INFORMATION MANAGEMENT AND SECURITY PROTECTION FOR INTERNET OF VEHICLES

A Dissertation presented to

the Faculty of the Graduate School

at the University of Missouri

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

Jian Kang

Dr. Dan Lin, Supervisor

May 2022

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

INFORMATION MANAGEMENT AND SECURITY PROTECTION
FOR INTERNET OF VEHICLES

presented by Jian Kang,
a candidate for the degree of Doctor of Philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Dan Lin

Dr. Chi-Ren Shyu

Dr. Jian Lin

Dr. Wei Jiang

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor Prof. Dan Lin for the continuous support of my Ph.D. study and research, and for her motivation, patience, and enthusiasm that helped to shape my research skills. Without her guidance and persistent help, this study would not have been possible.

I would like to thank Prof. Yingjie Wu for his immense knowledge and plentiful experience that encourage me all the time in my academic research. I would like to thank Prof. Wei Jiang for his valuable advice during my course learning and research process. I would also like to thank Dr. Adam Bowers, whose passion for research has inspired me to keep working hard on projects and research work.

Besides, a thank you to my colleagues in the iPrivacy lab. I learned a lot from the discussions with them, both in academics and in life. I am proud to be a member of this fantastic research group.

I also acknowledge with a deep sense of reverence, my gratitude to my parents and my wife who have always supported me through all these years.

Last but not least gratitude goes to all of my friends for their company and support.

TABLE OF CONTENTS

LIST OF PUBLICATIONS

Referred Journal Papers

- Jian Kang, Doug Steiert, Dan Lin, and Yanjie Fu. MoveWithMe: Location Privacy Preservation for Smartphone Users. IEEE Transactions on Information Forensics and Security (TIFS), 15 (1): 711-724, 2019.

- Jian Kang, Dan Lin, Wei Jiang, and Elisa Bertino. Highly efficient randomized authentication in VANETs. Pervasive and Mobile Computing (PMC), 44:31-44, 2018.

- Dan Lin, Jian Kang, Anna Squicciarini, Yingjie Wu, Sashi Gurung, and Ozan Tonguz. MoZo: A Moving Zone Based Routing Protocol Using Pure V2V Communication in VANETs. IEEE Transactions on Mobile Computing (TMC), 16(5):1357-1370, 2017.

- Yingjie Wu, Liqun Zhang, Jian Kang, and Yilei Wang. An Algorithm for Differential Privacy Streaming Data Adaptive Publication. Journal of Computer Research and Development, 54(12):2805-2817, 2017. (in Chinese)

- Jian Kang, Yingjie Wu, Siyong Huang, Hong Chen, and Lan Sun. An Algorithm for Differential Private Histogram Publication with Non-uniform Private Budget. Journal of Frontiers of Computer Science and Technology, 10(6):786-798, 2016. (in Chinese)

Referred Conference Papers

- Jian Kang, Alian Yu, Wei Jiang, Dan Lin. NWADE: A Neighborhood Watch Mechanism for Attack Detection and Evacuation in Autonomous Intersection Management. The 42th IEEE International Conference on Distributed Computing Systems (ICDCS), 2022.

- Jian Kang, Dan Lin. DASH: A Universal Intersection Traffic Management System for Autonomous Vehicles. The 40th IEEE International Conference on Distributed Computing Systems (ICDCS), 2020.

- Jian Kang, Dan Lin, Elisa Bertino, and Ozan Tonguz. From Autonomous Vehicles to Vehicular Clouds: Challenges of Management, Security and Dependability. The 39th IEEE International Conference on Distributed Computing Systems (ICDCS), 2019.

- Jian Kang, Yousef Elmehdwi, and Dan Lin. SLIM: Secure and Lightweight Identity Management in VANETs with Minimum Infrastructure Reliance. In International Conference on Security and Privacy in Communication Systems (SecureComm), Springer, Cham, 823-837, 2017.

- Zhenqin Huang, Jian Kang, Yingjie Wu, Siyong Huang, and Shaozhen Ye. An Algorithm for Differentially Private Histogram Publication Based on the Probability of Range Query. Proc. of the 31th National Database Conference of China (NDBC), Tai Yuan, China, 2014. (in Chinese)

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABSTRACT

Considering the huge number of vehicles on the roads, the Internet of Vehicles is envisioned to foster a variety of new applications ranging from road safety enhancement to mobile entertainment. These new applications all face critical challenges which are how to handle a large volume of data streams of various kinds and how the secure architecture enhances the security of the Internet of Vehicles systems. This dissertation proposes a comprehensive message routing solution to provide the fundamental support of information management for the Internet of Vehicles. The proposed approach delivers messages via a self-organized moving-zone-based architecture formed using pure vehicle-to-vehicle communication and integrates moving object modeling and indexing techniques to vehicle management. It can significantly reduce the communication overhead while providing higher delivery rates.

To ensure the identity and location privacy of the vehicles on the Internet of Vehicles environment, a highly efficient randomized authentication protocol, RAU+ is proposed to leverage homomorphic encryption and enable individual vehicles to easily generate a new randomized identity for each newly established communication while each authentication server would not know their real identities. In this way, not any single party can track the user.

To minimize the infrastructure reliance, this dissertation further proposes a secure and lightweight identity management mechanism in which vehicles only need to contact a central authority once to obtain a global identity. Vehicles take turns serving as the captain authentication unit in self-organized groups. The local identities are computed from the vehicle's global identity and do not reveal true identities.

Extensive experiments are conducted under a variety of Internet of Vehicles environments. The experimental results demonstrate the practicality, effectiveness, and efficiency of the proposed protocols.

Chapter 1

INTRODUCTION

In the Internet of Vehicles environments, huge amounts of data are collected by embedded sensors, processed by on-board computing units, and disseminated via the various communication networks to which vehicles are connected. By utilizing such rich resources, many new applications are emerging, such as autonomous driving management, traffic management, content sharing, etc. These applications can enhance road safety, increase traffic efficiency and improve the comfort of drivers and passengers [1].

Despite the benefits of the Internet of Vehicles applications, how to utilize those resources of vehicles efficiently, effectively, and securely is far from simple. Due to the highly dynamic and heterogeneous nature of vehicles, the network topology in the Internet of Vehicles is much more dynamic. The short-term relationships among vehicles make it an extremely challenging task for the information management for the Internet of Vehicles environment. Therefore, a fundamentally supporting architecture is needed to provide a number of critical functions: organizing large amounts of heterogeneous vehicles, routing messages among vehicles, infrastructures, and servers through frequently interrupted links, and managing the various entities without imposing too much burden on the entire system.

Besides that, to enhance the security of the Internet of Vehicles environment, it is critical for vehicles to verify the identities of other vehicles in the Internet of Vehicles

environments. Without knowing vehicles' identities, it is not possible to determine whether a vehicle is a legal node and what kind of role the vehicle serves. The authentication protocols should be designed specifically to fit the unique characteristics of the Internet of Vehicles.

To address these problems, this dissertation proposes: (1) a comprehensive routing protocol that can deliver messages in the Internet of Vehicles via a self-organized moving-zone based architecture using pure vehicle-to-vehicle communication, (2) a highly efficient authentication protocol, RAU$^+$, which leverages homomorphic encryption to preserve vehicles' privacy while ensuring traceability, and (3), a secure and lightweight identity management mechanism, SLIM, for vehicle-to-vehicle communications to minimize the reliance on the infrastructure support.

1.1 MESSAGE ROUTING PROTOCOL FOR INTERNET OF VEHICLES

A key requirement for the realization of the Internet of Vehicles applications is the availability of efficient and effective routing protocols for message dissemination. Without well-defined and efficient routing protocols, vehicles may be unable to share important messages and enjoy the benefits of the advanced technologies offered by the Internet of Vehicles. To address these issues, many routing protocols have been proposed. Broadly, these existing protocols can be classified into five main categories, namely broadcasting protocols [2], route-discovery protocols [3, 4, 5], position-based protocols [6, 7], clustering-based protocols [8, 9] and infrastructure-based protocols [10]. While effective for specific applications and contexts, these protocols are still limited in their applicability and practical use. The broadcasting protocols rely on large message dissemination and hence may cause a high communication overhead and message congestion on the network. To prevent this, broadcast storm mitigation techniques have been proposed [11]. The route-discovery protocols require discovering

a route before sending out a message, and hence may not be suitable for applications with strict time constraints. The position-based protocols require vehicles to pass messages to nearby vehicles moving towards the final destination of the message. Such protocols require each vehicle to maintain information about neighboring vehicles, resulting in frequent message exchange between each pair of vehicles, and hence their overall communication cost is typically higher than clustering-based protocols that arrange vehicles into clusters and only need the cluster heads to maintain neighboring information. The infrastructure-based routing protocols heavily rely on road-side units (RSUs) which are currently not widely available and have experienced a very slow deployment rate due to their high cost.

Among all types of protocols, clustering-based protocols appear to be the most promising as they attempt to capture the mobility of nodes in the Internet of Vehicles in a natural way and provide relatively stable units (i.e., the clusters of vehicles) for communication. However, most of existing clustering-based approaches [9], [12], [13] focus on how to cluster vehicles but do not provide the follow-up routing strategies. There lacks of study on whether the expected improvement in routing efficiency can offset the overhead (i.e., computing delay, amount of message exchanged for clustering) incurred by gaining stable clusters. For example, if forming stable clusters of vehicles requires significantly more message exchanges than simply delivering messages without using clusters, such clustering may not be useful in practice.

This dissertation proposes a comprehensive routing solution that delivers messages in the Internet of Vehicles via a self-organized moving-zone-based architecture formed using pure vehicle-to-vehicle communication. The proposed routing protocol was compared with both clustering-based approaches and non-clustering-based approaches to demonstrate the advantages of the proposed approach. Figure 1.1 illustrates an overview of the key concepts behind the proposal, where the cloud symbol denotes moving zones and arrows indicate the message propagation route.

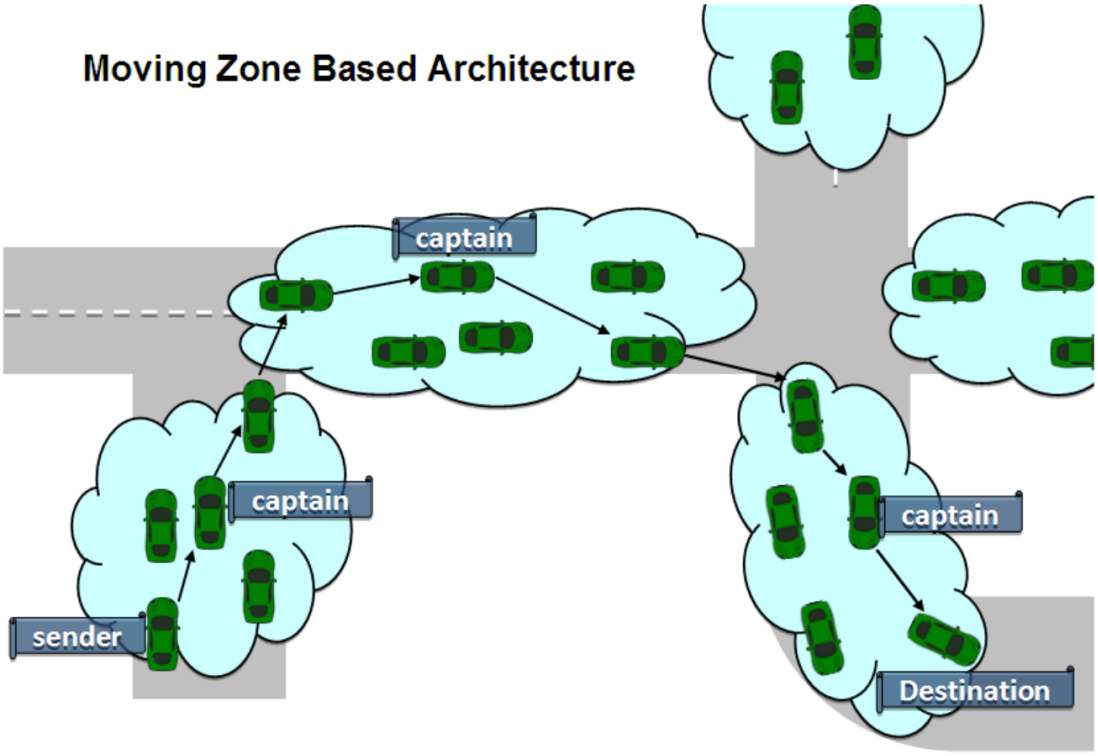


Figure 1.1: Moving Zone Based Architecture for Internet of Vehicles

The proposed approach integrates moving object modeling and indexing techniques [14] to vehicle management. Moving object techniques allow us to provide a realistic cluster-based representation, in that vehicles are grouped according to their actual moving patterns. Further, the use of indexes allows for efficient movement of information storage and management. Specifically, the proposed approach reduces the update frequency since vehicles no longer need to periodically send location updates to the cluster head (called "captain vehicle"). Instead, vehicles just need to update their movement functions when their moving direction or speed changes dramatically. Second, unlike cluster heads in other existing protocols, the captain vehicle in the proposed protocol has the ability to estimate vehicle positions in the near future so that decisions (e.g., zone splitting, message routing) can be made without requiring constant location updates from member vehicles. Third, the use of an index reduces the need for the captain vehicle to contact and examine every member vehicle

for each event or operation since information of vehicles affected by the event can be quickly accessed via the index. As demonstrated by the experimental results, the proposed approach significantly reduces the existing routing protocols' communication overhead to 1/10 while providing higher delivery rates.

1.2 RANDOMIZED AUTHENTICATION FOR INTERNET OF VEHICLES

With the fundamental support of the message routing protocols, vehicles can act as network nodes and communicate with one another to share information. Considering a large number of vehicles on roads, a variety of new services are envisioned, ranging from driving safety enhancement [15], dynamic route planning [16], to mobile entertainment [17]. For example, a vehicle may send inquiries to vehicles around certain landmarks to obtain the up-to-date traffic situation, the condition of a road, or parking information; passengers in vehicles can exchange files or chat with people in other vehicles along the trip.

One of the key components toward the successful roll-out of Internet of Vehicles applications is to provide security and privacy guarantees. Otherwise, the rich functionality and services provided by the Internet of Vehicles may be abused, jeopardizing the safety of drivers and passengers. For example, a malicious vehicle can claim a fake traffic jam to gain the right of the road and cause other vehicles to make an unnecessary detour. Therefore, vehicles should be authenticated before they are allowed to exchange messages in the Internet of Vehicles.

Meanwhile, users' privacy should be preserved during authentication. Specifically, their real identities should be kept private and their locations should not be disclosed to the servers [18]. Otherwise, the authentication server may obtain the behavior pattern or track the user locations by keeping the records of when and where the user requests authentication. Similarly, peer vehicles may also be able to track each other

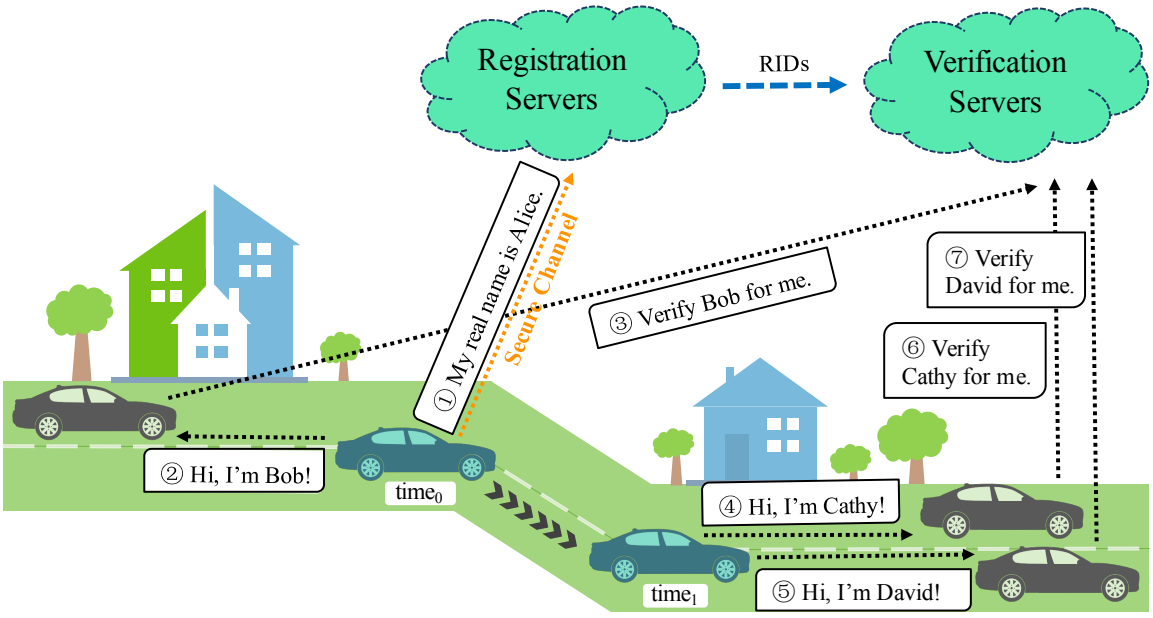


Figure 1.2: Example Scenarios of the Proposed Randomized Authentication in Internet of Vehicles.

by linking users with the same pseudonyms. On one hand, such server-wise and peer-wise privacy concerns should all be addressed in the Internet of Vehicles applications. On the other hand, the Internet of Vehicles application should still ensure traceability whereby law enforcement authorities are able to reveal the locations that the suspect vehicle has been to when disputes occur. Privacy preservation and traceability are two seemly conflicting requirements and hence it is one of the critical challenges that the proposed work aims to address.

