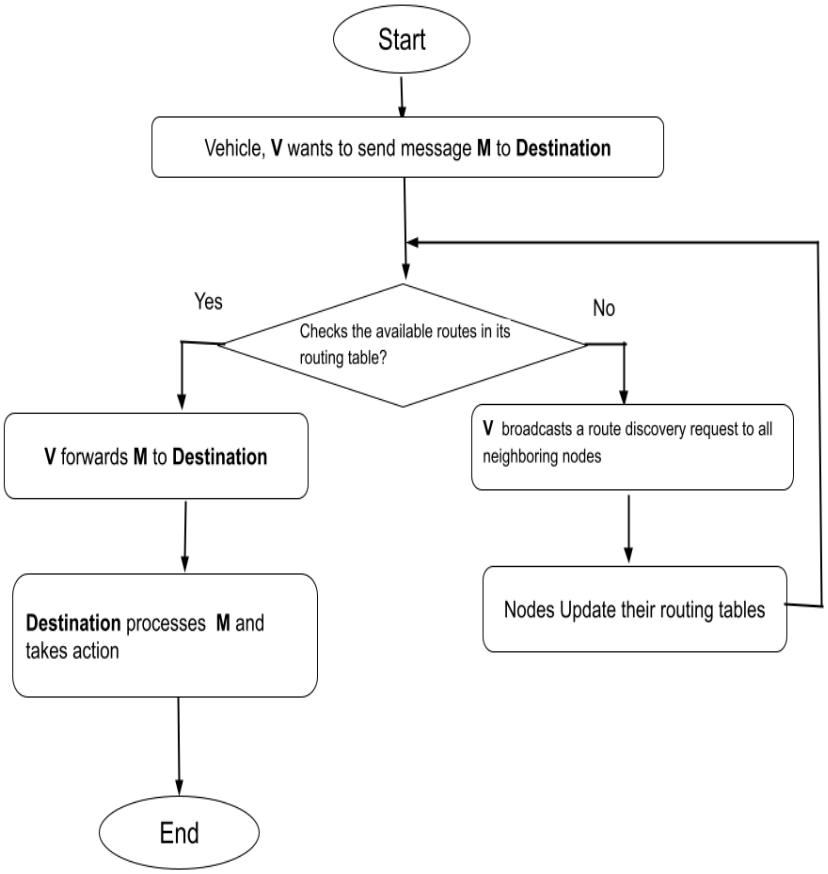

Figure 5.9: Flow chart of the message transmission in our proposed method.

In particular, the source node sends the control packets to the intermediate nodes in its routing table. This process continues until the request packet gets to either an intermediate node with a route to the destination or the destination node itself. The route request packet contains the IP address of the source node, the current sequence number, the hop count, and the IP address of the destination node. When an intermediate node forwards the route request packet, it records the neighbor's address in its routing table from which the first copy of the packet was received. This recorded information is then used to construct the reverse path for the route reply packet. Finally, when the route reply packet arrives from the destination or the intermediate node, the nodes forward it along the established reverse path and store the forwarded route entry in their routing table.

5.2.4 Hardware Implementation and Discussion

We implemented OBU hardware for vehicles and RSUs. We assembled 5 OBUs (vehicles), 1 OBU (RSU), and one destination (PC).

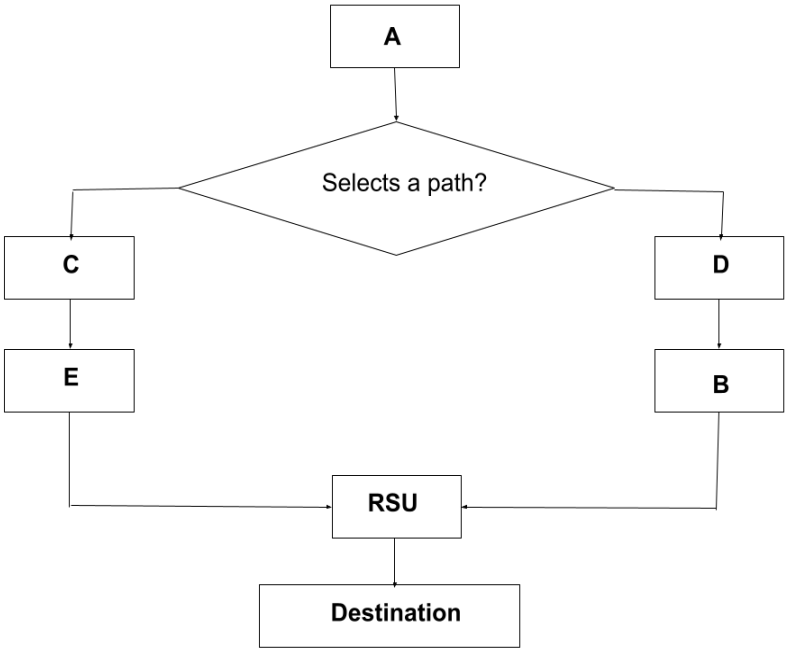

Figure 5.10: Sending messages to Destination.

Figure 5.10 shows the process of sending data messages between vehicles and destinations in our implementation. Vehicle A (the coordinator) has two available paths to send data messages to the destination node using the AODV protocol. The data messages are sent using the shortest path. Specifically, the coordinator sends the data messages either through the path A-C-E-RSU-Destination or A-D-B-RSU-Destination. Note that, the data packets are sent wirelessly from the coordinator to the RSU, which are automatically forwarded serially to the destination. The received data messages can be viewed on the PC (Destination) screen using open terminal software (putty). The received data at the destination can be forwarded to the cell phone, Arduino nano, and Arduino mega, which act as the processing units as they control the data flow from the RF ZigBee module to the HC-05 module and vice versa.

5.2.5 Result and Analysis

In this section, we present the performance evaluation results of our implemented hardware system using a realistic testbed.

Our main performance metric is network throughput (the sum of successfully received bits by destination). Figure. 5.11 illustrates network throughput versus the transmission time from the coordinator to the destination. We used five OBUs, one RSU, and one destination. One of the 5 OBUs is a coordinator, and the other OBUs are distributed randomly in different locations. The packet size used to compute throughput is 32 bytes, and the number of packets sent from the coordinator to the destination varies between 9 and 100. This figure shows that network throughput

decreases as the time elapses from 0 to 9000 milliseconds. This happens because the network data load increases with different vehicle locations.

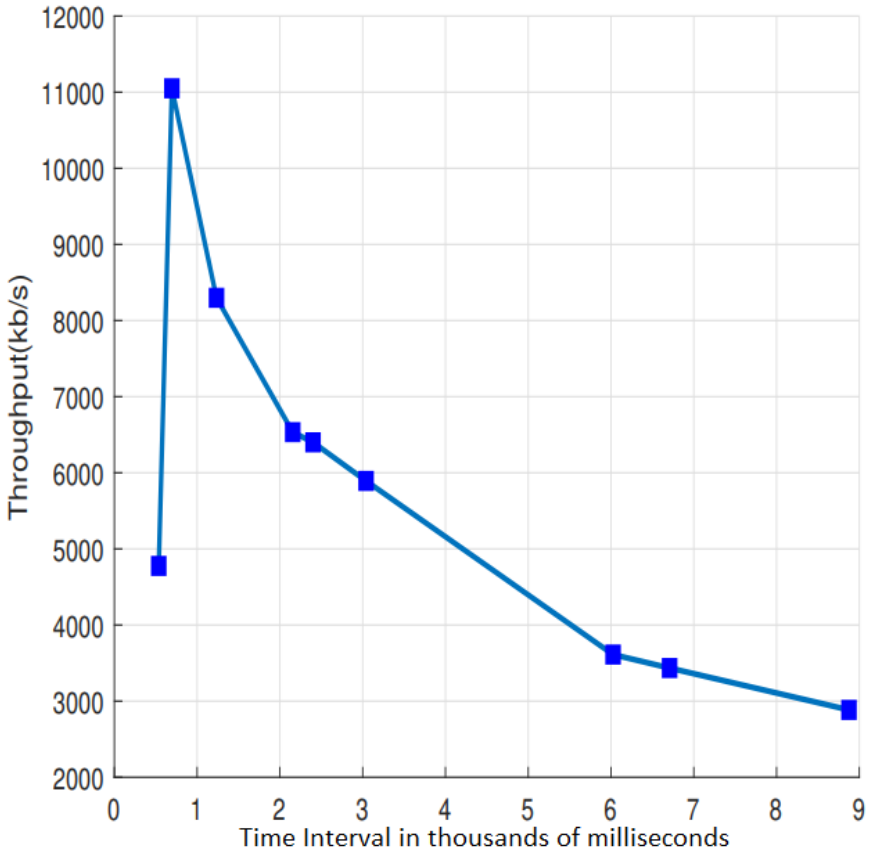

Figure 5.11: Throughput versus Time Interval.

Figure. 5.12 shows the delay as a function of the distance from the coordinator to the destination. The distance varies with the location of routers (vehicles); hence, the selected path may change. The packet size of 32 bytes transferred from the coordinator to the destination at distances ranging from 5 and 50 meters. The figure shows that as the distance increases, the delay increases because the protocol makes the decision based on direction, distance, and signal strength. Our hardware module helps deliver the data packets faster by avoiding re-transmissions using the AODV protocol.

5.2.6 Related Work

The schemes proposed in [34, 88, 4, 74, 90, 99] used simulation to investigate the performance of the AODV protocol in VANET and evaluate them with respect to packet loss/delay and throughput. In [74], the authors compared the performance of AODV [81], AOMDV [62], GSR [28], DSR [32], and OLSR [46] protocols in terms of average delay and packet delivery ratio through simulation. They showed that

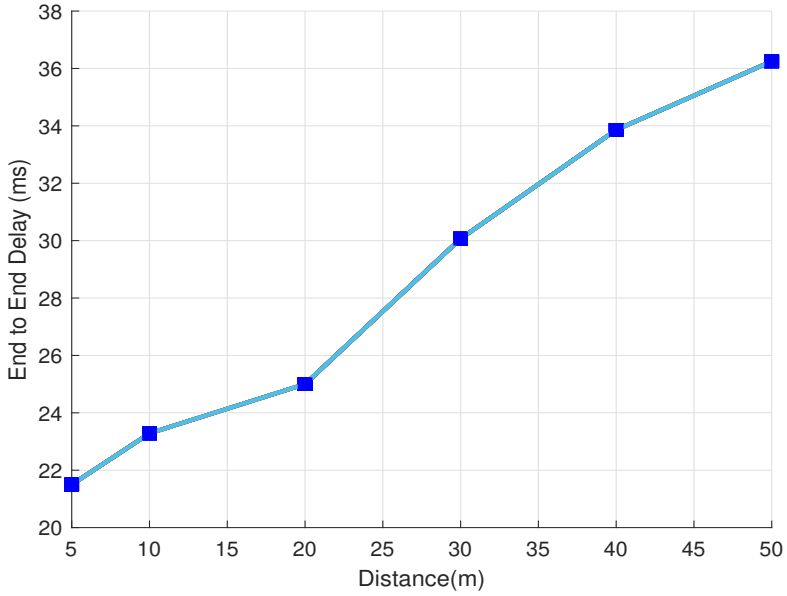

Figure 5.12: The End to End Delay vs distance from coordinator to destination.

the AODV protocol is more suitable for VANETs. Authors in [99] analyzed the performance of AODV and DSR routing protocols in the Zigbee network. Throughput and delay metrics were used to analyze the performance of these routing protocols. Their results showed that the AODV routing protocol is more efficient than the DSR routing protocol.

We note that none of the previous efforts provided a hardware implementation of the AODV protocol in VANETs. In this chapter, we presented a hardware implementation for the AODV protocol for VANETs. Our implementation combines cell phones and OBUs, it can benefit from cell phone applications and services (e.g., GPS, symmetric key Matlab algorithm Encrypter/Decrypter, etc.). In addition, the developed OBUs can be distributed in areas with low coverage (e.g., rural areas) to provide better connectivity for VANETs.

5.2.7 Summary

In this chapter, we presented an on-board hardware implementation of the AODV routing protocol for VANET. We have used less expensive, but efficient devices to implement the OBU. Experimental results show that the implemented method can transmit and receive messages to a destination in a timely manner.

Chapter 6 Privacy-preserving Approach for Collection and Dissemination of Messages in VANETs

6.1 Introduction

In VANET, malicious vehicles may send false messages (e.g., regarding an accident) to other vehicles to benefit from using a certain road. To prevent such attacks, researchers have designed efficient schemes for secure message dissemination and privacy-preserving authentication in VANETs. In some schemes [47, 56], vehicles authenticate each other and exchange messages about events. Other schemes [29, 115, 67], use RSUs for authenticating and disseminating messages transmitted by vehicles within the $RSU's$ region. If vehicles get a notification about events promptly, accidents and traffic jams, etc., could be prevented, especially on dangerous roads with a low vehicle density. If the number of vehicles is limited in an area, the packets may not get sent on time because of the non-availability of a sufficient number of vehicles to collect and disseminate information; moreover, limited roadside infrastructure would worsen the problem. This chapter addresses this issue by proposing a scheme to solve this problem.

In our scheme, we use fixed OBUs. These fixed OBUs are devices distributed in fixed locations with a low density of vehicles present in a region. One of these fixed OBUs works as the Group Leader (GL). The GL is in charge of collecting messages sent by fixed OBUs and vehicles within its transmission range, authenticating them, aggregating them, and sending them to the Department of motor vehicles (DMV). In addition, the fixed OBUs ($FOBU$) can be distributed within regions with RSU to help collect data and increase the privacy of vehicles by changing $FOBU$s pseudonyms with the pseudonyms of vehicles present in the RSU region [70].

A summary of the main contributions of this chapter is as follows:

- We develop an improved safety message collection with an increased privacy-preserving scheme for VANETs. We propose installing fixed OBUs along the roadside of dangerous roads (i.e., roads that are likely to have more ice, accidents, etc., but have a low density of vehicles and roadside infrastructure) to help collect data about the surrounding environment. This would help vehicles to be notified about the phenomena on such roads (such as ice, accidents, etc.).

- We use a digital signature to guarantee the integrity and authenticity of the transmitted packets.

- We use pseudo IDs to ensure the privacy of vehicles. To enhance the privacy of vehicles, the proposed scheme allows vehicles to change their pseudo IDs in all traffic conditions. Therefore, regardless of whether the number of vehicles is low in the RSU/GL region, it would be hard for an attacker to know the actual number of vehicles in the RSU/GL area.

The remainder of the chapter is structured as follows: First, in Section 6.2, we present our proposed approach. Then, in Section 6.3, we describe the security and performance analysis of our approach. Next, we discuss related works in Section 6.4. Finally, we summarize the contribution in Section 6.5.

6.2 The Proposed Approach

This section describes the system model and proposed method in detail. Table 6.1 shows the acronyms used in this chapter.

Table 6.1: Notations used in this chapter.

Notation	Description
ID_A	Identity of Entity A
PID_A	Pseudo-Identity of Entity A
M	A Message
V_a	Vehicle a
$FOBU$	Fixed OBU
ts	Timestamp
PR_A	Private Key of Entity A
PU_A	Public Key of Entity A
K	Shared Secret Key
$SIG_A(M)$	Signature of M Signed with Private Key PR_A of A
$H()$	Hash Function
$E(M, K)$	Encryption of M with Key K
RSU	Roadside Unit
GL	Group Leader
DMV	Department of Motor Vehicles
$Cert_v$	Vehicle v Certificate
$Cert_{RSU}$	RSU Certificate

6.2.1 System Model

The system model of our approach is depicted in Figure 6.1. The system model consists of the DMV, RSUs, several vehicles, $FOBU$s, and GLs. The functions of each of these entities, as well as the assumptions we make about these entities, are described next.

- **DMV:** The DMV (Department of Motor Vehicles) is in charge of the registration of all vehicles and assumed to be trusted. The DMV generates its private and public keys (PR_{DMV}, PU_{DMV}) and distributes a PU_{DMV} to all vehicles, $FOBU$s, and RSUs securely. Moreover, the DMV generates pseudo IDs (PID_v) for every vehicle, certificates related to every pseudo ID of a vehicle ($Cert_v$), where $Cert_v = E((PID_v, PU_v, ts), PR_{DMV})$, and certificates of RSUs ($Cert_{RSU}$) where $Cert_{RSU} = E((ID_{RSU}, PU_{RSU}, ts), PR_{DMV})$. The DMV also generates and shares a symmetric key between itself and each RSU so they can encrypt messages using that key and send between them.

- **RSU:** The RSUs (Roadside Units) are fixed entities distributed along the roadside. The RSUs are connected to the DMV through the Internet. In our approach, an RSU collects the messages received from the vehicles and/or $FOBU$s within its transmission range, authenticates them, aggregates them, and sends them to DMV, as well as to vehicles in its region. The DMV is in charge of further aggregating and disseminating these messages to vehicles in appropriate regions through the respective RSUs.

- **Vehicle:** It is assumed that each vehicle is equipped with OBUs for computation purposes and communication with the RSUs and other vehicles. The OBU stores the vehicle's private and public keys (PR_v, PU_v), its pseudo-IDs (PID_v) along with their certificates issued by the DMV, and the DMV public key (PU_{DMV}).

- **Fixed OBU ($FOBU$):** Fixed OBUs are devices like OBUs that do not move and are installed along dangerous roads (i.e., roads in which vehicle density is low and accidents, icy road conditions, etc., happen frequently), and within the vicinity of RSU so they can collect information and forward it to the RSUs through other OBUs. The fixed OBUs help in collecting safety messages in such areas and forwarding them to the RSUs so the RSUs can aggregate and forward them further to the DMV and vehicles in their region.

- **Group Leader (GL):** After installing the $FOBU$s in a region, if there is no RSUs in that region, one or more of the $FOBU$s are designated as GLs in that region, so that the GLs can work as RSUs.

The following section provides a comprehensive description of the proposed approach.

6.2.2 The Proposed Method

In our proposed method, when vehicles/$FOBU$s transmit messages to the RSU/GLs, the RSU/GLs verify their authenticity and integrity before disseminating them to DMV or other vehicles in its region. To improve the safety message collection, especially in the low-traffic environment, we use $FOBU$s. As shown in Figure 6.1-a, these $FOBU$s are installed in regions with a low density of vehicles, and in each region, one of them works as GL. Moreover, the $FOBU$s can be installed within regions that have RSUs (Figure 6.1-b) so the $FOBU$s can collect information and forward them to the nearby RSU for further dissemination. In addition, we use them to increase the vehicles' privacy in the RSU/GL region.

We assume the following in our work:

1. The clocks of DMV, RSUs, vehicles, and $FOBU$s are loosely synchronized. Time received from the Global Positioning System (GPS) can be used for this purpose.