At a first glance, one may feel that the aforementioned security and privacy concerns resemble those encountered in other communication networks, especially Mobile ad-Hoc Networks (MANETs). However, compared to MANETs, Internet of Vehicles environments have a larger number of nodes with higher mobility and the communication links break more frequently than in MANETs [19]. Due to these differences in the network environment, solutions proposed in MANETs or other types of networks may not be suitable for the Internet of Vehicles.

This dissertation proposes a highly efficient authentication protocol RAU$^+$. The

RAU$^+$ inherits all the security properties from RAU, i.e., preserves vehicles' privacy while ensuring traceability. In particular, the RAU$^+$ leverages homomorphic encryption and enables individual vehicles to easily generate a new identity for each newly established communication. These randomized identities can be verified through the collaboration of a pair of authentication servers while each authentication server would not know the real identity of the authentication requester. Figure 1.2 shows simple example scenarios. For traceability, the pair of authentication servers will execute a collaborative protocol so that the real identity of the malicious vehicle can be identified. In this way, not any single party in the Internet of Vehicles is able to track the users.

Compared to RAU [20], the RAU$^+$ is more advanced in terms of efficiency and usability. Specifically, the RAU$^+$ provides a new type of authentication, namely aggregated authentication, which allows one vehicle to verify multiple vehicles simultaneously with a single request message to the verification server. In RAU, such verification with multiple vehicles will have to be conducted separately for each vehicle and hence these requests are very time-consuming. Compared to existing Internet of Vehicles authentication works [21, 22, 23, 24, 25], the proposed RAU$^+$ has a number of advantages. First, the RAU$^+$ does not require any pre-generation of a long list of pseudonyms which could cause complicated ID revocation problems. Second, the RAU$^+$ does not need the server to generate pseudonyms every time which prevents pseudonym generators, such as road-side units or group managers (i.e., peer vehicles) used in other works, from tracking the vehicles. Third, the RAU$^+$ does not require the availability of road-side units which are not widely available in the real world due to deployment costs. Fourth, the RAU$^+$ is efficient and meets the real-time constraints in the Internet of Vehicles applications well. A more detailed security analysis and performance studies will be presented in Chapter 4.

1.3 IDENTITY MANAGEMENT IN THE INTERNET OF VEHICLES WITH MINIMUM INFRASTRUCTURE RELIANCE

Most of the existing identity management approaches for the Internet of Vehicles assume the existence of Road-Side Units (RSUs) to serve as the trusted party during the authentication. However, building RSUs is costly and may not be able to capture the speed of the deployment of the Internet of Vehicles networks in the near future.

Aiming at minimizing the reliance on infrastructure support, this dissertation proposes a Secure and Lightweight Identity Management (SLIM) mechanism for vehicle-to-vehicle communications. Specifically, the SLIM scheme has an initial registration phase where the vehicles only need to contact a central authority once the first time they log on Internet of Vehicles to obtain a global identity. This global identity is tied to the vehicle's identification number (VIN) without explicitly revealing this information. Then as vehicles move around, they self-organize into groups of similar interest or destinations using the previously proposed moving-zone forming protocols [26]. Inside each moving zone, vehicles take turns to serve as the captain authentication unit (CAU) which will be in charge of generating a temporary local identity for each member vehicle to communicate with peers. The local identities are computed from the vehicle's global identity and do not reveal the true identity of vehicles to the CAU or peer vehicles. Moreover, the SLIM mechanism also supports traceability in that the true identity of a malicious vehicle can be recovered through collaboration between other peer vehicles and the central authority. The proposed approach had been implemented and the performance had been compared with the most related vehicle-to-vehicle-based authentication approach [27]. The experimental results show that the SLIM is much faster during vehicle-to-vehicle authentication.

The proposed SLIM mechanism has the public key infrastructure as the building block similar to many existing works. However, compared to the existing works,

the SLIM has three major advantages: (i) the SLIM mechanism does not rely on infrastructure support during vehicle-to-vehicle communication; (ii) the SLIM mechanism is more secure than other vehicle-to-vehicle-based authentications such as [27] in that the SLIM can defend against more types of attacks as discussed in Section 5.3; (iii) the SLIM mechanism is more efficient for vehicle-to-vehicle authentication by distributing the authentication workload such as the key generation over time.

1.4 OUTLINE OF THE DISSERTATION

The rest of this dissertation is organized as follows:

- Chapter 2 reviews routing protocols for the Internet of Vehicles, and surveys the state-of-the-art privacy-preserving authentication approaches.
- Chapter 3 presents the proposed MOving-ZOne-based (MoZo) architecture and the routing protocol for the Internet of Vehicles. The proposed approach greatly reduces communication overhead and improves the message delivery rate compared to other existing approaches.
- Chapter 4 presents a highly efficient randomized authentication protocol, RAU+, that leverages homomorphic encryption and enables individual vehicles to easily generate a new identity for each newly established communication.
- Chapter 5 presents a Secure and Lightweight Identity Management (SLIM) mechanism for vehicle-to-vehicle communications aiming at minimizing the reliance on infrastructure support.
- Chapter 6 concludes the proposed work and discusses directions for future work.

Three papers have been published from the work reported in this dissertation. The main idea of the MoZo routing protocol for the Internet of Vehicles, presented in Chapter 3, has been published in [26]. The work on privacy-preserving authentication

protocol, presented in Chapter 4, has been published in [28]. The work on secure and lightweight identity management for the Internet of Vehicles, presented in Chapter 5, has been published in [29].

Chapter 2

LITERATURE REVIEW

2.1 ROUTING PROTOCOLS FOR INTERNET OF VEHICLES

Many types of routing protocols have been proposed for the Internet of Vehicles, as surveyed in [30, 31, 32]. Since the proposed message routing protocol is closely related to clustering-based approaches, this section first briefly reviews works under this category. Then, discusses two approaches (CBDRP [33] and BRAVE [34]) in more detail since they have been selected for comparison in the experimental study.

2.1.1 Clustering-based Approaches:

Although the Internet of Vehicles shares some similar features with Mobile Ad-hoc Networks (MANETs), clustering techniques for MANETs cannot be directly applied. Fan et al. [35] attempted to adapt MANET algorithms to vehicular ad-hoc networks. However, such adaptation still cannot address the unique characteristics that vehicular networks possess [36]. For example, energy is no longer an issue and vehicles have high mobility making network topology highly dynamic.

As for vehicular ad-hoc networks, one of the earliest vehicle clustering algorithms is proposed by Kayis and Acarman [37]. The proposed passive clustering algorithm conducts clustering only when data is to be communicated. The clustering is based on predefined speed intervals such as [0, 30 mph] and [30mph, 45mph]. Vehicles traveling within the same speed interval form a cluster, and the vehicle which first claims

to be the cluster head becomes the cluster head. However, the speed interval is not sufficient to capture the similarity in mobility. For example, two vehicles with very similar speeds 29mph and 31mph are grouped in different clusters. Moreover, this approach does not consider location proximity either. Vehicles that are very close to one another may stay together for a certain period of time even if they travel at different speeds. In [38], Chen et al. use only the distance between vehicles as the clustering criteria in that vehicles close to one another are grouped in the same cluster. Further, their approach relies on a central server to handle cluster merging and splitting events, while the proposed approach is fully decentralized. In [39], Want et al. proposed a priority-based clustering algorithm. Each vehicle calculates its priority according to its estimated travel time and speed deviation. A vehicle having longer travel time and less speed deviation will have higher priority and have a higher chance to become the cluster head. Each vehicle shares its cluster information with its neighbors. This approach requires continuous communication among vehicles. The authors do not discuss how to take advantage of these clusters for routing, and hence it is not clear whether the overhead introduced by clustering will be offset during information routing. In [8], Shea et al. developed a clustering algorithm, called affinity propagation. Neighboring vehicles exchange their IDs, current positions, current velocities, etc., and compute the affinity function to select the cluster head. However, this approach may not be suitable for routing purposes as a large number of messages are exchanged exhausting the available bandwidth. In [40, 41], clustering is done for some specific applications, for instance, to calculate the amount of traffic. Such approaches do not care about the stability of clusters and are not suitable for supporting routing protocols. To increase cluster stability, [12] and [13] both consider vehicle mobility during clustering, but they did not provide any routing strategies. In addition, clustering is also used for message authentication purposes and is conducted when needed. Most recently, Hadded et al. [9] develop a vehicle clustering approach based on a multi-

objective genetic algorithm. To sum up, all the aforementioned works mainly focus on measuring the stability of generated clusters but did not provide any routing algorithm on how to use the generated clusters to conduct efficient message routing. Unlike these works, the proposed approach provides a complete routing algorithm that consists of efficient clustering, long-term maintenance, and efficient routing.

There has been some work that includes both clustering and routing algorithms. However, some of these approaches rely on infrastructure support which may not be available soon in the near future due to the deployment cost. For example, in [42], Alawi et al. propose to find the route from a vehicle to the closest infrastructure using the signal strength as a guiding criterion. Similarly in [43], message delivery is conducted with the aid of the infrastructure. Unlike these works, the aim of this work is to design an approach that utilizes vehicle-to-vehicle communications only. Related works using pure vehicle-to-vehicle communication are summarized as follows. In [44], Little and Agrawal proposed to utilize a cluster header and a trailer at the front and the rear end of each cluster for information routing. However, a detailed election protocol is not presented. In [45], Goonewardene et al. designed a vehicle precedence algorithm to adaptively identify the nearby 1-hop neighbors and select optimal cluster heads based on vehicle locations and velocities. The main limitation of this approach is that the proposed algorithm requires each vehicle to keep sending out updated information to neighbors which can introduce lots of communication overhead. In [46], Luo et al. form clusters based on geographically divided grids, but they did not consider velocity and direction which are important for accommodating the dynamic nature of vehicular ad-hoc networks. In [47], Ohta et al. use positions and moving direction of vehicles for clustering. Unlike these studies, the proposed work considers the rich mobility information during the clustering. Also, the cluster heads in [47] need to continuously broadcast MEP (cluster MEmber Packet), and it is only at this point that it can discover neighboring clusters for further routing. Another recent

work is by Song et al. [33], who consider moving directions for cluster head selection. In summary, none of the existing clustering approaches in pure vehicle-to-vehicle environments considers the use of moving object techniques to reduce communication overhead, and improve efficiency and effectiveness, as presented in this dissertation.

However, these clustering algorithms have at least one of the following limitations: (i) Clusters are formed based on a partitioning of road networks instead of object mobility, which reduces the lifetime of clusters; (ii) Clustering process requires each vehicle to periodically broadcast messages or has a complicated voting mechanism, which can incur high communication overhead; (iii) Clustering needs the assistance of road-side units which may not be available in many environments; (iv) Clustering is focused on small-scale scenarios (e.g., hundreds of vehicles). The proposed research will overcome these limitations.

2.1.2 Representative Approaches for Comparison:

In order to thoroughly evaluate the proposed approach, a comparison has been conducted between the proposed approach and both clustering-based approaches and non-clustering-based approaches.

As the representative approach of clustering-based routing protocol, the proposed work selects CBDRP [33] since its schema is most similar to ours. As the representative approach of non-clustering-based routing protocols, the proposed work selects BRAVE [34] because it has shown to outperform many existing protocols including GSR [48], SAR [49], A-STAR [50], GPCR [7], GeOpps [51].

The CBDRP (Clustering-Based Directional Routing Protocol) first divides each road into equal-length segments. Vehicles in the same road segment and moving in the same direction are grouped in one cluster, and the vehicle closest to the center of the cluster is the cluster head. To route a message, the source sends the message to its cluster head, and the cluster head establishes the routing path first and then forwards

the message along the path. There are two major limitations to this approach. First, the clusters are formed based on fixed partitioning of roads, without considering the similarity of movement among vehicles. As a result, the members in each cluster are updated very frequently, and this incurs heavy communication overhead. Second, the routing protocol requires the establishment of the path beforehand. The path may need to be maintained when the actual message is forwarded due to the dynamic nature of vehicles. The broken path problem is more severe when the distance between the sender and the receiver is far from each other. Such routing protocol not only introduces extra communication costs but also delays the transmission of messages.

The BRAVE (Beacon-less Routing Algorithm for Vehicular Environments) approach adopts an optimistic routing approach to reduce the message overhead in traditional broadcasting approaches. In BRAVE, a forwarding vehicle that has a message to send out will broadcast the message to its 1-hop neighbors. Every neighbor receiving the message will send back a response message. After the forwarding vehicle receives a response message, it will broadcast a select message to indicate which neighbor has been selected to forward the message. There have been some variants of BRAVE, such as BIIR [52] which achieves slightly better performance than BRAVE by reducing the message overhead to 2/3. Compared with BRAVE and its variants, the proposed work does not rely on broadcasting but more on target-oriented communication. The message overhead in the proposed approach is an order of magnitude less than BRAVE.

2.2 PRIVACY-PRESERVING AUTHENTICATION FOR INTERNET OF VEHICLES

Existing works on privacy-preserving authentication for the Internet of Vehicles can be classified into two main categories: (i) pseudonym-based protocols; and (ii) group-based protocols.

2.2.1 Pseudonym-based Authentication Protocols

The general goal of the pseudonym-based authentication protocols is to enable vehicles to use different pseudonyms during communication rather than using their real identities. One of the earliest works in this category is by Raya and Hubaux [24]. They suggested that when a vehicle needs to sign a message, it randomly selects a private key from a huge pool of certificates issued by the authority. The message receiver will verify the sender's signature by checking the validity of the corresponding public key certificate. The problem with this protocol is that vehicles need to check a long list of revoked certificates when verifying each received signed message, which is very time-consuming. Raya et al. in [53] proposed efficient revocation schemes. However, these schemes violate the location privacy requirement and are subject to a movement tracking attack. In order to reduce the average overhead of message authentication, Calandriello et al. [54] proposed a hybrid scheme, which is also computationally expensive because it needs to check if the group signature is from a revoked vehicle [55]. Other pseudonym-based protocols can be found in [23, 25, 56, 57, 58, 59, 60], achieving different degrees of improvement over the key revocation problem. However, in most of these protocols, the identity management authority is required to maintain the certificates associated with each vehicle so as to retrieve the vehicles' real identities when disputes occur. This allows the authority to track the vehicles' movement; hence, the vehicles' privacy is not fully preserved.

2.2.2 Group-based Authentication Protocols

Another category of privacy-preserving authentication protocols is group-based [21, 61, 62, 63]. The typical idea is to utilize group managers to group and authenticate vehicles, which enables vehicles to anonymously communicate with group members. Many group-based protocols leverage the group signature scheme. Under the group signature scheme, vehicles can only verify that the messages are from a valid group

member but do not know who is the actual sender, and hence vehicles are anonymous to their group members. For example, in the ECPP protocol proposed by Lu et al. [21], RSUs (Road Side Units) serves as the group manager who assigns the group keys to passing vehicles. The security and privacy of ECPP are later strengthened by Jung et al. [62] whose protocol guarantees unlinkability and traceability when multiple RSUs are compromised. Since the computation cost of the group signature scheme is very high, some techniques have been proposed to improve efficiency, such as the distributed key management framework by Hao et al. [64] and the decentralized certificate authority with the biological-password-based two-factor authentication by Wang et al. [65]. Besides group-based signature schemes, other techniques have also been proposed to achieve anonymity within a group. For example, Zhang et al. [23] adopted the k-anonymity concept for preserving user privacy so that a vehicle is indistinguishable from $k-1$ other vehicles. However, k-anonymity requires at least k vehicles in the vicinity which may not always be feasible in areas with few vehicles. In [22], Squicciarini et al. proposed a PAIM protocol that dynamically constructs groups via pure vehicle-to-vehicle communication, and leverages Pedersen commitment and secret sharing scheme to achieve anonymous authentication of vehicles. However, the proposed protocol requires a complicated group management strategy which introduces extra overhead to the system. In addition, there have been some works [66, 67, 68, 69] developed based on the ring signature or blind signature for privacy-preserving authentication.