2. Certificates generated by the DMV are used for the authentication of RSUs, vehicles, and $FOBU$s;

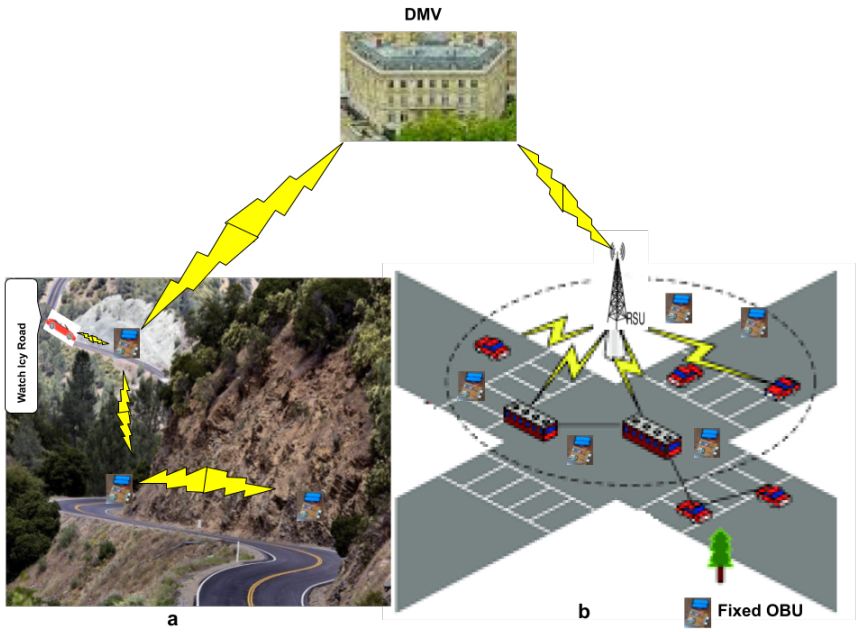

Figure 6.1: System model for VANETs.

3. The problem of identifying the malicious RSUs or vehicles is not addressed in this chapter. In our work, we assume that any of the existing previously proposed malicious detection methods can be used to determine malicious vehicles.

4. If the RSU/GL is not in the transmission range of v, v can adopt any of the existing routing protocols in literature to transmit messages to nearby RSU/GL through other vehicles/$FOBUs$ (e.g., [31, 48]).

Our scheme is described in detail next.

When a vehicle v enters an RSU/GL's region: Each RSU/GL continually broadcasts its $Cert_{RSU/GL}$. When a vehicle v enters an RSU/GL region, v checks RSU/GL's certificate in its CRL to determine if it is revoked. If the RSU/GL's certificate is not revoked, then v sends a join packet M to the RSU/GL. This packet M contains its PID_v, along with its certificate $Cert_v$, and a time-stamp (ts). When RSU/GL receives the message, the RSU/GL uses the ts to check the message freshness. Next, the RSU/GL retrieves pseudo ID PID_v, and the vehicle public key PU_v from $Cert_v$, and checks the vehicle's certificate in its CRL to see if it is revoked. If it is not revoked, the RSU/GL transmits an "accept" packet to v. This packet consists of a shared key K used for securing v and RSU/GL communications, group key GK used to broadcast the CRLs and other messages to vehicles in the RSU/GL's region, and a time-stamp ts, all encrypted using PU_v of v; the certificate of the RSU/GL, namely, $Cert_{RSU/GL}$, and the signature of the RSU/GL $SIG_{RSU/GL}$ are attached to the message as follows: $M_1 = (RSU/GL, PID_v, (E("Accept", K, GK, ts), PU_v), Cert_{RSU/GL}, SIG_{RSU/GL})$, where $SIG_{RSU/GL} = E(H("Accept", K, GK, ts), PR_{RSU/GL})$. Note that, an RSU/GL can use GK to encrypt packets and broadcast them to the

vehicles located within its region. Algorithm 9 contains the algorithm illustrating the joining process of a vehicle v when v enters the region of an RSU/GL.

Algorithm 9 When a vehicle v enters the RSU/GL region

`When a vehicle` v `enters an` RSU/GL `region:`
Verifies $Cert_{RSU/GL}$ received in the broadcasted message using PU_{DMV};
Retrieves $PU_{RSU/GL}$ from the $Cert_{RSU/GL}$;
Computes $M_1 = (\text{"}Join\text{"}, ts)$;
Encrypts M_1 using $PU_{RSU/GL}$;
Sends $M_1' = (PID_v, RSU/GL,$
$E(M_1, PU_{RSU/GL}), Cert_v, SIG_v)$ to the
RSU/GL, where $SIG_v = E(H(M_1), PR_v)$ and
$Cert_v$ is the certificate corresponding to PID_v.

`When the` RSU/GL `receives` M_1' `from` v`:`
Decrypts M_1' using $PR_{RSU/GL}$;
Verifies $Cert_v$ using PU_{DMV};
Retrieves PU_v from $Cert_v$;
Verifies SIG_v using PU_v;
If verification succeeds {
    Computes $M_2 = (\text{"}Accept\text{"}, K, GK, ts)$;
    // M_2 contains the acceptance message.
    // K is the shared symmetric key used between
    // RSU/GL and v and GK is the group Key;
    Encrypts M_2 using PU_v of v;
    Sends $M_2' = (RSU/GL, PID_v, E(M_2, PU_v),$
    $Cert_{RSU/GL}, SIG_{RSU/GL})$ to v, where
    $SIG_{RSU/GL} = E(H(M_2), PR_{RSU/GL})$};
Else { Discards $M2$; }

`When` v `receives` M_2' `from` RSU/GL`:`
Decrypts M_2' using PR_v of v;
Verifies $SIG_{RSU/GL}$ using $PU_{RSU/GL}$;
If verification succeeds {
    Stores (M_2);}
Else { Discards $M2$. }

Next, we illustrate sending messages by a vehicle/$FOBU$ to RSU/GL.

When a vehicle v or an $FOBU$ wishes to transmit a message M to its RSU/GL: When $v/FOBU$ wishes to transmit a message M regarding a certain event to its RSU/GL, $v/FOBU$ encrypts and signs M and sends M_1 to the RSU/GL, where $M_1 = (PID_{v/FOBU}, GL, E((M, ts), K), SIG_v)$; $PID_{v/FOBU}$ is the pseudo ID of v, ts is the time-stamp, and K is the shared key established between v and RSU/GL in Algorithm 9.

Upon receiving M_1, RSU/GL decrypts the packet using the shared key K and utilizes the time-stamp ts to check the message freshness. Next, the RSU/GL verifies the message authenticity and integrity using the $v/FOBU$ signature $SIG_{v/FOBU}$. After that, it aggregates the received packets from all other vehicles located in its region. Then, it sends the aggregated information to the DMV. The DMV can disseminate the messages to the vehicles in appropriate regions. Note that the purpose of using $FOBU$s is to sense events from surrounding regions, especially in regions with low vehicle density. Therefore, the GL can inform DMV and vehicles within its transmission range about these events to take appropriate actions. Algorithm 10 shows the message collection and dissemination process.

Algorithm 10 Vehicle v or $FOBU$ sending a message M to its RSU/GL

`When a vehicle` v `or an` $FOBU$ `observes an event and wants to send a message` M `about the event to` RSU/GL`:`
  Computes $M_1 = (PID_{v/FOBU}, RSU/GL,$
  $E((M, ts), K), SIG_{v/FOBU})$;
  Sends M_1 to RSU/GL;
  // K is a shared key established in Algorithm 9.

`When the` RSU/GL `receives` M_1 `from` $v/FOBU$`:`
  Decrypts M_1 using K;
  Checks the timestamp ts;
  Verifies $SIG_{v/FOBU}$ using $PU_{v/FOBU}$;
  Aggregates M with messages received from
  other vehicles/$FOBU$s;
  Computes
  $M_2 = (RSU/GL, E((M, ts), K), SIG_{RSU/GL})$;
  Sends M_2 to DMV;
  // K is a shared key established
  // between the RSU/GL and the DMV
  // during the registration process

`When the` DMV `receives` M_2 `from` RSU/GL`:`
  Decrypts M_2 using the shared key K;
  Checks the time-stamp ts;
  Verifies $SIG_{RSU/GL}$ using $PU_{RSU/GL}$;
  Aggregates M with messages received from
  other $RSUs/GLs$;
  Disseminates messages to the proper regions;

6.2.3 Increasing privacy in our approach

To ensure privacy in VANETs, pseudonyms are used for communication between vehicles and RSUs. Many existing schemes proposed updating the pseudonyms of vehicles more frequently to decrease the chances of linking messages sent by a vehicle with two different pseudonyms. To enhance privacy and make it hard for the adversary to link pseudonyms of vehicles, many researchers proposed mixed zones schemes [39, 59, 25, 96, 78, 35, 42, 112]. In these schemes, changing the pseudonyms of vehicles happens in a specific region simultaneously to prevent attackers to link previous and current pseudonyms to a vehicle. Most of these approaches work well when the vehicles density is high but not when the density of vehicles is low. Therefore, we propose a mix-zone approach that achieves and increases the privacy of vehicles in low and high-density vehicles. In our approach, we reduce the chance of linking the pseudonyms to a vehicle using the help of $FOBU$s. The vehicles and $FOBU$s update their pseudonyms simultaneously. This makes the attacker think there are K physical vehicles in the RSU/GL region.

In our approach, the DMV generates a pseudonym set for each vehicle. v transmits a message M_p to RSU/GL for requesting a new set of pseudonyms. The packet M_p is described as follows:

$M_p = ID_{RSU/GL}, PID_{v/FOBU}, E((M, ts), K), SIG_{v/FOBU}$; ts is the time-stamp, K is the shared secret key established between RSU/GL and v, and $PID_{v/FOBU}$ is the pseudo-ID of $v/FOBU$.