In general, the existing group-based protocols have at least one of the following disadvantages. First, the group manager has all the knowledge about group members and hence is able to track them. Second, the process of group updates and membership revocation is usually very costly due to a large number of vehicles and high mobility of vehicles. Third, the communication is constrained to group members. This requires an efficient and dynamic grouping algorithm which currently is still a

challenging issue. Moreover, those protocols relying on the presence of infrastructure support (e.g., RSUs) may not be feasible in a reality where RSUs rarely exist. Most recently, some hybrid approaches like CACPPA [70] have been proposed, which utilize both the concept of pseudonym-based approaches and group-signature-based approaches. They also use a cloud authority but it is different from the proposed work that utilizes multiple cloud authorities to achieve separation of duty.

2.2.3 Anonymous Credential System

The anonymous credential (AC) system [71, 72, 73] has been chosen as a comparative approach in the experiments due to its popularity and the similar security goal of providing randomized identities for users. However, the AC system has the following limitations that make it not the best fit for the Internet of Vehicles. First, the AC system uses zero-knowledge proofs to verify if a user possesses a valid credential. Zero-knowledge proofs are computational expensive which may not be practical in the Internet of Vehicles since vehicles may have already moved a far distance when waiting for authentication. Second, the AC system provides an authentication protocol between users and organizations and hence assumes that they are already connected by pre-established secure channels. However, establishing security channels between vehicles is a tricky task and could also be time-consuming if it is not integrated with the authentication protocol. Third, a valid credential in the AC system may be shared among malicious users who are not authorized to use the credential to obtain services. To discourage users from sharing their credentials, one solution is to ask each user to give the organizations (or verifiers) verifiable encryption of his valuable secret information that can be decrypted with his secret key. Unfortunately, this approach is not practical either in the scenario of vehicle-to-vehicle communication because vehicles are not trustworthy organizations and hence servers need to assist the process which could be very time-consuming and requires modification of the original AC protocols.

Although there is another solution to avoid credential sharing by using hardware [74, 75, 76], the proposed approach does not assume a user's computing device is equipped with such specialized hardware. Further, the AC system does not have an efficient and effective method to revoke a credential. The credential revocation problem was discussed in [77], but the solution only works when the system is adopted as a regular credential system (without randomizing a user's credential for each authentication). There are some other extensions of AC [72, 73], but none of them directly addresses the aforementioned disadvantages.

Most existing privacy-preserving authentication schemes such as those discussed above, all heavily rely on some sort of infrastructure such as RSUs. However, RSUs would be expensive to deploy and are not expected to be widely available anytime soon. Very few works provide privacy-preserving authentication based on pure vehicle-to-vehicle communication. One representative work could be the PAIM scheme proposed by Squicciarini et.al [27]. Since the proposed work will be compared with PAIM, a more detailed review of this system has been provided as follows. The PAIM protocol dynamically constructs groups via pure vehicle-to-vehicle communication and leverages Pedersen commitment and secret sharing scheme to achieve anonymously authentication of vehicles. The biggest drawback of the Pedersen commitment scheme is that it is malleable. A commitment scheme is non-malleable [78, 79, 80] if one cannot transform the commitment of another person's secret into one of a related secret. Unfortunately, this property is not achieved by Pedersen commitment scheme [81, 82] because it is only designated to hide the secret. Compared to PAIM, the SLIM scheme also has the concepts of global identities and local identities. However, the protocols to generate the global and local identities are totally different, which makes the proposed SLIM scheme more secure and more efficient during vehicle-to-vehicle authentication.

Chapter 3

THE MOVING ZONE BASED ROUTING PROTOCOL

3.1 MOVING ZONE-BASED VEHICLE MANAGEMENT ARCHITECTURE

The MOving-ZOne-based (MoZo) architecture consists of multiple moving zones that are formed by vehicles with similar movement patterns. A captain vehicle is elected for each zone and is responsible for managing information about other member vehicles as well as the message dissemination. In the following subsections, we first introduce how to model vehicle movement, and then present the detailed algorithms for zone construction.

3.1.1 Vehicle Movement Modeling

This work assumes that each vehicle is equipped with an on-board unit (OBU) for networking and computing messages, a global positioning system (GPS), and a digital map. Vehicles communicate with one another using data link technology (e.g., ASTM E2213-03 [83]), within a range of 10s minimum travel time (the minimum range is 110 meters and the maximum is 300 meters) [84]. Further, this work assumes each vehicle has a unique identity which can be either a pseudonym or a real identity. Existing security and privacy protection techniques can be integrated with our approach while a detailed discussion of this possibility is beyond the scope of this chapter.

The road network is represented as a graph whereby edges represent roads and

vertexes represent intersections. The two ends of the roads are designated as the starting and ending points respectively. The vehicle's movement is modeled as a linear function of time. Specifically, let $r(st, ed)$ be a road segment, where st is the starting point of the road and ed is the endpoint of the road. Given a vehicle on road r, let l_u be the vehicle's distance to st at time t_u, and let v be vehicle's speed at t_u. Let δ denote the vehicle's moving direction, which has value 1 if the vehicle moves toward ed, otherwise, -1 if the vehicle moves toward st. Let t'_u denote the next possible update timestamp when the vehicle changes its moving speed or direction. Then, the vehicle's position at timestamp $t(t_u \leq t \leq t'_u)$ is computed as follows: $l(t) = l_u + \delta * v * (t - t_u)$.

This model will be adopted by the captain vehicle to estimate its member vehicle relative positions on a road. Vehicles need to send the update message to the captain vehicle if they change their moving directions or speed dramatically. The movement of vehicles is modeled as a straight line between two consecutive update messages, which is analogous to the widely adopted idea of using line segments to approximate curvy roads. Since moving functions usually change much less frequently than locations, the adoption of such modeling will reduce the need for the member vehicles to send location updates to the captain vehicles constantly. The moving function will also play an important role in determining the members of a moving zone as discussed in the subsequent sections.

3.1.2 Moving Zone Construction

Moving zone construction starts from a vehicle logging onto the Internet of Vehicles networks. The vehicle will execute the joining protocol to find a nearby moving zone or form its own zone. The zone forming criteria is configured based on the similarity of vehicle movement. The captain vehicle of each zone maintains a moving object index that manages up-to-date information about all its member vehicles. In what

follows, we discuss the operations that need to be conducted at member vehicle side and the captain vehicle side respectively.

Member Vehicle Side

When a vehicle Vs enters the Internet of Vehicles networks, it sends a hello message to its one hop neighbors. The hello message consists of its unique identifier Vs, current road ID (ID_r) and moving direction (δ). The vehicle waits for τ amount of time to accumulate the responses to its hello message. τ is the estimated total time for a single message to be received, processed by the receiver, and transmitted and propagated back to the sender within the communication range of the sender vehicle.

If a captain vehicle moving in the same direction (δ) receives the hello message, it sends a response to the corresponding vehicle. The response includes its unique identifier V_{cap}, current location l, speed v, and the next intersection Int that it is heading to. We will discuss how to select the captain vehicles in Section 3.3.

When τ expires, the vehicle calculates a similarity score for each response received from the neighboring captain vehicles. The goal is to assign a higher score to the captain vehicle which will stay closer to the vehicle for a longer time period so that the vehicle can find a zone in which it can stay longer. To accomplish this, we define the similarity score based on the average distance between the two vehicles' anticipated trajectories within a certain time period. The computation includes the following three steps.

The first step is to determine how far into the future the anticipated trajectories should be considered, i.e., the time period for computing the average distance. Figure 2 shows an example. The two arrow lines indicate the anticipated trajectories of the captain vehicle and the new vehicle respectively, and we can see that they are up to the intersection of the roads. We consider the following timestamps after which the vehicle is likely to update its moving parameters.

- Let t_1 (or t_2) be the timestamp that the sender vehicle (or the captain vehicle) reaches the next intersection. To maintain a high prediction accuracy, we do not predict beyond the intersection since trajectories after this point are hard to be predicted based on current movement function.

- Let t_3 be the timestamp when the distance between the two vehicles exceeds the communication range, because these two vehicles will not be in the same zone after t_3.

- Let t_4 be the possible timestamp that the new vehicle may send an update to the captain vehicle. t_4 is computed as $t_4 = t_c + \tau_u$, where t_c is the current timestamp and τ_u is the maximum interval between two consecutive updates of a member vehicle that is recorded by the captain vehicle.

The first three timestamps can be easily computed using the vehicles' current moving speed and direction. Finally, the earliest timestamp among the four: $t_f = \min(t_1 \ , \ t_2 \ , \ t_3 \ , \ t_4)$, will be selected. The time period to be considered is thus $\Delta_t = t_f - t_c$.

The second step is to compute the positions of the two vehicles at timestamps $t_c + \frac{1}{2}\Delta_t$, and t_f , respectively. These positions are relative positions on the corresponding road, i.e., the distance from the road starting point. Together with their current locations, these three sample positions are used to represent the vehicles' anticipated trajectories. The reason to choose sample points instead of using integral of moving functions is to reduce the computational complexity and satisfy the strict temporal constraints of Internet of Vehicles networks.

Finally, the similarity score of the two vehicles is defined as shown in Definition 1.

Definition 1. *Given two vehicles V_1 and V_2, let l_{c1} (l_{c2}), l_{m1} (l_{m2}), and l_{f1} (l_{f2}) denote the positions of the two vehicles at t_c, the middle timestamp and t_f, respectively. Let*

w_c, w_m and w_f be weight values, where $w_c > w_m > w_f$. The similarity score of V_1's and V_2's projected moving trajectories is computed as follows:

$$S_{V_1V_2} \triangleq \frac{\Delta_t}{w_c|l_{c1} - l_{c2}| + w_m|l_{m1} - l_{m2}| + w_f|l_{f1} - l_{f2}|} \tag{3.1}$$

The above equation integrates the effects of two factors. First, the numerator in the formula is the time interval during which the two vehicles' trajectories are considered. A higher value will be returned for vehicles that stay together for a longer time period of Δ_t. Second, the denominator in the formula is the distance between the two vehicles at the three sample timestamps. A higher similarity value will be returned for vehicles that stay closer to one another, i.e., have shorter distance. The distance between vehicles is computed as a weighted distance. The use of decreasing weights allows modeling of predicted positions that become less accurate as time passes.

After computing the similarity scores with respect to the neighboring captain vehicles, the vehicle selects the captain vehicle with the highest score and sends a join request to the captain vehicle. The join request consists of the vehicle's ID, current position, and moving speed. The respective captain vehicle will send a confirmation message to this vehicle to complete the joining process.

In case that there is no moving zone nearby, the vehicle will form a new moving zone of its own and becomes the initial captain vehicle. As time passes, this new moving zone may have more members, and the initial captain vehicle may conduct a captain vehicle re-assignment as discussed in Section 3.3.

Figure 3.1 summarizes the joining protocol. The "ZoneConstruction()" function in line 15 is discussed in the next subsection.

Protocol: Vehicle Joining Event (V)
Vehicle V:
1. Send HELLO messages to neighbors
Captain Vehicle V_c:
2. Receive the HELLO message from V
3. **If** V_c moves on the same road at the same direction as V
4. Send a response message to V
Vehicle V:
5. **While** wait time is less than τ
6. Receive response messages
7. **If** no response message is received
8. ZoneConstruction(V_c) // Form the moving zone itself
9. **Else**
10. **For** each responding captain vehicle V_c
11. Compute the similarity score $Sim(V, V_c)$
12. Send the join request to V_c with the highest score
Captain Vehicle V_c:
13. Receive the join request from V
14. Send a confirmation message to V
15. ZoneConstruction(V)

Figure 3.1: Protocol for Vehicle Joining Event

Captain Vehicle Side

Each captain vehicle needs to keep up to date information about its member vehicles in order to carry out message dissemination and zone maintenance. To achieve this, we propose two simple yet effective data structures be maintained by each captain vehicle. One is the Combined Location and Velocity Tree (CLV-tree). The other is the Leaving Event queue.

The CLV-tree is a hybrid moving object index consisting of a B+-tree and a hash table. Figure 3.2 illustrates an example CLV-tree. Each entry in the leaf node of the B+-tree stores a member vehicle's identity, the latest update timestamp t_u, location l_u and speed v_u at t_u, its index key, and estimated leaving timestamp t_{ex}. Each row in the hash table has two entries: one store the vehicle's identity, and the other stores the pointer linking to the leaf node that contains the vehicle. Both base structures are very efficient in terms of insertion and deletion, which will not impose much workload on the captain vehicle.

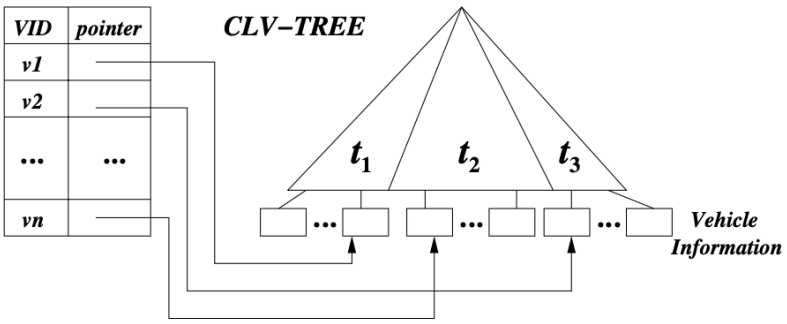


Figure 3.2: The Structure of the CLV-tree

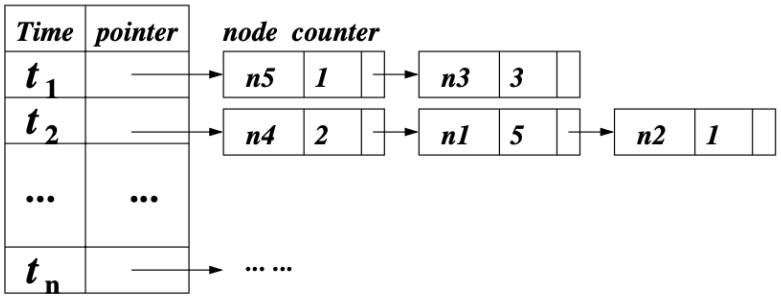


Figure 3.3: The Structure of LE Queue

The Leaving Event (LE) queue stores the estimated timestamps when member vehicles may be out of the communication range of the captain vehicle, in ascending order. As shown in Figure 3.3, each entry in the LE queue contains a leaving timestamp and a pointer to a list of nodes that contain the vehicles leaving at that timestamp. A counter is associated with the node to record the number of leaving vehicles. This LE queue is updated whenever a vehicle joins the zone or sends an update to the captain vehicle. Upon receiving the latest movement information of a vehicle, the captain vehicle computes the leaving timestamp t_{ex}. Note that t_{ex} may be infinity when the vehicle traveling at the same speed and direction as the captain vehicle. In that case, no entry in the LE queue is needed for that vehicle. The LE queue will be used during the zone maintenance phase which will be discussed later in Section 3.3. The overall protocol at the captain vehicle side is outlined in Figure 3.4.

Algorithm: ZoneConstruction (V)
Input: V is a vehicle that sends the join request
Begin Algorithm
1. Compute the index key for V
2. Insert V to the CLV-tree
3. Node$\leftarrow$ leaf node in CLV that contains V
4. Compute V's leaving time t_{ex}
5. **If** t_{ex} is a finite number then
6. **If** t_{ex} exists in LE queue
7. $L_{ex} \leftarrow$ list of nodes pointed by t_{ex}'s entry in LE
8. **If** Node exists in L_{ex}
9. Increase the counter of Node in L_{ex}
10. **else** insert Node to L_{ex}
11. **else** create a new entry for t_{ex} and Node in LE

End Algorithm

Figure 3.4: Zone Construction at Captain Vehicle Side

3.2 MOVING ZONE ROUTING PROTOCOL

We now discuss how to take advantage of the MoZo architecture to route a message to a specified destination for the example applications discussed in the introduction. In particular, suppose that a vehicle has a piece of information (I) that it would like to share with vehicles around location $l(x, y)$. The overall routing protocol is summarized in Figure 3.7. It specifically consists of the following steps.

Step 1: The sender vehicle sends a message in form of $< IDs, I, l(x, y) >$ to its captain vehicle, where IDs is the sender vehicle's unique identity, I is the message and $l(x, y)$ is the location of the message destination.

Step 2: Upon receiving the message, the captain vehicle first checks if the message destination is within its moving zone. If not, it looks for the member vehicle in its moving zone which is closest to the message destination and forwards the message to the selected member vehicle.

The algorithm for finding a good candidate vehicle for the message propagation (or propagation vehicle) is the following. The captain vehicle first computes the shortest route to the destination $l(x, y)$ using the Dijkstra algorithm, and then computes the

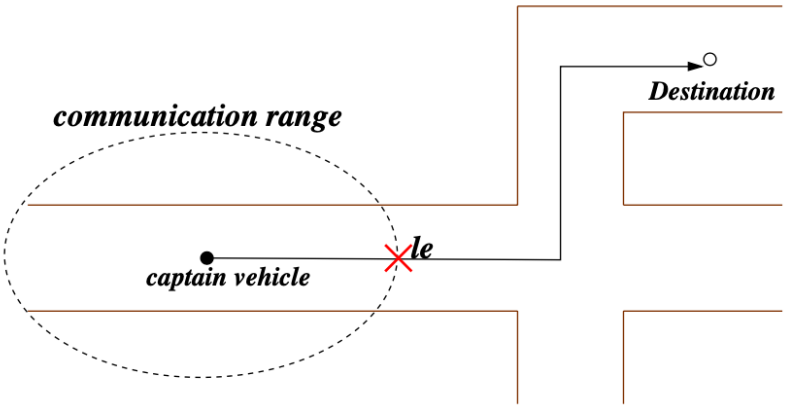


Figure 3.5: Computation of Message Delivery Route

intersection point l_e of the shortest route and its communication range as shown in Figure 3.5. Member vehicles which are around this location l_e and move towards the message destination are considered good candidate vehicles for propagating the message. To find such vehicles in the moving zone, the captain vehicle will execute a query algorithm.

Specifically, the captain vehicle generates a query key by encoding the expected location l_e and the desired moving direction. The obtained query key will be treated as the key belonging to a virtual vehicle. A virtual insertion algorithm will be conducted to locate the leaf node in the index for this virtual vehicle. Once the leaf node is found, this virtual insertion algorithm will stop, which is unlike the regular insertion algorithm that actually inserts a data to the index. The vehicles in the resulting leaf node contain similar keys to the virtual vehicle. In other words, they are likely to be near the location l_e. For further verification, the location of each vehicle at the current timestamp will be computed based on their latest moving function. The vehicle which is closest to l_e is chosen as the propagation vehicle (denoted as V_p). Figure 3.6 outlines the search algorithm. If not any candidate vehicle can be reached, the captain vehicles will wait for μ seconds and then try to find the candidate vehicle again. Up to three attempts will be made to deliver one message.

Algorithm: SearchCLV-tree(x,y,δ)
Input: (x,y) is the query location,
and δ is the moving direction
Output: propagation vehicle V_p
1. $L_v \leftarrow \emptyset$
1. For each time partition t_i in the CLV-tree
2. qKey←Encode(x,y,δ,t_i)
3. Node← CLV.root
4. While (Node is not leaf node)
5. Find the entry e in Node that contains qKey
6. Node← e
7. Add vehicles in Node to L_v
8. For all vehicles in L_v
9. $V_p \leftarrow$ the one nearest to (x,y)
End Algorithm.

Figure 3.6: CLV-tree Query Algorithm

If the message is located in the current moving zone, the captain vehicle will deliver the message to the member vehicle near the message destination. In particular, the captain vehicle will encode the message destination to the query key and employ the aforementioned query algorithm to locate the receiver vehicles.

Step 3: If the message is received by the selected propagation vehicle (V_p), V_p will be responsible for sending the message to vehicles in nearby moving zones. This operation will utilize the previously stored information about nearby captain vehicles. In particular, each vehicle keeps a list of captain vehicles that responded to the hello message sent when the vehicle requested to join a moving zone. Vehicle V_p checks its list to find the captain vehicles which have an update timestamp not earlier than the current time minus 2τ (τ is the wait time introduced in Section 3.1.2), and move toward the message destination. V_p sorts these vehicles in ascending order of their distance to the message destination. Then V_p pings these vehicles. Once V_p receives responses, V_p selects the captain vehicle that is on the top of the sorted list and sends out the message. If no response is received within τ in Equation 1, which is possible since the captain vehicles in the list may have already changed their moving functions, V_p will ping its one-hop neighbors. Based on the response from neighbors,

Protocol: Message Routing
Vehicle V_{sender}:
1. Send $M = \langle ID_s, I, l(x,y) \rangle$ to its captain vehicle V_c
Captain Vehicle V_c:
2. Receive message M
3. If $l(x,y)$ inside the zone
4. $V_{receiver} \leftarrow$SearchCLV-tree(x,y,δ)
5. Send message M to $V_{receiver}$
6. Else
7. compute intersection point $l_e(x',y')$
8. $V_p \leftarrow$SearchCLV-tree(x',y',δ)
9. Send message M to V_p
Vehicle V_p:
10. Receiver message M
11. Sort the captain vehicle list
12. For each V_c' in the captain vehicle list
13. If V_c' is available and move towards l(x,y)
14. Send message M to V_c' and done
15. Ping neighbors
16. For all responding vehicles
17. Find the V_p' closest to and move towards l(x,y)
18. Send message M to V_p'
Vehicle V_p':
19. Receive message M
20. If V_p' is a captain vehicle
21. Conduct operations from step 2
22. Else
23. Send message M to its captain vehicle V_c'
24. V_c' conduct operations from step 2
End Protocol.

Figure 3.7: Message Routing Protocol

V_p will select the one closest to the message destination as the next propagation vehicle.

Step 4: There are two cases in this step. In case a captain vehicle from a different moving zone receives the message from V_p, this captain vehicle starts the tasks of Step 2. In case a regular vehicle from a different moving zone receives the message, the vehicle will forward the message to its captain vehicle and the captain vehicle will start the tasks as per Step 2 as well.

3.3 MOVING ZONE MAINTENANCE

Zone maintenance is a continuous process that monitors the quality of the existing moving zones and conducts zone reformation accordingly to ensure the success of message routing. To maximize the information usage and reduce the communication overhead, the maintenance process leverages information collected during message routing. It includes four major tasks: (1) handling vehicle updates; (2) selecting a replacement captain vehicle; (3) zone splitting; and (4) zone merging.

3.3.1 Handling Vehicle Updates

We start with the first task. A member vehicle transmits new movement information to its captain vehicle only when its moving function (described by speed and direction) has changed dramatically. In-between two consecutive updates, the captain vehicle estimates the vehicle's location using its latest speed and direction. This update strategy can dramatically reduce the number of updates compared to existing works which require each vehicle to update its location every timestamp.

Upon receiving an update from a member vehicle, the captain vehicle sends back a ping message to confirm that it receives the updated information. Then, the captain vehicle updates the CLV-tree and the LE queue. It follows the hash table of the CLV-tree to locate the leaf node that stores the old information of the vehicle, and then insert the new information to the CLV-tree as discussed in Section 3.1.2. The captain vehicle also computes the new leaving time for the vehicle and updates the LE queue accordingly.

3.3.2 Captain Vehicle Reassignment

At a certain time point, there may be a need to find a new captain vehicle to replace the current one. For example, the current captain vehicle changes its moving function significantly and will soon be out of reach for most of its member vehicles. Or, the

captain vehicle notices that there is a member vehicle that is more suitable to be the captain than itself. The second case can be detected when the captain vehicle notices that its information is stored at the left or rightmost leaf node of the CLV-tree, which implies that its movement pattern has become less similar to its member vehicles.

Once the current captain vehicle can no longer serve this role, it conducts the captain vehicle re-assignment process. A good captain vehicle is expected to stay relatively in the center of the moving zone and moves at an average speed with most of the other member vehicles. We propose the following heuristic approach to quickly locate such a candidate captain vehicle. In particular, we take advantage of the CLV-tree instead of examining all member vehicles' movement functions. Recall that the CLV-tree has three indexing timestamps and organizes vehicles based on their relative positions on the road. The vehicle stored in the middle of the largest time partition of the CLV-tree will be selected as the new captain vehicle.

The reason for such selection is two-fold. First, the time partition that contains the largest number of member vehicles implies that these vehicles have been updated not long ago and the information will be up-to-date for a while. Second, the vehicle stored in the middle of this partition is the one that has the positions in the middle of this group of vehicles. After the candidate captain vehicle is identified, the current captain vehicle will contact the candidate vehicle and pass information about member vehicles to it. The new candidate vehicle will broadcast a message to inform current members about its new status.

3.3.3 Zone Splitting

Vehicles in the same zone have similar but still different movement functions. The possibly small difference among vehicles moving patterns is accumulated and may eventually enlarge the distance between vehicles. Consequently, after a certain time period, some vehicles in the same zone may be out of the communication range of

one another. Hence, the zone should be split periodically. The specific algorithm is the following.

The captain vehicle monitors the leaving timestamps of member vehicles using the LE queue. When a leaving timestamp t_{ex} is approaching (e.g., the current time is $t_{ex} - \tau$), the captain vehicle sums up the counters in the list linked to this timestamp, which is the total number of vehicles that will leave at t_{ex}. If this number is smaller than $\varphi * N_v$ (N_v is the number of vehicles in the zone, and φ is a percentage parameter), the captain vehicle will locate these vehicles in the CLV-tree and send a message to inform each of them that they will soon be out of the current moving zone. Here, φ is a tunable threshold. Upon receiving the notification from the captain vehicle, the leaving vehicles will prepare to execute the vehicle joining protocol (Figure 3.4) to find new moving zones at the leaving time.

Otherwise, if a large number of vehicles (i.e., more than N_l) is about to leave, the captain vehicle will split the zone into two new zones. The zone splitting leverages the CLV-tree. Considering the properties of the CLV-tree which groups vehicles with similar movement functions in nearby nodes, the captain vehicle generates the first zone containing vehicles stored in the first half nodes of each time partition in the CLV-tree, while the second zone containing the remaining vehicles. Then, the captain vehicle selects the new captain vehicle for each newly constructed moving zone using a procedure similar to the captain vehicle re-assignment process. The only difference is that the new captain vehicle for each zone is selected from the half of the CLV-tree instead of the original CLV-tree. After that, the current captain vehicle informs the two new captain vehicles of their zone members. The new captain vehicles take over the management task from here and notify their member vehicles. Zone splitting has the following benefits. A large number of vehicles leaving at the same time are re-assigned efficiently and simultaneously. This saves time in sending out individual notification messages to each leaving vehicle as well as the time to execute vehicle

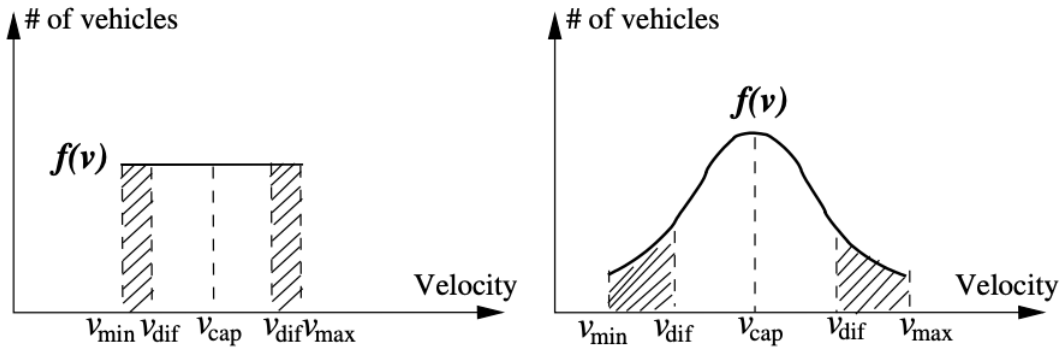


Figure 3.8: Velocity Distribution

joining protocol separately by each leaving vehicle.

The following theorem gives a formula for computing the estimated zone splitting time.

Theorem 1. *Given a moving zone, let N_v be the total number of vehicles in the zone, $f(v)$ denote the distribution function of vehicle's velocity, and let v_{min} and v_{max} be the minimum and maximum vehicle speeds of the vehicles in the zone respectively, and let v_{cap} denote the captain vehicle's speed. Let R denote the one-hop communication radius. The time interval (t_s) from the zone being constructed till zone splitting is computed in Equation 3.2:*

$$t_s = \frac{\frac{1}{2}R}{v_{dif}} \tag{3.2}$$

where, v_{dif} is obtained by solving the following equation:

$$\int_{v_{min}}^{v_{cap}-v_{dif}} f(v)dv + \int_{v_{cap}+v_{dif}}^{v_{max}} f(v)dv = \varphi N_v \tag{3.3}$$

Proof: We prove this theorem by assuming two common distributions of $f(v)$. Specifically, we consider vehicles' speed in a moving zone follows either a uniform distribution or a normal distribution as illustrated in Figure 3.8. In either distribution, according to the captain vehicle selection criteria, the captain vehicle's speed is

expected to be the mean speed so that the captain vehicle moves along with most of its member vehicles for a long time.

Recall that zone splitting occurs when there are more than φN_v of vehicles leaving, i.e., out of the communication range of the captain vehicle. Since vehicles that have speed more different from the captain vehicle will leave the zone earlier, these φN_v vehicles are likely to have speed close to either v_{min} or v_{max} as indicated by the shaded area in the figure. At the splitting moment, the number of vehicles in the shaded area would be φN_v, and hence we obtain Equation 3.3.

In Equation 3.3, the only unknown variable is v_{dif} . Solving the equation for v_{dif} , we can obtain the value of v_{dif} . Among the φN_v leaving vehicles, the one with speed closest to the captain vehicle will be the last to leave the zone. As shown in the figure, the possible speeds of the latest leaving vehicles are $v_{cap} - v_{dif}$ and $v_{cap} + v_{dif}$. Therefore, the leaving time of the vehicles with speed $v_{cap} - v_{dif}$ or $v_{cap} + v_{dif}$ is the splitting time. Given the communication range of R, we consider that in average case the latest leaving vehicles are $\frac{1}{2}R$ away from the captain vehicle. Then, we compute the splitting time as follows:

$$\begin{aligned} t_s &= \frac{\frac{1}{2}R}{v_{cap} - (v_{cap} - v_{dif})} = \frac{\frac{1}{2}R}{v_{dif}} \\ t_s &= \frac{\frac{1}{2}R}{(v_{cap} + v_{dif}) - v_{cap}} = \frac{\frac{1}{2}R}{v_{dif}} \end{aligned} \tag{3.4}$$

Theorem 1 indicates that the less the speed difference between the captain vehicle and the member vehicles, the later the zone splitting will occur. This is also in line with our algorithm that aims to find vehicles with similar moving trend.

3.3.4 Zone Merging

As time passes, some moving zones may overlap with one another. Heavily overlapped moving zones introduce unnecessary management and communication overhead. If vehicles in overlapping zones are merged into one and managed by only one captain

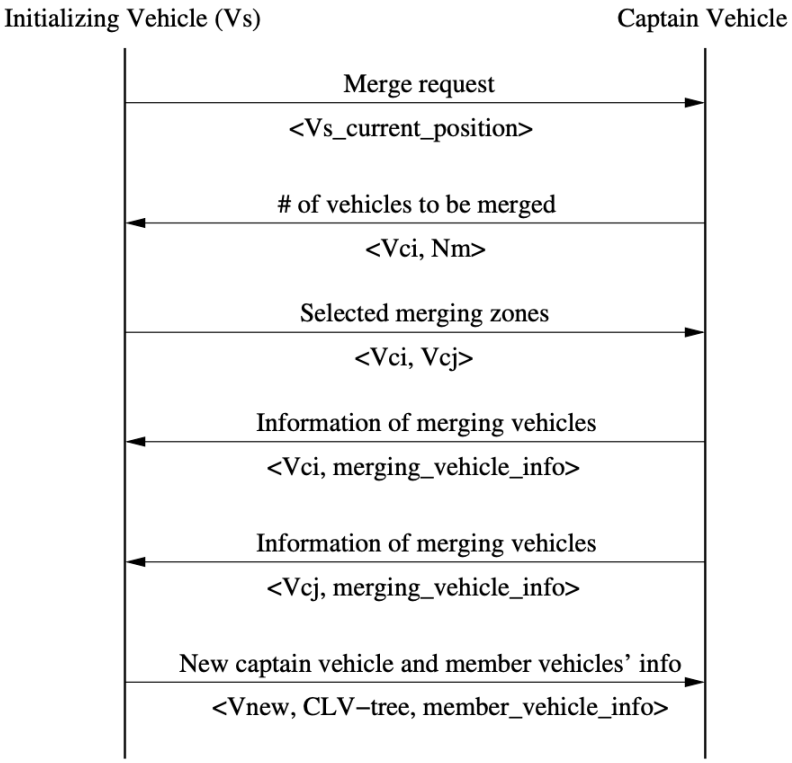


Figure 3.9: Zone Merging Protocol

vehicle, only one captain vehicle needs to respond to joining requests from other vehicles instead of multiple captain vehicles. Therefore, we propose the following zone merging protocol.