When the nearby RSU/GL decodes the packet, it verifies the integrity and authenticity of M_p and sends it to DMV. Then, DMV generates pseudo-IDs for the $v/FOBU$ and certificates associated with each pseudo-ID of a vehicle ($Cert_{v/FOBU}$) and forwards the packet $M_{p_1}^{'}$ to the RSU/GL. The message $M_{p_1}^{'}$ is defined as follows:

$M_{p_1}^{'} = ID_{RSU/GL}, ID_{DMV}, E(((PID_1, ..., PID_n), Cert_{v/FOBU}, ts), PU_{v/FOBU})$, SIG_{DMV}.

The message $M_{p_1}^{'}$ includes a set of pseudonyms $PID_1, ..., PID_n$ for the vehicle $v/FOBU$. When the RSU/GL receives the message, it verifies the integrity/authenticity of $M_{p_1}^{'}$ and sends it to the requested $v/FOBU$. The packet $M_{p_2}^{'}$ is sent to the $v/FOBU$ and defined as follows:

$M_{p_2}^{'} = PID_{v/FOBU}, ID_{RSU/GL_i}, M_{p_1}^{'}, SIG_{RSU/GL}$.

Algorithm 11 shows how the vehicle v or fixed OBU $FOBU$ gets new pseudo-IDs.

In our approach, vehicles change their pseudonyms in all traffic conditions. The $FOBU$s help in increasing the privacy of vehicles. When vehicles are required to change their pseudonyms in the RSU/GL region, the vehicles, and the fixed OBUs will change their pseudonyms. This makes the adversary believe that there are physical k vehicles joining in the process of pseudonyms updates, which makes it very hard to link pseudonyms to them. This results in increasing the privacy of vehicles.

The message that each $v/FOBU$ sends to the RSU/GL in the RSU/GL region is defined as follows:

$M_{p_1}^{''} = ID_{RSU/GL}, PID1_{v/FOBU}, E((M, ts), K),\ SIG_{v/FOBU}$

Algorithm 11 When $v/FOBU$ wants to get a new set of pseudonyms

When v or $FOBU$ wants to get a new set of pseudo-IDs:
$v/FOBU$ sends a message M_p to RSU/GL asking
about getting a set of pseudonyms;
$M_p = ID_{RSU/GL}, PID_{v/FOBU}, E((M, ts), K)$,
$SIG_{v/FOBU}$;
// K is the shared secret key established in Algorithm 9.

When an RSU/GL receives M_p:
Decrypts M_p using K;
Checks the time stamp ts;
Verifies $SIG_{v/FOBU}$ using $PU_{v/FOBU}$;
$M_{p_1} = ID_{DMV}, ID_{RSU/GL}, E((M, ts), K), SIG_{RSU/GL}$;
`Send(`M_{p_1}`)` to DMV;

When a DMV receives M_{p_1} from an RSU/GL:
Decrypts M_{p_1} using K;
Checks the time stamp ts;
Verifies $SIG_{RSU/GL}$ using $PU_{RSU/GL}$;
compute($M^{'}_{p_1}$);
$M^{'}_{p_1} = ID_{RSU/GL}, ID_{DMV}, E(((PID_1, ..., PID_n)$
$, Cert_{v/FOBU}, ts), PU_{v/FOBU}), SIG_{DMV}$; // $M^{'}_{p_1}$ contains a set of pseudo-IDs $(PID_1, ..., PID_n)$ for $v/FOBU$ and certificates corresponding to every pseudo ID of a vehicle $(Certv)$
send $M^{'}_{p_1}$ to the RSU/GL;

When an RSU/GL receives $M^{'}_{p_1}$ from DMV:
Verifies SIG_{DMV} using public key PU_{DMV} of DMV;
`compute(`$M^{'}_{p_2}$`)`;
$M^{'}_{p_2} = PID_{v/FOBU}, ID_{RSU/GL_i}, M^{'}_{p_1}, ts$
$SIG_{RSU/GL}$.
send $M^{'}_{p_2}$ to $v/FOBU$;
$v/FOBU$ authenticate $M^{'}_{p_2}$;
$v/FOBU$ store its new set of pseudonyms and starts using them;

When such physical vehicles and $FOBU$s transmit packets using their new pseudonyms to the RSU/GL, the attacker would believe that there are n vehicles transmitting packets at any certain time period. Hence, the chance for the attacker to link between messages with different pseudo-IDs is low. Algorithm 12 shows our pseudonym-changing algorithm.

Algorithm 12 Changing Pseudonyms

When it is a time for vehicles/$FOBU$s to change their pseudo-IDs in the RSU/GL region:

vehicles and $FOBU$s replace their current
pseudo-IDs to a new pseudo-IDs

Each $v/FOBU$ can send messages to the RSU/GL
using their new pseudo-IDs;
// This results in n indistinguishable new
pseudo-IDs in the RSU/GL region

6.3 Performance Analysis

Ensuring Message Authentication and Non-repudiation. In our approach, mutual authentication between vehicles and RSU/GLs is done using certificates signed by DMV before VANET communication. We used digital signatures to ensure the integrity and authenticity of messages. Each message has a digital signature attached to it. When the receiver (vehicle or RSU/GL) receives the message, it uses the digital signature to authenticate and process it. An adversary can not generate signatures of other entities as the adversary does not possess their private keys. In addition, a sender cannot deny sending the message because the message is attached to its signature. Hence, the authenticity and non-repudiation of the messages are ensured.

Ensuring Confidentiality: The transmitted messages by the vehicles and $FOBU$s to the RSU/GLs are encrypted based on a shared secret key K. To modify the message, the attacker should decrypt it, modify it, and then encrypt it using the vehicle-shared secret key. The attacker does not have this key, which makes it hard to modify the message. RSU/GL disseminates the message to other vehicles by encrypting it with GK in its region. Vehicles validate the message's authenticity using $SIG_{RSU/GL}$. Hence, attackers cannot modify the messages.

Ensuring Privacy Preservation: Vehicles use pseudo IDs in any communication. Thus, the privacy of vehicles is ensured.

Ensuring Unlinkability: Unlinkability means an attacker cannot link transmitted messages by the same vehicle with 2 distinct pseudonyms. In our scheme, when vehicles are required to update their pseudo-IDs in the RSU/GL region, the $FOBU$s will change their pseudo-IDs with the vehicles. The adversary would think that there

are k vehicles joining in the pseudonym-changing process. As a result, linking the pseudonyms of vehicles is difficult for the attacker. Thus, the privacy of vehicles is increased.

Resistance to Man in a Middle Attack: In this attack, the attacker makes all the communicating entities (vehicles, $FOBU$s, and $RSUs/GLs$) think that they are directly communicating with each other, but they are not. In our approach, every message is attached with the signature $SIG(M) = E(H(M), PR)$. The attacker cannot forge a signature as the attacker does not have the sender's private key. Therefore, a man-in-the-middle attack is prevented in our proposed method.

Resistance to Replay Attack: Messages sent by Vehicles and $RSUs/GLs$ carry time stamps ts with every message to prevent the attacker from re-sending transmitted messages. When the receiver receives the message, it checks the ts to resist a replay attack (vehicles, $FOBU$s, and RSU/GL can use GPS for synchronizing clock).

Computation and Communication Overhead: In our approach, the $RSUs/GLs$ authenticate and aggregate the messages received from vehicles or $FOBU$s. There is no vehicle-to-vehicle communication. Vehicles only authenticate received packets from its RSU/GL. Therefore, the overhead of communication and computation is low for the vehicles.

A comparison of the total number of exchanged messages in our scheme and the SEMA scheme [102] is shown in Figure 6.2, in terms of the number of exchanged packets between the vehicles and the RSU/GL. For comparison, the the average number of exchanged messages between a vehicle v and RSU/GL is assumed to be two, and the number of $FOBU$s ($FOBU$s distributed within the region of an RSU/GL) is set to 10; if there are vehicles in the RSU/GL region, the RSU/GL processes the received messages from all those vehicles in scheme [102], where in our scheme, the RSU/GL processes received messages from all vehicles and $FOBU$s in its region. In our scheme, if no vehicles are present in the RSU/GL region (low-density area), the RSU/GL receives messages from the $FOBU$s, which helps vehicles get notified about such incidents in advance before they reach the area. As a result, in our scheme, the total messages received by RSU/GL is more than that in scheme [102]. As an example, if there are only 40 vehicles and 10 $FOBU$s exist in the RSU/GL region, the number of exchanged messages between RSU/GL and $vehicles/FOBU$s will be$((40 + 10) * 2 = 100$ messages) in our proposed method, where in SEMA [102], the number of exchanged packets equals to $(40 * 2 = 80)$. If there are no vehicles exist in an RSU/GL region, the number of exchanged messages between RSU/GL and $vehicles/FOBUs$ is $((0 + 10) * 2 = 20)$ in our method, and in SEMA [102], the number of exchanged messages equals to $(0 * 2 = 0)$. Consequently, the number of collected messages increases as the number of vehicles in our scheme increases. While, in SEMA [102], the total number of collected messages is lower compared to our scheme. This is primarily because, in our scheme, $FOBU$s help in message collection even if no vehicles are present in the $RSU's/GL's$ region.

Computation Overhead on *RSU/GL*: Using a Toshiba computer with an Intel i3 quad-core processor with a 2.50-GHz clock frequency, Windows 8.1 operating system, and 6 gigabytes of memory, RSA (Rivest–Shamir–Adleman) signature veri-

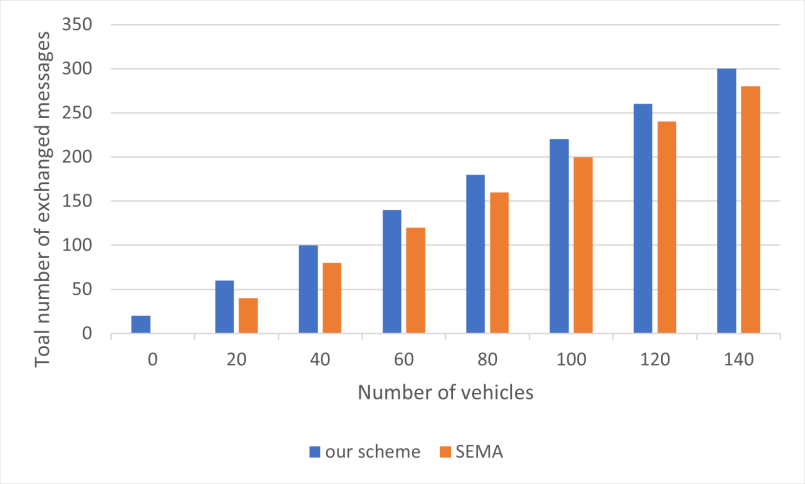

Figure 6.2: Total number of exchanged messages.

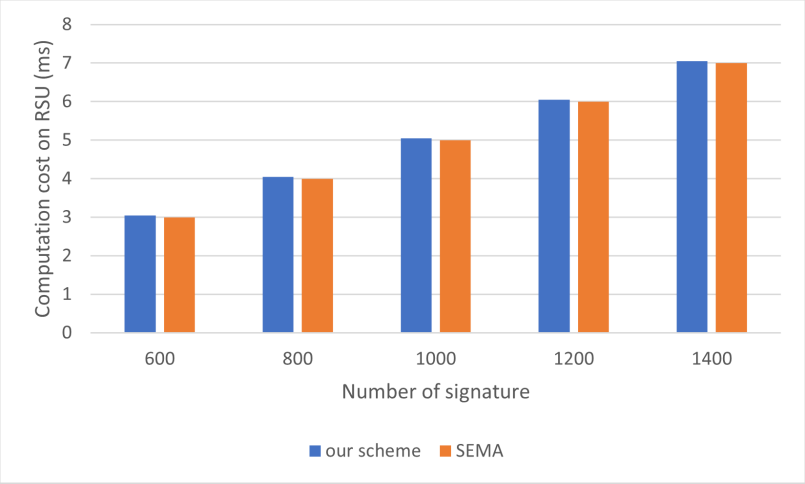

Figure 6.3: Computation time at the RSU/GL.

fication takes 0.005 ms [69]. A comparison of the computation overhead is shown in Figure 6.3 between our approach and SEMA [102] at an RSU/GL for various numbers of signatures. Our approach incurs a slightly higher overhead than SEMA [102]. This is due to the use of the $FOBU$s ($FOBUs$ increase safety message collection). As an example, when the amount of signatures exceeds 1,400, the total cost is 7.05 ms for our scheme, whereas it is 7 ms for the scheme in [102]. However, in our scheme, vehicles can benefit from getting information about regions where vehicles are sparsely present.

Anonymous set size: The number of vehicles participating in the pseudonym change procedure defines the anonymous set [54]. The confusion of the attacker increases when increasing the anonymous set size. Figure 6.4 shows the total number of vehicles that changed their pseudonyms simultaneously in the RSU/GL region. We assume the number of $FOBU$s is 10; as shown in this figure, the $FOBU$s help

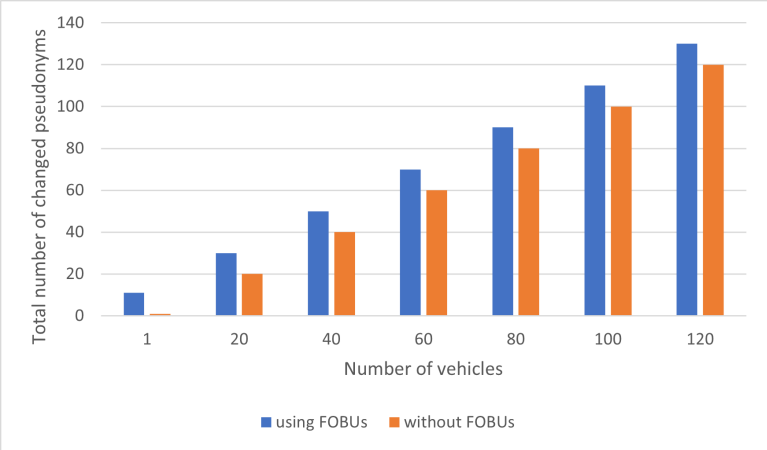

Figure 6.4: Total number of changed pseudonyms in the RSU/GL region.

in confusing the attacker with the number of physical vehicles that changed their pseudonyms even under low density of vehicles. For example, if there is one vehicle in RSU/GL region and it is time for vehicles to change their pseudonyms (RSU/GL determines the time when all the vehicles in its region need to perform the pseudonym change), the vehicle and 10 $FOBU$s will change their pseudonyms, which makes the attacker think there are 11 vehicles that changed their pseudonyms. The anonymous set increases in size as the number of vehicles and $FOBU$s participating in the pseudonym change process increases. Some related works are discussed next.

6.4 Related Works

Many privacy-preserving authentication schemes [53, 30, 72, 69, 14, 15] have been proposed for VANET. They are secured against different attacks, such as impersonation attacks, message modification attacks, etc. However, they need to consider collecting data in low-density vehicle scenarios.

Recently, several pseudonym-changing methods have been proposed in the literature. The vehicles must change their pseudo-IDs in mixed-zone areas (e.g., gas stations). The purpose of the mixed zones is to prevent the attacker from linking two different pseudonyms of the same vehicle. The authors in [20] proposed a traffic-aware pseudonym-changing scheme. In this scheme, a traffic congestion detection protocol is used to find a suitable location where a mixed zone can be established. Vehicles change their pseudo-IDs only when traffic congestion is found. However, vehicles do not change their pseudo-IDs if congestion is not found.

The scheme in [23] proposed a dynamic pseudonym-changing strategy to address privacy in VANET. Trusted Authority (TA) issued an accurate pseudonym (Rpseud) for the vehicle. Then, a vehicle communicates with the RSU and gets an initial pseudonym. A vehicle searches, continuously, for a trusted neighbor vehicle by exchanging information with the RSU; when the trusted neighbor vehicle is identified, the vehicle updates its pseudonym (Npseud). The duration current pseudonym ends

when a vehicle enters a new RSU region, and another cycle begins. Security and privacy are ensured between entities in this scheme. However, this scheme incurs computation overhead because the vehicles change their pseudonyms each time they find a new trusted neighbor vehicle. Authors in. [67] proposed a scheme that uses the mix zone strategy. In this scheme, vehicles change their pseudonyms in all traffic conditions. If the traffic density is low in the RSU region, the RSU sends a message to some vehicles and notifies them to act as k vehicles. As a result, the attacker will believe that there are n vehicles joining in the pseudonym-changing process. Authors in [16] proposed a pseudonym-changing strategy. Their scheme works in all traffic conditions (Low and High density) to prevent linkability between pseudonyms of a vehicle. In case of a low traffic density in a region, the vehicle broadcasts beacons with modified speed and location information for a particular period to deceive the adversary and his/her expectations. The drawback of this scheme is that the vehicle broadcasts beacons with real-time speed and position if there is an emergent event. This gives an attacker a chance to track the vehicle in a low-density region.

Most of the existing privacy-preserving authentication schemes [53, 30, 72, 69, 14, 15] address security and privacy issues, while other schemes [20, 23, 67, 16] use mix zone areas to increase the unlinkability of pseudonyms. To our knowledge, existing schemes did not address the collection of safety messages in low-density vehicle scenarios, and some others that use mixed zone schemes did not prevent the linkability of pseudonyms in low-density vehicle scenarios. Our scheme improved the safety message collection and enhanced privacy preservation in VANET. In our work, we use $FOBU$s distributed in areas with a low density of vehicles, especially on roads that are considered dangerous. They can also be distributed in the RSU region. The purpose of the $FOBU$s is to collect data from the surrounding regions and send it to the RSU/GL. The RSU/GL collects, authenticates, and aggregates the messages transmitted by the vehicles. Then, it forwards them to the vehicles within their transmission range and to DMV for further dissemination. Moreover, our scheme helps increase the unlinkability of pseudonyms. When it is time to change the pseudonyms of vehicles, the $FOBU$s change their pseudonyms with vehicles, which makes the attacker think there is a certain number of vehicles in the RSU/GL region. However, it is not, thus preventing the attacker from linking pseudonyms for a vehicle. Therefore, our scheme improves safety message collection and, at the same time, enhances the unlinkability of pseudonyms for a vehicle, preserving the privacy of vehicles.

6.5 Summary

In this chapter, we presented an improved safety message collection and increased privacy-preserving scheme for VANETs. We use fixed OBUs to help collect data from the surrounding environment in areas where vehicles would be sparsely present. Vehicles can be notified in advance about incidents such as accidents, icy roads, etc., in those regions to take proper action. Messages are authenticated, and the privacy of vehicles is preserved. Our scheme's communication and computation overheads are analyzed and compared with an existing scheme.

Chapter 7 Conclusion and Future work

Due to the growing interest of customers and the industry in self-driving vehicles, interest in the design and implementation of the Internet of Vehicles (IoV) has grown significantly. Based on some estimates, in the next two years, the global market for IoV will exceed $200 billion. As a result, several auto manufacturers have developed programs and platforms for connecting to IoV services, e.g., intelligent parking and collision prevention. Moreover, vehicles participating in VANETs are likely to utilize clouds to store information as well as retrieve information. In this section, we summarize the results of our dissertation and also discuss future work.