The zone merging protocol is typically initialized by a relay vehicle that detects the need for merging. We first discuss the timing and the necessary conditions for the merging. When two moving zones get closer to one another, the distance between their captain vehicles is shorter. When the distance is less than the communication range, half of the two moving zones are nearly overlapping and have the potential to be merged. This situation can be detected by the vehicles at the borders of zones utilizing information collected during routing, without any extra message communication. In particular, recall that during the message routing, the relay vehicles receive a response from their neighboring captain vehicles. According to the response, the relay vehicles know the number of nearby moving zones as well as the distance to their captain

vehicles. When a relay vehicle detects that there are more than two captain vehicles within half of its communication range and these captain vehicles have been in this range for at least two message transmissions, the relay vehicle will execute the merging protocol. Note that here we consider two conditions for merging. One is the distance between the two moving zones, and the other is the duration of the moving zones being in the close range. This is because, if a moving zone is just quickly passing by another one, the overlap of the two zones is temporary and would not affect the overall performance in the long term.

The merging protocol is sketched as follows.

- The initializing vehicle (denoted as V_s) sends a merging request including its current position to the captain vehicles that qualify the merging condition.

- Each captain vehicle (denoted as V_{c_i} which receives the merging request, computes the number of vehicles (denoted as N_m) in its zone that are within the communication range of V_s. If many vehicles in V_{c_i} 's zone are also in the communication range of V_s, that means the two zones are overlapping and they would be better off by being merged. To quantify the amount of vehicles, we use the same threshold N_l. If N_m is greater than N_l, V_{c_i} will send a message containing N_l back to V_s.

- Upon receiving the response from the captain vehicles, V_s selects two zones which contain the maximum numbers of vehicles to be merged.

- V_s broadcasts the captain vehicles of the selected zones. When the selected captain vehicles receive this message, they send the merging vehicle information to V_s and inform the remaining vehicles to start finding new zones. V_s constructs a CLV-tree to store the received vehicle information and selects the median vehicle in the tree to be the new captain vehicle.

- V_s passes all vehicle information to the new captain vehicle, and the new captain vehicle informs its members about the change.

Figure 3.9 illustrates the messages exchanged between the initializing vehicle and captain vehicles. Note that if during the execution of one merging protocol, the captain vehicles receive merging requests initialized by other relay vehicles, these requests will be ignored to avoid duplicate merging.

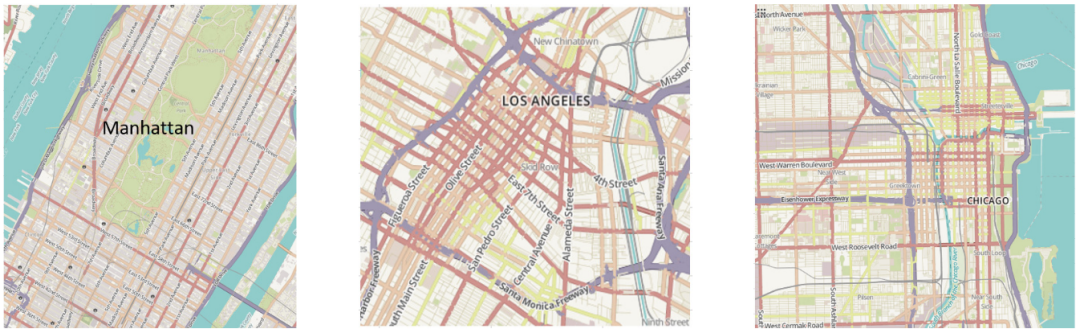
3.4 PERFORMANCE STUDIES

In this section, we first introduce the experimental settings, and then report the experimental results.

3.4.1 Experimental Settings

We compare our approach with two approaches, i.e., CBDRP [33] and Brave [34] (as described in Section 2.1.2), representing the clustering-based routing protocols and non-clustering-based routing protocols, respectively.

The experiments were conducted using the Network Simulator NS-2 (version 2.35) and vehicular mobility simulator SUMO (version 0.23.0) under Ubuntu 15.04 (64bit). The SUMO simulates the vehicles' continuous movements along the roads of three real maps as shown in Figure 3.10: Manhattan (4.5kmx5.5km), Los Angeles (5kmx4.5km), and Chicago (6kmx7km). Due to the limitation of the simulation platform, we simulate vehicles up to 1400 on each map. The vehicles' starting positions are randomly distributed on the road map. Similar to [34], vehicles move at a maximum speed of 30mile/hour inside the city and 60mile/hour on the highway. We use NS-2 and SUMO to simulate scenarios with and without traffic light controls. In both scenarios, vehicles will slow down when approaching the intersections and wait in the queue to make their turns. The vehicle behavior in the simulator is very close to that in the real life. NS-2 implements 802.11 physical and MAC models for vehicle-to-vehicle



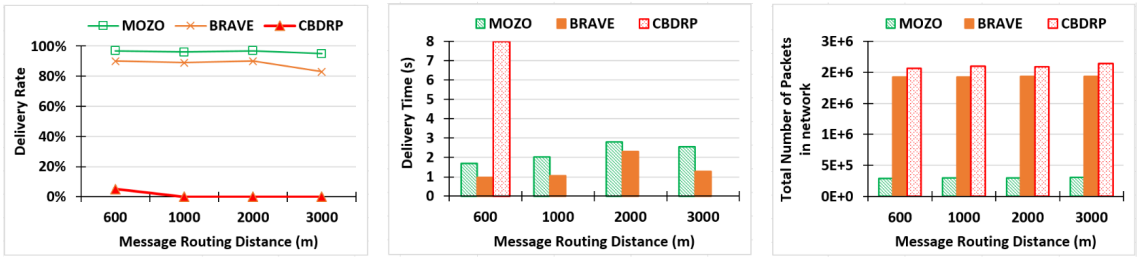
(a) Manhattan (b) Los Angeles (c) Chicago

Figure 3.10: Maps Used in the Experiments

communication and the maximum transmission range is set to 500m. Unless noted otherwise, we use the Manhattan map and set the total number of vehicles in the Internet of Vehicles networks to 800. By default, 100 512-byte messages are generated for each run of the experiments and the default distance between a message source and destination is 2000m. The simulation was run for 50 seconds to insert all vehicles and let vehicles move around on the network for a bit. After 50 seconds, vehicles issue message requests and the total simulation time is 200 seconds.

In the simulation, we vary the following parameters: (i) the distance between the message source and destination, (ii) the number of total vehicles in the Internet of Vehicles networks at the same time; (iii) the total number of messages to be delivered at the same time; (iv) the map topology. The range of these parameters will be elaborated along with the performance analysis in the following section.

The performance is measured using the following criteria: (i) message delivery time; (ii) successful delivery rate; (iii) communication overhead in terms of the total amount of messages received by all the vehicles during the whole simulation time which includes the message to be delivered, and maintenance messages for updating vehicles locations with the captain vehicles and zone merging and splitting. The reported result is the average of 10 independent runs for the same configuration.



(a) Delivery rate (b) Delivery time (c) Total Number of Packets

Figure 3.11: Effect of Message Delivery Distance

3.4.2 Experimental Results

In all approaches, the maximum attempts to deliver the message is set to 3, and each message will be kept by a vehicle for maximum 15s in the message queue for delivery. The beacon interval in the other two approaches is set to 2s as reported in [34]. In MoZo, vehicles need to send updates to the captain vehicle when they deviate from their original moving functions more than 5m/s or the time from the last update is more than 4s (Note that this maximum update interval of 4s can be increased to further reduce the message overhead in our approach). The zone splitting threshold is set to 30%, and the weight value assignment for trajectory prediction is: w_c =0.5, w_m =0.3, w_f =0.2, which have been identified to be the best parameters in most cases after multiple rounds of experiments under different scenarios.

Effect of Message Delivery Distance

In the first set of experiments, we randomly select message senders and then select message destinations that are d meters away. We vary d from 600 meters to 3000 meters. For each value of d, 100 messages are generated. As shown in Figure 3.11(a), MoZo achieves the highest delivery rate among all, BRAVE has a slightly lower delivery rate while CBDRP is the lowest. The CBDRP's delivery rate drops quickly to zero when the distance is increased to 1000m. The reason is the following. The clusters of vehicles established by MoZo are more stable than CBDRP since MoZo

models vehicle movement into the near future while CBDRP only considers vehicles' current moving directions. Moreover, CBDRP needs to explore the route first before sending the actual message. Due to the frequent changes of the clusters in CBDRP, the established route needs to be frequently maintained and may not be valid when the actual message is sent. When the route distance becomes longer, the probability of the established route being invalid increases, which severely affects the message delivery rate. Unlike CBDRP, BRAVE does not explore the whole route before sending the message. Instead, BRAVE only detects the next available message forwarder and hence it adapts to the dynamic nature of the Internet of Vehicles network topology much better than CBDRP. However, compared with MoZo that relies on clusters of vehicles for delivery, BRAVE may not always be able to find the forwarder since it is possible that an individual vehicle cannot find any forwarding vehicle on the direction to the message destination. In MoZo, the captain vehicle of the message sender has contacts with more vehicles and hence there is a higher chance of finding qualifying forwarding vehicles, leading to a higher delivery rate. This also demonstrates the advantages of clustering-based routing protocols. Another observed trend for all approaches is that the message delivery rate decreases when the message routing distance increases. This is because the longer the distance, the higher the probability that the message being dropped in the middle of the route due to various reasons such as the sparse distribution of vehicles on a certain road.

We also measure the message delivery time which is measured from the sender vehicle sending out the message until the recipient vehicle receives the message. Figure 3.11(b) shows the results. The time taken by CBDRP is longest and not reported for a distance greater than 600 meters because CBDRP has no message being delivered for long distances. As for MoZo and BRAVE, they perform similarly while MoZo delivers messages slightly slower than BRAVE. This is because the cluster maintenance and forwarder selection algorithms in MoZo are more complex than that in BRAVE. We

would like to mention that we have also conducted additional experiments by reducing the number of message attempt to 1 for MoZo. In that case, MoZo achieves shorter delivery time but a similar delivery rate as BRAVE.

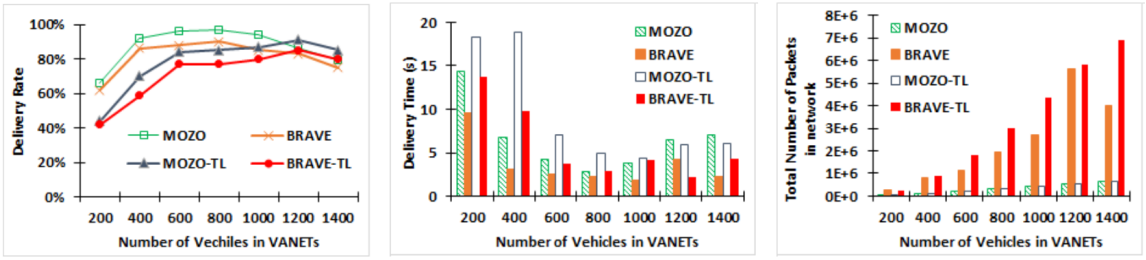
Finally, we would like to emphasize that the major advantage of MoZo is the substantially smaller number of messages transmitted in the Internet of Vehicles networks to accomplish the same task of message delivery. Figure 3.11(c) shows the total number of packets received by vehicles in the Internet of Vehicles networks in order to deliver 100 messages to the destinations at the specific distance (varying from 600m to 3km). We can see that the communication overhead in MoZo is about an order of magnitude less than the other two approaches. The CBDRP has the highest communication overhead since it needs to establish a route first and then send the actual message. The BRAVE is more efficient than CBDRP as it adopts a beaconless strategy. BRAVE does not need to discover the entire route before sending the message, but it still needs the vehicle that receives the message to broadcast to neighboring vehicles to identify the next forwarding vehicle. Unlike BRAVE, MoZo does not use broadcasting to find candidate forwarding vehicles. Instead, using MoZo, vehicles that need to send a message only need to contact their captain vehicle. MoZo uses the captain vehicle to monitor member vehicles and select forwarding candidates by keeping their moving functions and a very small number of moving function updates. Such low communication overhead accomplished by MoZo will be important for scaling Internet of Vehicles networks applications in the real world.

Effect of the Number of Vehicles

In what follows, the default road distance between a pair of the message sender and receiver is set 2000 meters (4 times of the communication range). Since the CBDRP has close to 0 delivery rate when the message delivery distance is more than 600 meters, we report the results of the MoZo and BRAVE in the remaining experiments.

Figure 3.12 shows the performance when varying the total number of vehicles from 200 to 1400 under two scenarios: (i) without traffic light control and (ii) with traffic lights (denoted as "-TL" in the figure). The reason for setting the range to [200, 1400] is the fact that this range is sufficient for revealing the underlying performance trends and where the optimal performance is observed. The range also covers the cases where the vehicle density increases from low to high. Specifically, when there are 200 vehicles in the whole network, the average vehicle density is about 0.58/100 meters; when there are 1400 on the road, the average density is about 4.08/100 meters. There are a total of 12 traffic lights on the Manhattan map.

It is worth noting here that, in all the conducted experiments, the proposed MoZo scheme achieves better delivery ratio, similar delivery time, and much smaller communication overhead compared to BRAVE, due to the same reasons discussed in the previous experiments. Here, we discuss some other interesting trends observed for both approaches. Specifically, Figure 3.12(a) shows that the delivery rate first increases with the number of vehicles and then decreases in the scenarios without traffic light control. This is because when there are very few vehicles on the roads, it may be hard to find the nearby forwarding vehicles and hence messages are dropped after the wait time. On the other hand, when there are many vehicles on the roads, there are two possible scenarios causing the low delivery rate. One possible scenario is that too many vehicles cause the traffic jam and non-uniform distribution of vehicles, resulting in fewer vehicles within communication range in the middle of the routing paths. The other scenario is that the vehicle distribution is uniform during a certain period of time, but due to the increased amount of communication among a large number of vehicles, the communication channel is jammed and hence causes some messages being dropped. Therefore, the delivery rate reaches the optimal point when vehicles on the road are relatively uniformly distributed within the communication range. However, in the case with traffic light controls, we can see that the delivery



(a) Delivery rate (b) Delivery time (c) Total Number of Packets

Figure 3.12: Effect of Number of Vehicles with and without Traffic Light Controls

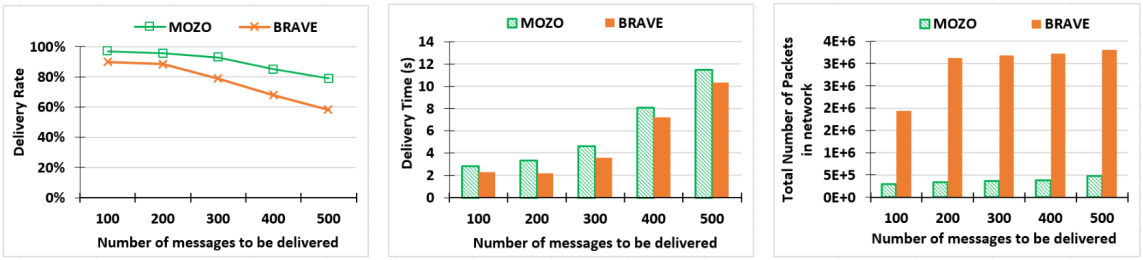
ratio does not decrease when there are a large number of vehicles in the network. This could be attributed to the traffic light controls that help direct traffic better and hence reduce some traffic congestions.

Figure 3.12(b) shows that when the number of vehicles increases, the delivery time first decreases and then increases. Such behavior is related to the delivery rate. In the scenarios when the delivery rate is low, vehicles typically need to make multiple attempts to deliver the messages or wait longer to deliver the message, which increases the overall delivery time.

Figure 3.12(c) shows that the communication overhead when one uses BRAVE increases much faster than our approach under both scenarios (with and without traffic light controls). This again proves the benefit of the proposed MoZo scheme that eliminates a large number of unnecessary message exchanges. In addition, we also see that the communication overhead increases with the increase of vehicles. The reason is straightforward. The more vehicles, the more communication among vehicles.