7.1 Dissertation Summary

First, we proposed a secure and distributed architecture for the vehicular cloud. This architecture is hierarchical and consists of vehicles, roadside units, regional clouds, and the central cloud. Each regional cloud covers a region (e.g., city, state) and processes the information collected from vehicles through the *RSU*s, and provides on-demand services to vehicles in its region. These regional clouds further communicate with the central cloud and exchange information between themselves to provide a wide range of services to vehicles. Our architecture also copes with *RSU* failures. In addition, we designed a scheme for changing pseudonyms that ensures the privacy of vehicles in all traffic conditions. Our scheme depends on mix zones that allow vehicles to change their pseudonyms to make it difficult for an attacker to link pseudonyms assigned to the same vehicle. Our scheme also ensures confidentiality and authentication for messages.

Second, we presented a low-overhead *RSU*-aided message authentication and dissemination scheme. In this scheme, when the overhead for collecting, authenticating, aggregating, and disseminating messages increases for an *RSU*, the *RSU* can designate some of the vehicles in its region as Group Leaders and make them share the overhead involved in authenticating, aggregating, and disseminating messages. Thus, this scheme helps the *RSU*s by reducing the computation and communication overhead related to collecting, authenticating, aggregating, and disseminating messages. We have also shown that our scheme is privacy-preserving and secure and resilient to various attacks.

Third, we present a hardware implementation of on-board units in VANET. In our implementation, we have used less expensive, but efficient devices to implement the OBU.

Fourth, we proposed an improved safety message collection and increased privacy-preserving scheme for VANETs. We use fixed *OBU*s to help collect data from the surrounding environment in areas where vehicles would be sparsely present. Vehicles can be notified in advance about incidents such as accidents, icy roads, etc., in those regions to take proper action. Messages are authenticated, and the privacy of vehicles is preserved.

7.2 Future Work

In the future, we will continue our research in the following directions. We will expand our research domain to a broad range of topics on the issues related to cloud and vehicular network security. We will also apply our current implementation in various scenarios with different attacker models to improve performance and evaluate our scheme further. In addition, we will investigate new research topics in closely related areas such as mobile cloud, sensor networks, the Internet of things (IoT), and blockchain-based distributed systems.

Bibliography

[1] H. Abid, L. T. Phuong, J. Wang, S. Lee, and S. Qaisar. V-cloud: vehicular cyber-physical systems and cloud computing. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pages 1–5. ACM, ACM, October 2011.

[2] A. Abuashour and M. Kadoch. Control overhead reduction in cluster-based VANET routing protocol. *Ad Hoc Networks*, pages 106–115, 2018.

[3] D. Ahirwar, P. Verma, and J. Daksh. Enhanced AODV routing protocol for wireless sensor network based on ZigBee. *Advances in Computer Science and Information Technology*, 84:97–102, 2012.

[4] M. Al-Shabi. Evaluation the performance of MAODV and AODV protocols in VANETs models . *International Journal of Computer Science and Security (IJCSS)*, 14(1), February 2020.

[5] S. Al-Sultan, A. Al-Bayatti, and H. Zedan. Context-aware driver behavior detection system in intelligent transportation systems. *IEEE Transactions on Vehicular Technology*, 62(9):4264–4275, November 2013.

[6] S. Al-Sultan, M. Al-Doori, A. Al-Bayatti, and H. Zedan. A comprehensive survey on vehicular Ad Hoc network . *Journal of Network and Computer Applications*, 37:380–392, 2014.

[7] D. Anadu, C. Mushagalusa, N. Alsbou, and A. Abuabed. Internet of Things: Vehicle collision detection and avoidance in a VANET environment. In *Proceedings of IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, Houston, TX, USA, May 2018. IEEE.

[8] Arduino. Arduino boards, compared. `https://core-electronics.com.au/tutorials/compare-arduino-boards.html`.

[9] Arduino. Arduino mega 2560. `https://www.arduino.cc/en/Main/arduinoBoardMega2560`.

[10] H. Arkian, R. Atani, A. Diyanat, and A. Pourkhalili. A cluster-based vehicular cloud architecture with learning-based resource management. *The Journal of Supercomputing*, 71(4):1401–1426, April 2015.

[11] N. Aung, W. Zhang, S. Dhelim, and Y. Ai. Accident prediction system based on hidden Markov model for vehicular ad-hoc network in urban environments. *Information*, 9(311), 2018.

[12] N. Aung, W. Zhang, S. Dhelim, and Y. Ai. Dynamic traffic congestion pricing system for the internet of vehicles in smart cities. *Information*, 9(149), 2020.

[13] M. Azees, P. Vijayakumar, and L. Deboarh. EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad-hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2467–2476, 2017.

[14] 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.

[15] 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 4087, 2019.

[16] L. Benarous, B. Kadri, and S. Boudjit. Alloyed pseudonym change strategy for location privacy in vanets. In *proceedings of 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6, 2020.

[17] S. Bitam, A. Mellouk, and S. Zeadally. VANET-CLOUD: A generic cloud computing model for vehicular ad hoc networks. *IEEE Wireless Communications*, 22(1):96–102, February 2015.

[18] O. Bouachir, M. Aloqaily, I. A. Ridhawi, O. Alfandi, and H. B. Salameh. Uav-assisted vehicular communication for densely crowded environments. In *proceedings of 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–4, 2020.

[19] A. Boualouache and S. Moussaoui. S2SI: A practical pseudonym changing strategy for location privacy in VANETs. In *Proceedings of International Conference on Advanced Networking Distributed Systems and Applications (INDS'14)*, pages 70–75, 2014.

[20] A. Boualouache and S. Moussaoui. TAPCS: Traffic-aware pseudonym changing strategy for vanets. *Peer-Peer Networking and Applications*, 10(4):1008–1020, 2017.

[21] A. Boualouache, S. Senouci, and S. Moussaoui. Towards an efficient pseudonym management and changing scheme for vehicular ad-hoc networks . In *Proceedings of IEEE Global Communications Conference (GLOBECOM'16)*, pages 1–7, 2016.

[22] A. Boualouache, S. Senouci, and S. Moussaoui. VLPZ: The vehicular location privacy zone. In *Proceedings of 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)*, volume 83, pages 369–376, 2016.

[23] W. Bouksani and B. Bensaber. RIN: A dynamic pseudonym change system for privacy in VANET. *Wiley Online Library*, pages 1–13, 2018.

[24] J. Bravo-Torres, E. Ordonez-Morales, M. Lopez-Nores, Y. Blanco-Fernandez, and J. Pazos-Arias. Virtualization in VANETs to support the vehicular cloud experiments with the network as a service model. In *Proceedings of Third International Conference on Future Generation Communication Technology*, pages 1–6. IEEE, 2014.

[25] L. Buttyan, T. Holczer, A. Weimerskirch, and W. Whyte. SLOW: A Practical pseudonym changing scheme for location privacy in VANETs. *IEEE Vehicular Networking Conference (VNC)*, pages 1–8, October 2009.

[26] A. Carianha, L. Barreto, and G. Lima. Improving location privacy in mix-zones for VANETs. In *Proceedings of 30th IEEE International Performance Computing and Communications Conference*, pages 1–6, 2011.

[27] M. Chaqfeh, N. Mohamed, I. Jawhar, and J. Wu. Vehicular cloud data collection for intelligent transportation systems. In *Proceedings of IEEE Smart Cloud Networks and Systems*, pages 1–6. IEEE, December 2016.

[28] T. Chen and M. Gerla. Global state routing: a new routing scheme for ad-hoc wireless networks. In *Proceedings of International Conference on Communications. Conference Record*, pages 171–175, Atlanta, GA, USA, June 1998. IEEE.

[29] H. Cheng and Y. Liu. An improved RSU-based authentication scheme for VANET. *Journal of Internet Technology*, 21(4):1137–1150, 2020.

[30] 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.

[31] K. Darabkh, M. Judeh, H. Salameh, and S. Althunibat. Mobility aware and dual phase AODV protocol with adaptive hello messages over vehicular ad hoc networks. *International Journal of Electronics and Communications*, 94:277–292, 2018.

[32] J. David, D. Maltz, and J. Broch. DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc networking*, 5(1):139–172, 2001.

[33] L. Di Jasio. *Programming 16-bit pic microcontrollers in c: learning to fly the pic 24*. Embedded Technology. Elsevier Science, Burlington, MA, 2007.

[34] B. Ding, Z. Chen, Y. Wang, and H. Yu. An improved AODV routing protocol for VANETs. In *Proceedings of International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–5, Nanjing, China, November 2011. IEEE.

[35] S. Du, H. Zhu, X. Li, K. Ota, and M. Dong. MixZone in motion: Achieving dynamically cooperative location privacy protection in delay-tolerant networks. *IEEE Transactions on Vehicular Technology*, 62(9):4565–4575, November 2013.

[36] A. Dua, N. Kumar, A. Das, and W. Susilo. Secure message communication protocol among vehicles in smart city. *IEEE Transactions on Vehicular Technology*, 67(5):4359–4373, 2018.

[37] M. Eltoweissy, S. Olariu, and M. Younis. Towards autonomous vehicular clouds. *EAI Endorsed Transactions on Mobile Communications and Applications*, 1(1):1–11, September 2011.

[38] H. Eren, S. Makinist, E. Akin, and A. Yilmaz. Estimating driving behavior by a smartphone. In *Proceedings of Intelligent Vehicles Symposium (IV)*, pages 234–239. IEEE, Jun 2018.

[39] J. Freudiger, M. F. M. Raya, P. Papadimitratos, and J. Hubaux. Mix-zones for location privacy in vehicular networks. *In Proceedings of the First International Workshop on Wireless Networking for Intelligent Transportation Systems (WinITS)*, Aug 2007.

[40] M. Gerlach and F. Guttler. Privacy in VANETs using Changing Pseudonyms - Ideal and Real. In *Proceedings of IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, pages 2521–2525, 2007.

[41] Y. Gu, Q. Wang, and S. Kamijo. Intelligent driving data recorder in smartphone using deep neural network-based speedometer and scene understanding. *IEEE Sensors*, 19(1):287–296, Jan 2019.

[42] N. Guo, L. Ma, and T. Gao. Independent mix zone for location privacy in vehicular networks. *IEEE Access. 2018*, 6:16842–16850, 2018.

[43] S. Gyawali, Y. Qian, and R. Q. Hu. A privacy-preserving misbehavior detection system in vehicular communication networks. *IEEE Transactions on Vehicular Technology*, 70(6):6147–6158, 2021.

[44] R. Hussain and H. Oh. Cooperation-aware VANET clouds: Providing secure cloud services to vehicular ad hoc networks. *Journal of Information Processing Systems*, 10(1):103–118, 2014.

[45] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh. Rethinking vehicular communications: merging VANET with cloud computing. In *Proceedings of the 4th International Conference on Cloud Computing Technology and Science*, pages 606–609. IEEE, IEEE, December 2012.

[46] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Proceedings of IEEE International Multi-Topic Conference*, pages 62–68, Lahore, Pakistan, December 2001. IEEE.

[47] H. Jo, I. Kim, and D. Lee. Reliable cooperative authentication for vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1065–1079, 2018.

[48] A. Khreishah, H. Bany Salameh, I. Khalil, and A. Gharaibeh. Renewable energy-aware joint caching and routing for green communication networks. *IEEE Systems Journal*, 12(1):768–777, 2018.

[49] Q. Kong, R. Lu, H. Zhu, A. Alamer, and X. Lin. A secure and privacy-preserving incentive framework for vehicular cloud on the road. In *Proceedings of Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, December 2016.

[50] S. Kumar, S. Gollakota, and D. Katabi. A cloud-assisted design for autonomous driving. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, pages 41–46. ACM, August 2012.

[51] S. Lachdhaf, M. Mazouzi, and M. Abid. Secured AODV routing protocol for the detection and prevention of black hole attack in VANET. *Advanced Computing: An International Journal (ACIJ)*, 9(1), January 2018.

[52] E. Lee, M. Gerla, and S. Oh. Vehicular cloud networking: architecture and design principles. *IEEE Communications Magazine*, 52(2):148–155, February 2014.

[53] X. Li, T. Liu, M. Obaidat, F. Wu, P. Vijayakumar, and N. Kumar. A lightweight privacy-preserving authentication protocol for VANETs. *IEEE Systems Journal*, 2020.

[54] X. Li, H. Zhang, Y. Ren, S. Ma, B. Luo, and J. Weng. PAPU: Pseudonym swap with provable unlinkability based on differential privacy in VANETs. *IEEE Internet of Things*, 7(12):11789–11802, 2020.

[55] K. Lim, I. Abumuhfouz, and D. Manivannan. Secure incentive-based architecture for vehicular cloud. *Springer Lecture Notes in Computer Science*, 9143:361–374, 2015.

[56] X. Lin and X. Li. Achieving efficient cooperative message authentication in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 62(7):3339–3348, 2013.

[57] X. Lin, X. Sun, P. Ho, and X. Shen. GSIS: A secure and privacy-preserving protocol for vehicular communications. *IEEE Transactions on Vehicular Technology*, 56(6):3442–3456, November 2007.

[58] Y. Liu, Y. Wang, and G. Chang. Efficient privacy-preserving dual authentication and key agreement scheme for secure V2V communications in an IoV paradigm. *IEEE Transactions on Intelligent Transportation Systems*, 18(10):2740–2749, October 2017.

[59] R. Lu, X. Lin, T. Luan, X. Liang, and X. Shen. Anonymity analysis on social spot based pseudonym changing for location privacy in VANETs. *In Proceedings of IEEE International Conference on Communications*, 2011.

[60] X. Ma, J. Zhang, X. Yin, and K. Trivedi. Design and analysis of a robust broadcast scheme for VANET safety-related services. *IEEE Transactions on Vehicular Technology*, 61(1):46–61, Jan 2012.

[61] S. Mallissery, M. Pai, R. Pai, and A. Smitha. Cloud enabled secure communication in vehicular ad-hoc networks. In *Proceedings of IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 2014.

[62] M. Marina and S. Das. Ad hoc on-demand multipath distance vector routing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(3):92–93, July 2002.

[63] MathWorks. VANET simulation in Matlab. `https://www.mathworks.com/matlabcentral/fileexchange/68829-vanet-simulation-in-matlab`. Accessed: 09-05-2020.

[64] T. Mekki, I. Jabri, A. Rachedi, and M. ben Jemaa. Vehicular cloud networks: Challenges, architectures, and future directions. *Vehicular Communications*, 9:268–280, 2017.

[65] H. Mistareehi. Message dissemination scheme for rural areas using VANET (Hardware Implementation). In *Proceedings of International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 120–125, Jeju Island, Korea, August 2021. IEEE.

[66] H. Mistareehi, T. Islam, K. Lim, and D. Manivannan. A secure and distributed architecture for vehicular cloud. In *Proceedings of Advances on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pages 127–140. Springer, Oct 2019.

[67] H. Mistareehi, T. Islam, and D. Manivannan. A secure and distributed architecture for vehicular cloud. *Internet of Things*, Jan 2021.

[68] H. Mistareehi and D. Manivannan. Classification, challenges and critical comparison of proposed solutions for vehicular clouds. *International Journal of Next-Generation Computing*, 10(1):1–18, Mar 2019.

[69] H. Mistareehi and D. Manivannan. A Low-Overhead message authentication and secure message dissemination scheme for VANETs. *Network*, 2(1):139–152, 2022.

[70] H. Mistareehi, D. Manivannan, and H. B. Salameh. A secure and improved safety message collection with increased privacy-preserving algorithm for vanets. In *Proceddings of International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 1–8. IEEE, 2022.

[71] H. Mistareehi, H. B. Salameh, and D. Manivannan. An on-board hardware implementation of aodv routing protocol in vanet: Design and experimental

evaluation. In *Proceedings of International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pages 1–6. IEEE, 2022.

[72] S. Moni and D. Manivannan. A lightweight privacy-preserving V2I mutual authentication scheme using cuckoo filter in vanets. In *Proceedings of IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pages 815–820, 2022.

[73] F. Mubarek, S. Aliesawi, K. Alheeti, and N. Alfahad. Urban-AODV: An improved AODV protocol for vehicular ad-hoc networks in urban environment. *International Journal of Engineering and Technology*, 7(4):3030–3036, 2018.

[74] M. Nagaraj and M. Dhamal. Performance comparison of AODV, AOMDV, OLSR, DSR and GSR routing protocols in VANET. *International Journal of Advances in Computer Networks and its Security*, 2, 2012.

[75] A. Nambi, S. Bannur, I. Mehta, H. Kalra, A. Virmani, V. Padmanabhan, R. Bhandari, and B. Raman. Hams: Driver and driving monitoring using a smartphone. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 840–842, 2018.

[76] V.-L. Nguyen, P.-C. Lin, and R.-H. Hwang. Enhancing misbehavior detection in 5g vehicle-to-vehicle communications. *IEEE Transactions on Vehicular Technology*, 69(9):9417–9430, 2020.

[77] S. Olariu, I. Khalil, and M. Abuelela. Taking VANET to the clouds. *International Journal of Pervasive Computing and Communications*, 7(1):7–21, February 2011.

[78] B. Palanisamy and L. Liu. MobiMix: Protecting location privacy with mix-zones over road networks. *IEEE 27th International Conference on Data Engineering*, pages 494–505, April 2011.

[79] Y. Pan and J. Li. Cooperative pseudonym change scheme based on the number of neighbors in VANETs. *Network and Computer Applications*, 36(6):1599–1609, 2013.

[80] N. Parameshwara, J. Marhi, and P. Chakraborty. Hardware prototype design for real-time collision detection and prevention application of VANET. *International Journal of Scientific & Engineering Research*, 9(7), July 2018.

[81] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, USA, February 1999. IEEE.

[82] Y. Qin, D. Huang, and X. Zhang. Vehicloud: Cloud computing facilitating routing in vehicular networks. In *Proceedings of IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1438–1445. IEEE, June 2012.

[83] G. Rakesh and M. Belwal. Hardware implementation of VANET communication based collision warning system. In *Proceedings of International Conference on Communication and Electronics Systems (ICCES)*, pages 1734–1738, Coimbatore, India, July 2019. IEEE.

[84] Z. Y. Rawashdeh and S. M. Mahmud. Intersection collision avoidance system architecture. In *Proceedings of 5th IEEE Consumer Communications and Networking Conference*, pages 493–494, 2008.

[85] M. Raya, P. Papadimitratos, and J.-p. Hubaux. Securing vehicular communications. *IEEE Wireless Communications*, 13(5):8–15, 2006.

[86] T. Refaat, B. Kantarci, and H. Mouftah. Dynamic virtual machine migration in a vehicular cloud. In *Proceedings of IEEE Symposium on Computers and Communication (ISCC)*, pages 1–6, June 2014.

[87] M. Salahuddin, A. Al-Fuqaha, and M. Guizani. Software-defined networking for RSU clouds in support of the internet of vehicles. *IEEE Internet of Things Journal*, 2(2):133–144, April 2015.