Effect of Number of Messages to Be Delivered

In this set of experiments, we evaluate the performance of the routing protocols by varying the number of messages to be delivered from 100 to 500 with 800 vehicles. The distance between the sender and the destination is still 2000m. As shown in Figure



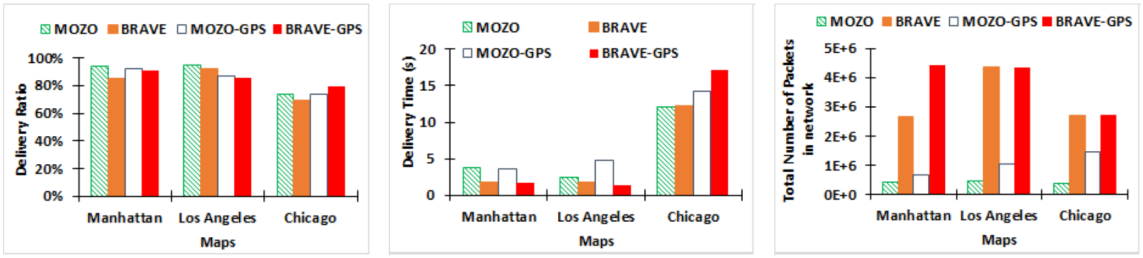
(a) Delivery rate (b) Delivery time (c) Total Number of Packets

Figure 3.13: Effect of Number of Messages to Be Delivered

3.13 (a) and (b), when the number of messages increases, the delivery rate decreases slightly. Correspondingly, the delivery time increases. The possible reason is that the more messages to be delivered, the higher probability that the communication channel becomes jammed at a certain point of the routing path and hence causes the loss of the message. We again observe that our proposed MoZo achieves a better delivery rate even when more half of vehicles (500) sent message requests simultaneously. As previously mentioned, this is because MoZo is capable of reaching more potential forwarding vehicles than BRAVE due to the use of clusters. Moreover, in Figure 3.13 (c), we observe consistent small communication overhead in MoZo, which is less than 1/10 of that in BRAVE. This demonstrates the advantages of the clustering-based strategy adopted by MoZo.

Effect of Road Topology and GPS Errors

In the last round of experiments, we study the effect of road topology as well as GPS errors on the performance of routing protocols using the three real maps shown in Figure 3.10. The total number of vehicles in each network is 1000 and the message delivery distance is 2000 meters. To simulate the GPS errors, each vehicle's position is shifted from its true position to anywhere within 15 meters (the civilian GPS' accuracy range). The results with the consideration of GPS errors are denoted using "-GPS" in Figure 3.14. Observe that the proposed MoZo scheme achieves a better delivery



(a) Delivery rate (b) Delivery time (c) Total Number of Packets

Figure 3.14: Effect of Map Topology and GPS Errors

rate, similar delivery time, and much lower communication overhead compared to the BRAVE scheme in most cases. We also observe that the map topology does not affect the performance much when the size of the map is similar, i.e., Manhattan and Los Angeles. When the map is larger (i.e., Chicago), the vehicle density decreases and hence it decreases the delivery rate while increasing the delivery time. Moreover, after introducing the GPS errors, the impact on the overall performance is very small, while our proposed MoZo has been affected slightly more than the BRAVE protocol. This is because the MoZo scheme has fewer location updates, whereas Brave requests location information more frequently and hence has more chances to correct the GPS errors.

3.5 SUMMARY

In this chapter, we propose a novel Moving Zone-Based Architecture (MoZo) that delivers messages in Internet of Vehicles networks via a self-organized moving-zone based architecture formed using pure vehicle-to-vehicle communication. We present the vehicle movement modeling and the construction of the moving zone at the member vehicle side and the captain vehicle side. We then propose a novel approach that introduces moving object modeling and indexing techniques from the theory of large moving object databases into the design of routing protocols. The moving object modeling and indexing techniques have been leveraged in various tasks including

zone construction and maintenance as well as information dissemination.

We compare our proposed routing protocol with both clustering-based approaches and non-clustering based approaches to demonstrate the advantages of our approach. As demonstrated by our experimental results, our approach significantly reduces the existing routing protocols' communication overhead to 1/10 while providing higher delivery rates compared to other existing approaches.

Chapter 4

HIGHLY EFFICIENT RANDOMIZED AUTHENTICATION FOR INTERNET OF VEHICLES

In this chapter, we present the highly efficient authentication protocol RAU$^+$. The RAU$^+$ preserves vehicles' privacy while ensuring traceability. In particular, the RAU$^+$ leverages homomorphic encryption and enables individual vehicles to easily generate a new identity for each newly established communication. These randomized identities can be verified through the collaboration of a pair of authentication servers while each authentication server would not know the real identity of the authentication requester. Figure 1.2 shows simple example scenarios. For traceability, the pair of authentication servers will execute a collaborative protocol so that the real identity of the malicious vehicle can be identified. In this way, not any single party is able to track the user.

Specifically, the RAU$^+$ provides a new type of authentication, namely aggregated authentication, which allows one vehicle to verify multiple vehicles simultaneously with a single request message to the verification server. Compared to existing authentication works [21, 22, 23, 24, 25] for Internet of Vehicles, the proposed RAU$^+$ has a number of advantages. First, the RAU$^+$ does not require any pre-generation of a long list of pseudonyms which could cause complicated ID revocation problem. Second, the RAU$^+$ does not need the server to generate pseudonyms every time which prevents pseudonym generators, such as road-side units or group managers (i.e., peer

vehicles) used in other works, from tracking the vehicles. Third, the RAU$^+$ does not require the availability of road-side units which are not widely available in the real world due to deployment cost. Fourth, the RAU$^+$ is efficient which meets the real-time constraints in Internet of Vehicles applications well. A more detailed security analysis and performance studies will be presented in the remaining of this chapter.

4.1 PRELIMINARY

For a better understanding, we first briefly review the additive homomorphic probabilistic public key encryption (HEnc$^+$) system which is the building block of the proposed authentication system.

Let E_{pk} and D_{sk} be the encryption and decryption functions in an HEnc$^+$ system with public key pk and secret key sk. Without sk, no one can discover x from $E_{pk}(x)$ in polynomial time. When the context is clear, we will omit pk and sk from the notations of the encryption and decryption functions. The HEnc$^+$ system has the following properties:

- The encryption function is additive homomorphic in that the product of the encryptions of x_1 and x_2 produces the encryption of $x_1 + x_2$.

$$E(x_1) * E(x_2) = E(x_1 + x_2) \tag{4.1}$$

- Given a constant c and $E(x)$:

$$E(x)^c = E(c * x) \tag{4.2}$$

- The encryption function has semantic security as defined in [85], i.e., a set of ciphertexts do not provide additional information about the plain-text to an adversary. E.g., suppose that y_1 and y_2 are the ciphertexts generated by

performing the encryptions of x at different times using the same key, there is very high probability that $y_1 \neq y_2$, but $D(y_1) = D(y_2)$ holds.

Any HEnc$^+$ system is applicable, but in this dissertation, we adopt Paillier's public-key homomorphic encryption system [86] for the actual implementation due to its efficiency. In Paillier, the public key is $N = p * q$, where p and q are large primes with similar size, and they are private information. In general, the size of N should be at least 1,024 bits. The encryption function is defined as follows for x:

$$E(x, r) = (N + 1)^x * r^N \mod N^2$$

where r is randomly chosen from $\mathbb{Z}^*_{N^2}$. Note that the encryption function is only based on the public key, and the group $\mathbb{Z}^*_{N^2}$ contains the elements from $\mathbb{Z}_{N^2} = \{0, 1, 2, \ldots, N^2 - 1\}$ which are co-prime to N^2. Since r is randomly selected each time a value is encrypted, $E(x, r_1) \neq E(x, r_2)$ if $r_1 \neq r_2$. On the other hand, $D(E(x, r_1)) = D(E(x, r_2)) = x$ regardless the value of r_1 and r_2.

4.2 RAU$^+$: ADVANCED RANDOMIZED AUTHENTICATION SYSTEM

In this section, we first present the system overview and then the threat model, followed by the details of the protocols.

4.2.1 An Overview of the System

The RAU$^+$ system consists of two major types of entities: users and authentication servers. Users are passengers in the car who would like to communicate with others in Internet of Vehicles. There are two authentication servers residing in two different clouds, which are Registration Server (RS), and Verification Server (VS). The two servers collaborate with each other to conduct privacy-preserving user authentica-

tions, and hence none of them would be able to track the user alone. We assume users can communicate with the servers via Internet.

When designing each specific protocol, we aim to achieve the following security requirements of the anonymous authentication system:

- Prevent users from being tracked: This includes two aspects. First, the real identity of a legitimate user should not be known by other peer users. Other peer users and any single authentication server would not be able to track the users' movement (i.e., a series of locations that the user has been to) by linking multiple authentication messages to the same user.

- Providing traceability: If necessary and under lawful request, the two authentication servers will be able to collaboratively reveal the real identity of a malicious user.

The proposed authentication system has three main phases: (1) user registration, (2) user authentication, and (3) identity tracing. At the beginning, users register at the RS server. The RS server shares an initial randomized authentication ID and secret key of each user with the VS server. Whenever a user wants to communicate with other vehicles, he/she can randomly generate pseudo identities and secret keys which can be verified by the VS server to prove the user's ownership of pseudonym. If there is any dispute, the two servers will conduct a tracing protocol to figure out the real identity of a malicious user. It is worth noting that our protocols ensure that all communications between vehicles and registration servers are via secure channels. The establishment of secure channels are integrated with the registration protocol. The fundamental technology adopted is Paillier encryption scheme [86]. The detailed steps in each phase are elaborated in the following subsections.

4.2.2 Threat Model

In our system, we adopt the following threat or adversary model.

Like all existing work, we assume that the two authentication servers adopted in our authentication system are semi-honest. That is they follow the prescribed procedures of the proposed protocols and do not collude. This is a legitimate assumption if the two servers reside in two different well-known cloud platforms such as Amazon EC2 and Microsoft Azure which have no financial incentive to collude to damage their reputations. We assume that servers are under security protection of existing approaches [87, 88] that will not be compromised by the attackers. The users can be malicious. A malicious user can impersonate another user. When the users are malicious, we will consider three common attacks under most authentication systems: man-in-the-middle, replay and credential sharing.

4.2.3 User Registration

The registration phase is for a vehicle (i.e., Internet of Vehicles user) to be authenticated by the server and obtain an initial random ID (RID_u^0), a random secret key (RK_u^0), a randomization seed γ_u and a randomization interval τ_u, based on which the user would be able to self-generate any number of random IDs and keys to communicate with other peer vehicles later on. The registration protocol is illustrated in Figure 4.1.

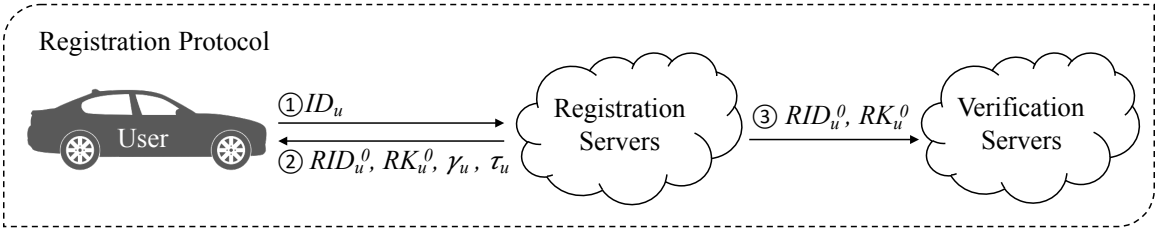


Figure 4.1: User Registration (Please note that we show only the message content here. All messages are in fact encrypted.)

As a one-time setup, the registration server (RS) generates its own public-private

key pair using the Paillier encryption scheme [86], and the public key is known by all entities. A new user can join the Internet of Vehicles system at any moment. To register, a user u sends certain identification information (ID_u) such as driver license number [1] to the registration server (RS) via the secure channel. If needed, the RS server can further verify u's identification information via a third party (e.g., an agency who performs background check for credit card applications). How to achieve robust identify verification is out of the scope of this dissertation, but the RS server can use any existing solutions.

The RS server computes an initial randomized authentication ID (RID_u^0) and secret key (RK_u^0) for user u as follows:

$$RID_u^0 = E(ID_u, r_u^0) \tag{4.3}$$

$$RK_u^0 = E(RK_u^b, r_u^0) \tag{4.4}$$

where $E(X, r)$ is a Paillier encryption of X with a random number r using the RS' public key, and RK_u^b is a basic secret key randomly generated by RS server. RID_u^0 and RK_u^0 are sent to both user u and the verification server (VS). Since RID_u^0 is encrypted using the RS server's public key, only the RS server is able to decrypt it and reveal the real identity of the user. The actual identity of the user is always kept secret from the verification server during the lifetime of the user.

After user u is registered, both the RS and VS servers store the user's initial randomized authentication ID RID_u^0 and random secret key RK_u^0 in their local databases. The plain texts of the real identities are discarded by the RS server to prevent attackers from hacking the system and stealing the sensitive information.

[1] Here we use driver license number for illustration only. Other information that verifies a user's identity such as SSN can also be used.

4.2.4 User Authentication

We now proceed to present the two-way authentication protocol for user i and j to verify each other's legitimacy. The authentication protocol consists of two main phases: (i) identity validation and (ii) generation of a new randomized authentication ID and secret key. If needed, user i can perform concurrent authentication sessions with multiple users by doing aggregated identity validation to reduce the communication overhead in the Internet of Vehicles networks and the workload of the verification server.

Single Identity Validation

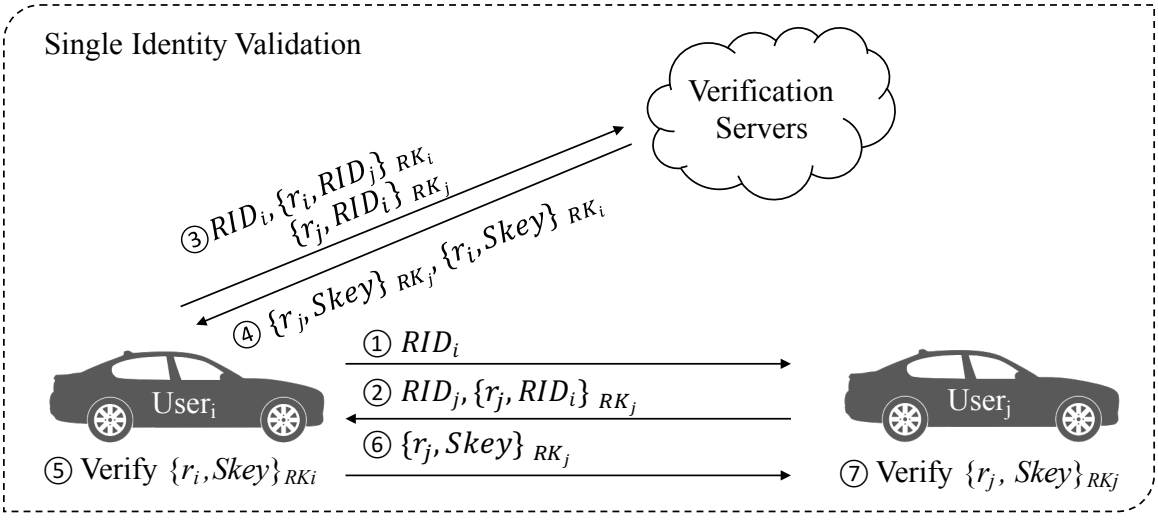


Figure 4.2: Single Identity Validation

After the registration, the user i obtained one initial randomized ID RID_u^0 and secret key RK_u^0 from the registration server. User i can use them directly for the follow-up communication with other vehicles, or generate other new randomized IDs and secret keys based on this initial randomized ID and secret key using the protocol in Section 4.2.4. Let RID_i denote the randomized ID that user i will use to authenticate himself with user j. The following steps will be performed (illustrated in Figure 4.2):

- **Generating random numbers**

  User i first generates a random number r_i for replay and man-in-the-middle attacks prevention purpose (discussed in Section 4.3). For two-way authentication, after receiving user i's randomized ID, user j will generates the random number r_j as well.

- **Exchanging the randomized ID and generating verification data**

  Before sending the verification data to the VS server, user i first sends its RID_i to user j. Then, user j reply his/her RID_j to user i. Besides, user j encrypts his/her verification data $E_j =< r_j, RID_i >$ by RK_j and sends it to user i. After receiving this reply message, user i generates his/her encrypted verification data $E_i =< r_i, RID_j >$ by RK_i as well.

- **Verifying the randomized ID and secret key**

  After exchanging the randomized IDs, user i sends RID_i, E_i and E_j to the VS server. The VS server will first check whether RID_i is exists in its database and still effective. Then, by decrypting E_i using RK_i and decrypting E_j using RK_j, the VS server verifies that E_i, E_j are generated by user i and j respectively. By further comparing the RID_i in E_j and RID_j in E_i in the verification data, the VS server can verify that both user i and user j are willing to authenticate each other's identity. Finally, the VS server verifies that RID_j is exists in its database and still effective as well.

- **Vehicle-side verification and secure channel establishment**

  Once the VS server verifies user i and user j's verification data, it will randomly generates a session key $Skey$, and sends the encrypted response data $R_i =< Skey, r_i >$ (encrypted by RK_i) and $R_j =< Skey, r_j >$ (encrypted by RK_j) to user i. User i will then forward R_j to user j. Both user i and user j decrypt R using their own RK and check if the random number is correct to avoid the

message replay attack. After that, the secure channel between user i and j can be established by using the session key $Skey$. Any attacker cannot obtain this session key because they do not know RK_i and RK_j.