[88] E. Sallum, G. Santos, M. Alves, and M. Santos. Performance analysis and comparison of the DSDV, AODV, and OLSR routing protocols under VANETs. In *Proceedings of International Conference on Intelligent Transportation Systems Telecommunications (ITST)*, pages 1–7, Lisboa, Portugal, October 2018. IEEE.

[89] G. Samara, W. Al-Salihy, and R. Sures. Security analysis of vehicular ad hoc networks. In *Proceedings of Network Applications Protocols and Services*, pages 55–60. IEEE, September 2010.

[90] G. Santoso and M. Kang. Performance analysis of AODV, DSDV and OLSR in a VANETs safety application scenario. In *Proceedings of International Conference on Advanced Communication Technology (ICACT)*, pages 57–60, PyeongChang, South Korea, February 2012. IEEE.

[91] F. Scheuer, K. Fuchs, and H. Federrath. A safety-preserving mix zone for VANETs. In *Proceedings of International Conference on Trust, Privacy, and Security in Digital Business*, pages 37–48, 2011.

[92] M. Sharma, R. Bali, and A. Kaur. Dynamic key based authentication scheme for vehicular cloud computing. In *Proceedings of 2015 International Conference on Green Computing and Internet of Things*, pages 1059–1064. IEEE, October 2015.

[93] S. Sharma and S. Sebastian. IoT based car accident detection and notification algorithm for general road accidents. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(5):4020–4026, October 2019.

[94] R. Singh, D. Saluja, and S. Kumar. Graphical Approach for V2V Connectivity Enhancement in Clustering-Based VANET. *IEEE Wireless Communications Letters*, 10(6):1217–1221, June 2021.

[95] S. Sivakumar, A. Alagumurugan, G. Vignesh, and S. Dhanush. Accident analysis and avoidance by V2V communication using LIFI technology. *International Research Journal of Modernization in Engineering Technology and Science*, 2(3), March 2020.

[96] J.-H. Song, V. W. S. Wong, and V. C. M. Leung. Wireless location privacy protection in vehicular ad-hoc networks. In *proceedings of IEEE International Conference on Communications*, pages 1–6, 2009.

[97] I. Standard-1609.2. IEEE standard for wireless access in vehicular environments–security services for applications and management messages. *IEEE Std 1609.2-2016 (Revision of IEEE Std 1609.2-2013)*, pages 1–240, 2016.

[98] F. Toutain, A. Bouabdallah, and R. Zemek. Interpersonal context-aware communication services. *IEEE Communications Magazine*, 49(1):68–74, January 2011.

[99] S. Vijayan and K. Arun. Performance evaluation of routing protocols in Zigbee network: DSR vs AODV. *Journal of Applied Technology and Innovation*, 5(2), 2021.

[100] J. Wan, D. Zhang, S. Zhao, L. Yang, and J. Lloret. Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. *IEEE Communications Magazine*, 52(8):106–113, August 2014.

[101] J. Wang, J. Cho, S. Lee, and T. Ma. Real time services for future cloud computing enabled vehicle networks. In *Proceedings of International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–5. IEEE, November 2011.

[102] P. Wang and Y. Liu. Sema: Secure and efficient message authentication protocol for vanets. *IEEE Systems Journal*, 15(1):846–855, 2021.

[103] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40:325–344, April 2014.

[104] Wikipedia. RF module. *https://en.wikipedia.org/wiki/RF_ module*, 2021.

[105] Wikipedia. Rf zigbee module. `https://en.wikipedia.org/wiki/RFmodule`, 2022.

[106] M. Won, A. Mishra, and S. Son. HybridBaro: Mining driving routes using barometer sensor of smartphone. *IEEE Sensors*, 17(19):6397–6408, Oct 2017.

[107] K. Xu, K. Wang, R. Amin, J. Martin, and R. Izard. A fast cloud-based network selection scheme using coalition formation games in vehicular networks. *IEEE Transactions on Vehicular Technology*, 64(11):5327–5339, November 2015.

[108] G. Yan, D. Wen, S.Olariu, and M.Weigle. Security challenges in vehicular cloud computing. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):284–294, March 2013.

[109] C.-Y. Yang and S.-C. Lo. Street broadcast with smart relay for emergency messages in VANET. In *Proceeding of 24th International Conference on Advanced Information Networking and Applications Workshops*, pages 323–328, 2010.

[110] Y. Yang, L. Zhang, Y. Zhao, K.-K. R. Choo, and Y. Zhang. Privacy-preserving aggregation-authentication scheme for a safety warning system in fog-cloud based VANET. *IEEE Transactions on Information Forensics and Security*, 17:317–331, 2022.

[111] M. Yassein and H. Mistareehi. Improvement on the lifetime of the WSN using energy efficiency saving of leach protocol (New Improved LEACH). *Sensors and Transducers Journal*, 130(7):142–154, 2011.

[112] B. Ying and D. Makrakis. Pseudonym changes scheme based on candidate-location-list in vehicular networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, pages 7292–7297. IEEE, 2015.

[113] C. You, M.Oca, T. Bao, N. Lane, G. Cardone, L. Torresani, and A. Campbell. Carsafe: A driver safety app that detects dangerous driving behavior using dual-cameras on smartphones. In *Proceedings of the 2012 ACM Conference of Ubiquitous Computing*, page 671–672, 2012.

[114] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang. Toward cloud-based vehicular networks with efficient resource management. *IEEE Network*, 27(5):48–55, October 2013.

[115] C. Zang, X. Lin, R. Lu, and P. Ho. RAISE: an Efficient RSU-aided message authentication Scheme in vehicular communication networks. In *proceeding of IEEE International Conference on Communications (ICC)*. IEEE, May 2008.

[116] X. Zhang, Y. Li, and Q. Miao. A cluster-based broadcast scheduling scheme for mmWave vehicular communication. *IEEE Communications Letters*, 23(7):1202–1206, July 2019.

[117] J. Zhao and G. Cao. VADD: Vehicle-assisted data delivery in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 57(3):1910–1922, May 2008.

Vita

Hassan Mistareehi

Education:

- University of Kentucky, Lexington, KY
  Ph.D in Computer Science, May. 2023
- Jordan University of Science and Technology, Irbid, Jordan
  M.Sc in Computer Science, June. 2011
- Jordan University of Science and Technology, Irbid, Jordan
  B.S. in Computer Science, June. 2008
  cum laude

Professional Positions:

- Graduate Teaching Assistant Jan 2017–May 2023
  Department of Computer Science, University of Kentucky
- Lecturer Sep 2011–June 2013
  Department of Computer Science, Jordan University of Science and Technology, Jordan
- Teacher Sep 2008–June 2013
  Ministry of Education, Jordan

Certifications AND Awards:

- Trainer's leader certificate from all Jordan Youth Agency.
- Leadership skill certificate from Community Development Center at Jordan University of Science and Technology.
- ASP.net certificate from al Mansi center.
- International Computer Driving Licence (ICDL) certificate
- Grant to attend and present a paper at the ICUFN conference, 2021.
- Travel Grant from the Department of Computer Science, the University of Kentucky 2019
- Travel Grant from the Department of Computer Science, the University of Kentucky 2022

Publications:
Journal Papers:

- **Hassan Mistareehi**, and D. Manivannan. ¨ A low-overhead message authentication and secure message dissemination scheme for VANETs¨. Network. 2022 Mar 7;2(1): 139-152.

- **Hassan Mistareehi**, Tariqul Islam, and D. Manivannan.¨ A secure and distributed architecture for vehicular cloud¨. Internet of Things. 2021 Mar 1;13:100355.

- **Hassan Mistareehi**, and D. Manivannan.¨ Classification, challenges and critical comparison of proposed solutions for vehicular clouds¨. International Journal of Next Generation Computing. 2019 Mar 1:01-18.

- Yassein, Muneer Bani, and **Hassan Mistareehi**.¨ Improvement on the lifetime of the WSN using energy efficiency saving of LEACH protocol (New Improved LEACH) ¨. Sensors & Transducers Journal. 2011 Jul 1;130(7):142-154.

Conference Papers:

- **Hassan Mistareehi**, Haythem Bany Salameh, and D. Manivannan.¨ An On-Board hardware implementation of AODV routing protocol in VANET: design and experimental evaluation¨. In Proceedings of International Conference on Internet of Things: Systems, Management and Security (IOTSMS) 2022, (pp 1–6). IEEE. Milan, Italy.

- **Hassan Mistareehi**, D. Manivannan, and Haythem Bany Salameh. ¨ A secure and improved safety message collection algorithm for VANETs¨. In Proceedings of International Conference on Social Networks Analysis, Management and Security (SNAMS) 2022, pages 1–8. IEEE. Milan, Italy.

- **Hassan Mistareehi**. ¨ Message dissemination scheme for rural areas using VANET (Hardware Implementation)¨. In Proceedings of Twelfth International Conference on Ubiqui- tous and Future Networks (ICUFN) 2021 Aug 17 (pp. 120-125). IEEE. Jeju Island, South Korea.

- Tariqul Islam, **Hassan Mistareehi**, and D. Manivannan.¨ SecReS: A secure and reliable storage scheme for cloud with client-side data deduplication ¨. In Proceedings of IEEE Global Communications Conference (GLOBECOM) 2019 Dec 9 (pp. 1-6). IEEE. Hawaii, USA.

- **Hassan Mistareehi**, Tariqul Islam, and D. Manivannan.¨ A secure and distributed architecture for vehicular cloud¨. In Proceedings of The 14th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC) 2019 Oct (pp. 127–140). Springer. Antwerp, Belgium.