It is worth noting that the RAU$^+$ protocols naturally support identity ownership verification to prevent identity sharing, which will be discussed in Section 4.3.

Aggregated Identity Validation

The single identity validation protocol deals with a pair of vehicles' authentication each time. In some cases, we may need a more efficient method to authenticate a group of vehicles. For example, in a safety enhancement application, when a vehicle i moves toward an intersection, it may need to know the speed and moving direction of vehicles approaching the intersection. At this point, vehicle i needs to authenticate itself to the vehicles near the same intersection, and these vehicles may need to verify vehicle i as well. If using the single identity validation protocol, a large number of verification requests may jam the VS server. Therefore, we propose an aggregated identity validation protocol for a vehicle to authenticate with multiple surrounding vehicles simultaneously. This protocol can reduce both the number of connection establishments and the total transfered bytes between users and the VS server.

To conduct the aggregated identity validation, vehicle i first sends a message to communicate with the group of vehicles that would like to verify its identity. After receiving the verification data from them, vehicle i aggregates the information and sends it to the VS server for verification. Finally, vehicle i forwards the verification results received from the VS server to the group of the vehicles. Figure 4.3 illustrates the protocol:

- **Generating verification data**

  User i first generates a random number r_i, then sends RID_i to users j, k, ..., z.

User x who receives this request generates encrypted verification data E_x and sends it back to user i.

- **Aggregating and submitting verification data**

  After receives RIDs and verification datas $E_j, E_k, ..., E_z$, user i: (1) generates the aggregated verification data $AE_i =< r_i, RID_j, RID_k, ..., RID_z >$ and encrypts it using RK_i, and (2) sends $RID_i, AE_i, E_j, E_k, ..., E_z$ to VS for verification. Considering that some responses may not be able to transfer to user i on time due to network delay or other reasons, after sends out request messages, user i will wait T_{w1} to collect, aggregate and send out verification datas. Then, he/she will wait another T_{w2} to deal with other delayed responses and sends out verification datas to the VS server one more time. After that, user i will cancel all the remaining authentication processes.

- **Verifying the randomized IDs and secret keys**

  Upon receiving the verification datas, the VS server firstly decrypts AE_i using RK_i to check that AE_i is actually generated by user i. Then, it will check whether $RID_i, RID_j, ..., RID_z$ are exist in its database and still effective. After that, the VS server will decrypts $E_j, E_k, ..., E_z$, verifies that those encrypted verification data are generated by correct RK. If the verification succeeds, the VS server will generates the session key $Skey_x$ for channel between user i and x, encrypts the verification results $R_x =< r_x, Skey_x >$ by RK_x, and then sends them back to user i. Besides, the VS generates aggregated verification result $AR_i =< r_i, Skey_j, Skey_k, ..., Skey_z >$, encrypts it by RK_i and sends it to user i as well.

- **Vehicle-side verification and secure channel establishment**

  Once the VS server sends back the verification results, user i first keeps AR_i, and then forwards $R_j, R_k, ..., R_z$ to users $j, k, ..., z$, respectively. After that, each

user x decrypts the verification result using his/her own random secret key RK_x and verifies the random number r_x. By doing that, user x can verifies that the result is generated by the VS server and obtains the session key $Skey_x$ from the decrypted messages to establish the secure channel between user i and x.

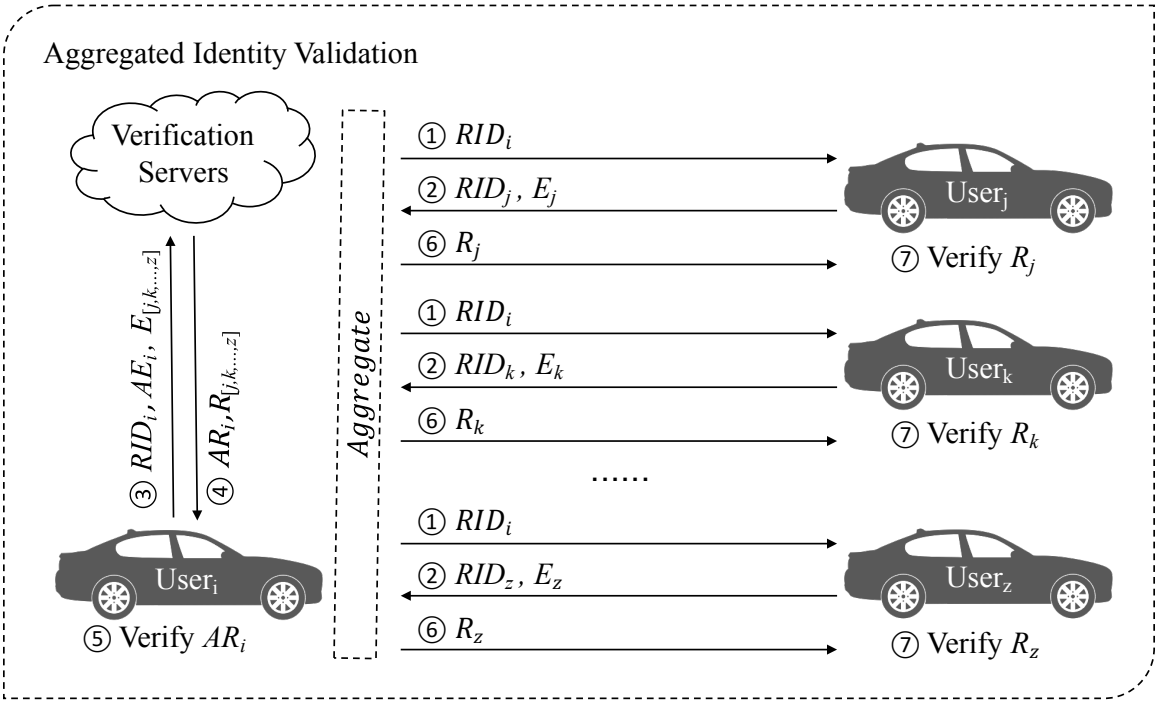


Figure 4.3: Aggregated Identity Validation

Generation of Randomized Authentication ID and Secret Key

In RAU$^+$, each randomized authentication ID is only used once or in a short duration so that a user's moving trajectory will not be tracked by any party in the system. To self-generate the k^{th} new randomized ID and secret key, the user i can compute them using Equation 4.5 and Equation 4.6.

$$RID_i^k = RID_i^{k-1} * E(0, r_i^k) \tag{4.5}$$

$$RK_i^k = RK_i^{k-1} * E(0, r_i^k) \tag{4.6}$$

Based on the addition property of Homomorphic encryption (Equation 4.1), the new randomized ID is again the encryption of the real identity which can be deduced as follows:

$$RID_i^k = E(ID_i, r_i^{k-1}) * E(0, r_i^k) = E(ID_i + 0, r')$$

It is worth mentioning that by leveraging this addition property, the generation of the new randomized ID is more efficient than directly encrypting the real identity again.

The challenge here is how to let the verification server (VS) know that this new ID is valid so that the authentication can be performed. Recall that the VS maintains a list of valid randomized IDs received from the RS server, but the VS is not able to compute any new ones. Only the RS server knows the real identity of the vehicle but the RS server is not in charge of verification. A straightforward method is to let the user inform the RS server about the new ID and then let the RS server forward it to the VS server, which however will disclose the user's locations to the RS server. Therefore, we propose a synchronization approach that avoids the communication between the user and the RS server during the authentication. The key idea is to let the RS server generate a randomization seed γ_i and a randomization interval τ_i for user i at the registration phase. Every τ_i time, the RS server will be able to generate the same random number as user i did based on the seed γ_i. Therefore, the RS server can directly compute the randomization ID and secret key of i using Equation 4.5 without communicating with user i and without knowing there is an authentication request. After that, RS sends these randomly permuted up-to-date random ID and key to the VS server. To further enhance security, the the old random IDs and keys are discarded by the RS server.

When the VS server received the new copy of the random IDs and keys (randomly permuted by the RS server), it would not be able to link each new ID to its previous version. It is worth noting that the randomization time intervals are not necessary to be the same for all users, and each time the RS server generates the same random ID

and secret key as user i did based on the seed γ_i, both RS server and user i can generate a same new τ_i as well so that the attacker cannot associate different RIDs with their time intervals. Since it's hard for users synchronize their local clock with RS and VS perfectly, there are some special designs in our protocol to overcome this problem: (1) After received by VS, every randomized ID will be discarded after a period of time T_d (T_d is a pre-defined time peroid that for every user i, T_d always larger than τ_i). (2) Users will resynchronize their local clock after every Registration/Authentication phase and adjust their local clock $T_d/2$ slower than RS/VS. These two designs ensure that as long as the time difference between servers and users is less than $T_d/2$, the RID will be validated successfully.

4.2.5 Identity Tracing

In some applications, disputes may occur due to various reasons. Sometimes a third-party law enforcement authority may want to know immediately the real identity of a suspect user who is undergoing an authentication. Sometimes there may be a need to discover the authentication history of a suspect user. Thus, we propose both real-time identity tracing and historical identity tracing.

The real-time identity tracing is easy to achieve. The law enforcement authority submits the tracing request that contains the suspect user's randomized authentication ID to either the VS server or the RS server. If the request is received by the VS server, the VS server will forward the suspect user's randomized ID to the RS server. Upon receiving the suspect user's randomized ID, the RS server uses its private key to decrypt the randomized ID and reports the real identity to the law enforcement authority.

In terms of historical identity tracing, the law enforcement authority captured one randomized ID of the suspect user and wants to know the authentication history of the user to figure out the user's behavior in the network. The law enforcement

authority sends the randomized ID of the suspect user to both RS and VS server. The RS server maintains a list of authentication history of all users. For example, each user has a list of randomized authentication IDs that have been or are planned to be used. The VS server maintains all valid authentication IDs.

First, the RS server finds a match in a user's list. If there is a match, the list of randomized IDs will be provided to the law enforcement authority which will subsequently send these IDs to VS. The VS will return the authority the verification records of these randomized IDs. Based on the location of the suspect, the authority may learn when and where the suspect has been before. To provide this kind of historical tracing, the only thing needs to be changed is that the RS and VS servers need more memory space to store previously used randomized IDs and verification records. In addition, when the VS server performs identity validation, it needs to make sure, old IDs cannot be used again. These modifications can be easily incorporated into our current scheme.

4.2.6 Credential or Identity Revocation

Identity revocation is very efficient in our system. Once a suspicious user is confirmed to be malicious, the RS and the VS servers just need to remove this user's randomized ID and secret key from their databases. Any subsequent authentication request for this malicious user will fail as no matching record will be found by the server any more.

4.2.7 Non-transferability

The RAU$^+$ system can prevent users from sharing their credentials with other users without any additional workload. If user i wants to lend his credential to user j, he has to share the RID_i and RK_i with user j. If any user requests a random number r from RS, encrypts r by RK_i and sends them back to RS, the RS will return the user's

real identity and other valuable information encrypted by RK_i as well. Because of that, if user i shares his credential with user j, user j will be able to use all of user i's credential and obtain user i's valuable information. This effectively discourages user i from sharing his credential. Besides, since the VS doesn't know the randomization seed and interval, it's impossible for VS to get user's valuable information.

4.3 SECURITY ANALYSIS

In this section, we will analyze the security and privacy features of the proposed RAU$^+$ system.

4.3.1 Unforgeability

Our authentication protocol guarantees that no one can use the identity that does not belong to him/her. Under the assumptions that the private key is kept securely at the RS server side, the only option left for the attacker to impersonate legitimate users is to exploit their randomized authentication IDs. There are several possible ways for an attacker to obtain a randomized authentication ID of a user. However, we show in the following that the attacker would not be able to use this ID as its own for authentication purpose.

The Replay Attack

Although an attacker may obtain another user's valid authentication ID during authentication, the attacker cannot directly use the received authentication ID again since each ID is associated with a random secure key RK. Without knowing this secure key, the attacker cannot obtain the session key and finish the authentication successfully. The encrypted verification data also cannot be replayed because it contains the random number which is only knowing by the user. If any attacker replays

this data, he will not be able to decrypt the session key in the response data from VS or pass the user side verification of the random number.

The Man-in-the-Middle Attack

We now discuss the case when an attacker attempts to forward a new (or never used) randomized IDs from user i to user j whereby the attacker tries to pretend to be user i. Even though these IDs have not been used by the real owner, the attacker will not be able to pass the user authentication phase. This is because the attacker does not know the random secret key RK_i belonging to the true ID owner–user i; hence, the attacker cannot encrypt the validation data using the correct secret key (step 3 in Figure 3). As a result, the VS server will not be able to decrypt the authentication request using the correct RK_i either, and hence will reject this authentication.

4.3.2 Full Privacy Preservation

Our authentication protocol provides full privacy preservation in that it guarantees both server-wise and peer-wise privacy for the users in terms of both anonymity and unlinkability. Considering the peer-wise privacy, under the proposed protocol, a user always self-generates a new randomized authentication ID when establishing a new communication session. Since the encryption scheme we adopted is semantically secure [86], it is computationally infeasible for peer users to know the real identity of others and to link different communication sessions or randomized authentication IDs to the same user as long as the size of the encryption key is large enough (such as 1024 bit).

As for the VS server, it does not have the secret key to decrypt the randomized IDs stored in its database, and hence it does not know the real identity of the user who submits authentication request (again here we assume the encryption key size is sufficiently large). Due to the fact that the RIDs, the RKs and the renew times

of them are randomized, and all the RIDs have the same effective duration, VS cannot link different RIDs and RKs to the same user. As for the RS server, since it does not handle any authentication request that contains randomized IDs during the authentication phase, the RS server does not know which user is sending the authentication request. Therefore, our protocol prevents the RS server from tracking the locations of the users.

4.3.3 Prevention of Credential Sharing

Credential sharing means that a legitimate user i gives his random ID RID_i to another user j so that user j may try to communicate with others using RID_i. Our proposed RAU$^+$ system prevents such credential sharing as long as the legitimate user does not share his personal identifiable information which is encrypted by RK_i. Since personal identifiable information (such as SSN) would be very sensitive, there would not be enough incentives for a legitimate user to share such sensitive private information even with their friends. By keeping the secret key RK and personal identifiable information secret, user j who obtained i's random ID would not be able to pass the validation phase which requires the knowledge of the randomized secret key RK.

4.3.4 Traceability

Traceability refers to the ability to reveal a user's real identity requested by the law authorities. This is a seemingly conflicting requirement with respect to the privacy preservation goal of our system. We achieve this by proposing the collaborative identity tracing protocol as presented in Section 4.2.5. The identity tracing protocol is capable of revealing a suspect user's real identity and his/her whole authentication history to the law authorities without violating the privacy of other legitimate users.

4.4 PERFORMANCE STUDIES

In this section, we first introduce the experimental settings and then report the experimental results.

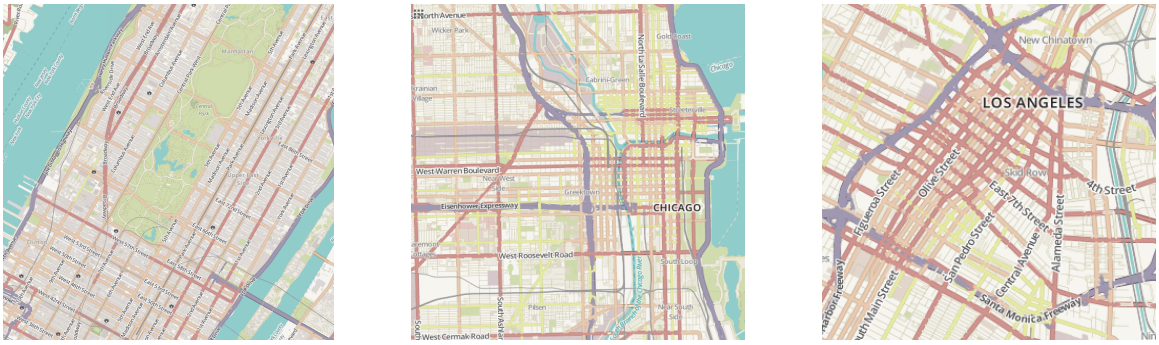
4.4.1 Experimental Settings

We compare our proposed RAU$^+$ with two existing approaches: (i) the RAU protocol [20] on randomized authentication; (ii) Anonymous Credential (AC) system [71] which is the most related work with similar security goals to our work.

It is worth noting that our protocols do not require the vehicles to be equipped with high performance computing equipments. The following hardware specification is used to simulate the whole system including network simulation, vehicles movement simulation and server/vehicle side computing simulation, but not the hardware carried by vehicles. We implemented the RAU$^+$ authentication protocol in Java and C++ languages and run the tests on a PC with Intel Core i7 CPU 2.6 Ghz and 16 GB memory. We evaluate the efficiency of the total authentication process in terms of communication and computational cost. The transmission and propagation delays were simulated as well.

The authentication protocols are implemented by NetBeans with JDK 8 to evaluate the time performance on the vehicle side and the server side. We use the vehicular mobility simulator SUMO (v0.23.0) to simulate the vehicles' movements in three real maps as shown in Figure 4.4: Manhattan (4.5km * 5.5km), Chicago (6km * 7km) and Los Angeles (5km * 4.5km).

The number of vehicles is ranging from 200 to 1000. Unless noted, we use the Manhattan map and set the number of vehicles to 800. The speed of vehicles is 30 miles per hour inside the city, and 60 miles per hour on the highway. In the SUMO simulation, vehicles will slow down when approaching an intersection and stop if there



(a) Manhattan (b) Chicago (c) Los Angeles

Figure 4.4: Real Maps Used in the Simulations

is a traffic jam. The simulation of network is conducted by the Network Simulator NS-3 (v3.26). The maximum transmission range in NS-3 is set to 100 meters and the transmission rate of the wireless channel among vehicles is 6Mbps. The network delay between vehicles and server is 20ms. The VS and RS are connected through 100Mbps wired network. The total simulation time is 120 seconds. Firstly, the simulation runs for 15 seconds to insert all vehicles. Then, vehicles begin the registration phase. At a random time after the 60th seconds, each vehicle selects several nearby vehicles within 80 meters to start the aggregated authentication process. The default value of the maximum number of the selected vehicles is 10.

4.4.2 Experimental Results

In all approaches, each vehicle performs one registration process and one group authentication operation whereby the vehicle randomly selects several nearby vehicles to start a two-way authentication between himself and other vehicles. For example, for vehicle i, it first starts the registration phase at 8th second, then chose 10 nearby vehicles to proves that it is a legitimate user. Those 10 vehicles also prove to vehicle i that they are legitimate users. The operations occur at random times chosen from a same random sequence so that in the simulations of the three approaches, each vehicle executes the same operation at the same time. In this way, we ensure a fair

comparison.

Performance of User Registration

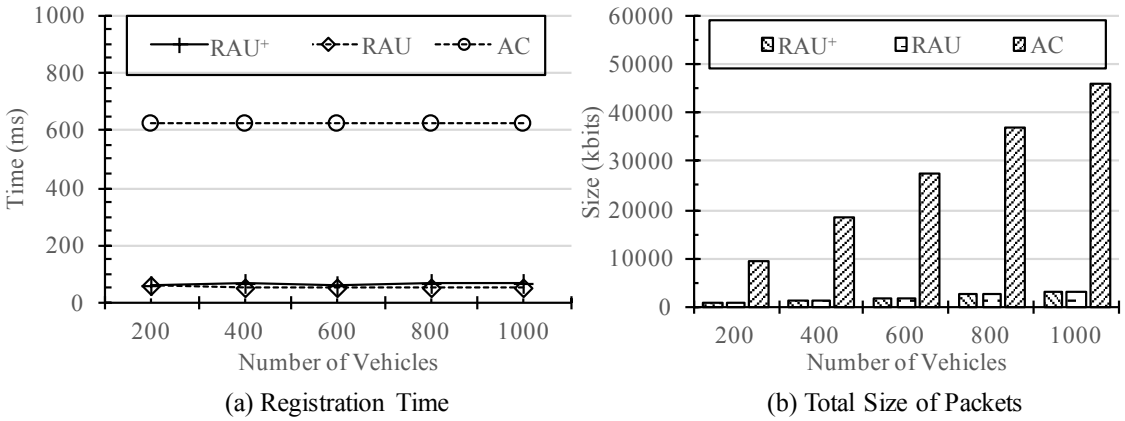


Figure 4.5: Performance of User Registration

The main computational cost involved in the user registration is the generation of the initial pseudonym for the new user. In RAU$^+$ and RAU, each randomized ID is 1024 bits, and for AC system, the length of RSA modulus is 1024 bits. Figure 4.5 (a) presents the time performance of user registration. We can observe that the RAU is slightly faster than RAU$^+$, and the RAU$^+$ is significantly faster than AC system. This is because the RAU$^+$ needs to perform one more operation, i.e., the generation of the randomized secret key RK associated with the RID, and the AC system needs more exponential computations and much more message exchanges which leads to a longer network delay. Specifically, as for the RAU$^+$ and RAU protocols, without considering network propagation delay, the randomized ID and secret key generation time is less than **23ms** per user. Moreover, besides the standard secure channel establishment, the RAU$^+$ and RAU require only one round of message exchange between the user and the RS server, whereby the user sends his personal identifiable information to the RS server and the RS server sends back the initial random ID to the user. However, in the AC system, the pseudonym and credential generation need many time-consuming

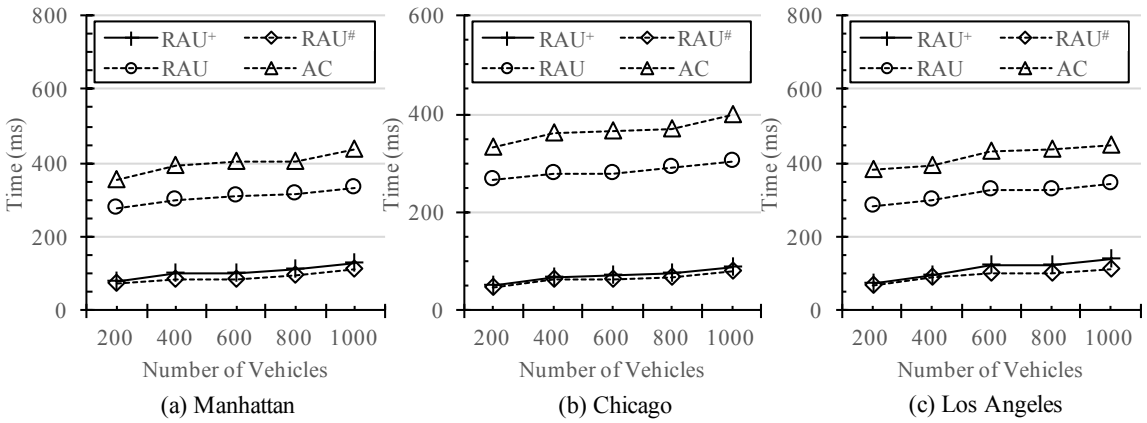


Figure 4.6: Time Performance of User Authentication

exponential computations, and 6 rounds or message exchange between the user and the RS server which takes about **380ms** excluding the network delay. Figure 4.5 (b) shows the total size of transfered packets for registration in network, from which we can see that our RAU$^+$ protocol can significantly reduce the network overhead.

Performance of User Authentication

Being the best among the exiting solutions, the Anonymous Credential (AC) system [72] achieves most criteria for anonymous authentication as our proposed RAU$^+$ system. However, as discussed in the related work and the security analysis, the AC is not perfectly suitable for Internet of Vehicles applications. One reason is that the AC system assumes that the communication channel is already secure, which is however really challenging for two vehicles before authentication. In this section, we compare the efficiency of (1) the full RAU$^+$ protocol with secure channel establishment, (2) the RAU$^\#$, the part of RAU$^+$ protocol without aggregated identity authentication, (3) the RAU protocol, and (4) the AC system without secure channel establishment.

Figure 4.6 reports the running time of each protocol by varying the number of vehicles from 200 to 1000 in three maps. Observe that our proposed RAU$^+$ protocol is significantly faster than the other two approaches. To better understand such behav-

ior, let us review the main steps in each protocol. For each round of authentication, in AC system, the user i shows a single credential to user j, and user j shows his/her single credential to user i as well. The computational complexity of the total authentication process is 22 exponentiations as the AC is using the zero-knowledge proof. In our RAU$^+$ protocol, user i and user j need to: (1) exchange their pseudonyms, (2) prove their ownerships of their randomized identity RIDs, and (3) establish a secure channel between them. It is worth noting that the first phase of our RAU$^+$ protocol already offers the same functionality as the AC protocol. With the additional two phases, RAU$^+$ provides more security guarantee than the AC system while still more than 10 times faster than the AC protocol. Compared to the previous RAU protocol, the RAU$^+$ protocol is also more efficient because the RAU$^+$ integrates the ownership validation in the authentication protocol via the random secret key RK and hence reduces the computational and communication time. The RAU$^+$ protocol is slightly slower than RAU$^\#$ because the aggregated identity verification needs to wait a group of users' responses before sending the aggregate verification data to VS. However, the RAU$^+$ protocol can significantly reduce the overhead of the network and the VS server, which will be discussed below.

Next, we compare the communication complexity in terms of the amount of messages and the total size exchanged throughout the protocol. Figure 4.7 shows the number and total size of transmitted packets of user authentication by varying the maps and the number of vehicles requesting simultaneous authentication. As shown in Figure 4.7 (a), the total number of packets transmitted during the authentication phase (including the communication with the servers) in RAU$^+$ is significantly less than the AC system, the original RAU protocol and the RAU$^\#$ protocol. In our RAU$^+$ protocol, there are total $(1 + 1.5 * k)$ rounds of communication including one round between u_i and the VS server and 1.5 rounds between u_i and each of its k neighboring users $(u_1, u_2, ..., u_k)$ for simultaneously two-way authentication. However, in the AC

system, there are three rounds of communication between each pair of vehicles and the total is $3k$. Without aggregated identity authentication, the RAU$^{\#}$ needs $2.5k$ rounds of communcations for k pairs of users.

Moreover, the size of the messages in the RAU^{+} protocol is $(12k + 3)l$ bits (l is the key size) which is also smaller than AC system ($30kl$ bits), the RAU ($22kl$ bits) and the RAU$^{\#}$ ($15kl$ bits). As expecited, in Figure 4.7 (b), we observe that the advantage of aggregating authentication in RAU^{+} has become more prominent with the increase of the vehicles requesting for authentication simultaneously. Since part of the communication cost in the RAU and RAU^{+} involves the server, we take a further look at it in Figure 4.7 (c). Observe that the aggregated identity authentication in RAU^{+} has largely reduced the number of authentication requests so that the number of connections established between user and VS can be reduced. All of these is because the aggregated identity authentication (1) avoids sending duplicate messages occurring in a group of pair-wise authentication, (2) reduces the number of connection establishment between users and the VS server (only one establishment is needed), and (3) transfers less bits between users and the VS server ($(6k + 3)l$ bits), which is both smaller than the RAU ($12kl$ bits) and the $RAU^{\#}$ ($9kl$ bits).

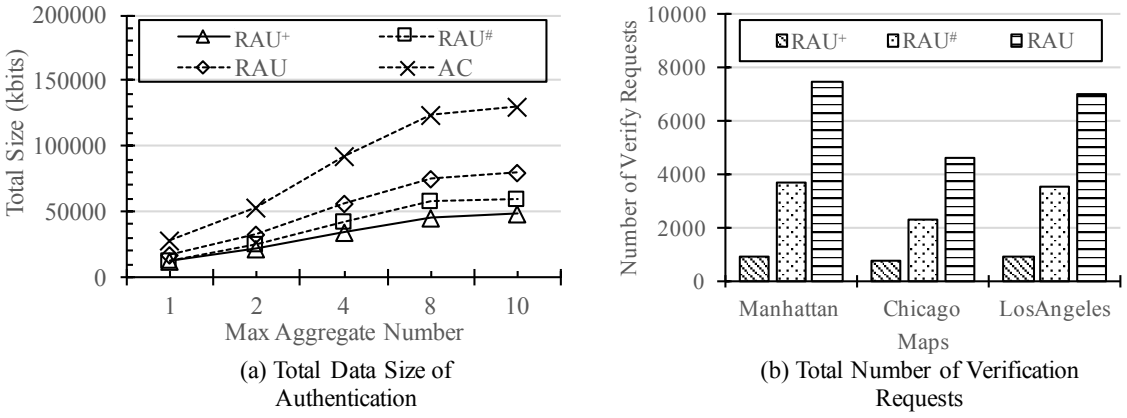


Figure 4.7: Transmission Performance of User Authentication

Performance of Identity Tracing

Our proposed RAU$^+$ protocol has a nice feature of providing user tracing in terms of any dispute while the AC system does not. There are two types of tracing available in the RAU$^+$ system: (i) the real-time identity tracing; and (ii) the historical identity tracing. Table 4.1 reports the running time of tracing a single user. It is not surprising to see that the real-time identity tracing is much more efficient than the historical identity tracing. This is because the real-time identity tracing only needs to recover a single user ID whereas the historical identity tracing needs to check the disputed randomized ID against a list of randomized IDs which have been used by the same user in the past.

Table 4.1: Identity Tracing Time

Protocol	**Real-time Tracing**	**Historical Tracing**
RAU$^+$	1.8ms	3.49s

4.5 SUMMARY

In this chapter, we present a highly efficient randomized authentication system in Internet of Vehicles environment, namely RAU$^+$. The proposed RAU$^+$ protocols leverage the properties of a semantically secure public-key additive homomorphic encryption scheme. The proposed RAU$^+$ system overcomes shortcomings in other existing works, and achieves a set of desired properties including unforgeability, full privacy preservation, identity tracing and being secure against various types of attacks. The experiment results demonstrates that our proposed RAU$^+$ protocol is more efficient than other protocols, and can effectively reduce the overhead of the network and the servers' workload. By further adjustment and improvement, our protocol may be extended to IoT schemes or other application scenarios.

Chapter 5

SECURE AND LIGHTWEIGHT IDENTITY MANAGEMENT FOR INTERNET OF VEHICLES WITH MINIMUM INFRASTRUCTURE RELIANCE

In this chapter, we propose a Secure and Lightweight Identity Management (SLIM) mechanism for vehicle-to-vehicle communications to minimize the reliance on the infrastructure support.

5.1 THREAT MODEL AND DESIGN GOALS

Threat Model

Our proposed SLIM scheme aims to defend the following attacks as some are also pointed out in [89]:

- **Eavesdropping Attack:** The attacker can eavesdrop on any communication in the Internet of Vehicles networks.
- **Impersonate Attack:** Attackers may pretend to be another vehicle in the network to fool the others.
- **Movement Tracking:** An adversary who constantly eavesdrops messages exchanged in Internet of Vehicles networks and therefore tracks other vehicles' traveling routes.
- **Message Replay Attack:** Replay the valid messages to disturb the traffic.

- **Man-In-The-Middle Attack:** Attackers may relay and alter the messages during the transmission between two vehicles who believe they are communicating with each other directly.

- **Denial of Service(DoS) Attack:** The attacker may send a large amount of junk messages to prevent legitimate users from accessing other vehicles' computing and communication resources.

Design Goals

Our proposed SLIM aims to achieve the following design goals:

- **Data Origin Authentication and Integrity:** Every exchanged message should be unaltered during the delivery and can be authenticated by the receiver. Authentication and integrity of the messages must be verified [90].

- **Anonymous User Authentication:** The process of authenticating the vehicle should not reveal the vehicle's real identity to other peer vehicles.

- **Vehicle Traceability:** In case there is any dispute, the authority should be able to reveal the real identity of the suspect vehicle.

- **Message Unlinkability:** Observers can not link messages observed in different groups to the same vehicle so that observers cannot track other vehicles.

We list description of the notations used throughout this chapter in Table 5.1.

5.2 SECURE AND LIGHTWEIGHT IDENTITY MANAGEMENT SCHEME

In this section, we present the details of the proposed Secure and Lightweight Identity Management (SLIM) scheme in Internet of Vehicles networks. The SLIM scheme is built upon moving zones self-organized by vehicles using the zone forming protocols in [26]. Each self-organized moving zone is formed by a group of vehicles with similar

Table 5.1: Notations and Definitions

Notation	Definition
v_i	Vehicle i
ID_i	Vehicle's Identity Encrypted by DMV_{pubkey}
CAU^j	Captain Authentication Unit of Zone j
GIT_i	Global Identity Token for Vehicle i
LIT_i^j	Local Identity Token for Vehicle i for a specific Zone j
$\{...\}_{key}$	Encryption using key
$Sign(...)_{key}$	Generate Signature using key
$key_{i,k}$	Session key between two Vehicles v_i and v_k
R_i	Role of Vehicles i (government car, emergence car, etc.)
r_i	Nonce generated randomly by CAU^j for Vehicle v_i

movement patterns or social interest. These moving zones are dynamic and will change as vehicles move. Each zone has a captain vehicle which helps pass messages among member vehicles. In SLIM, we assign the captain vehicle a new task to serve as the authentication unit and name it captain authentication unit (CAU) similar to [27]. The SLIM scheme ensures that the vehicles' identities are verifiable to each other while preventing any vehicle in the Internet of Vehicles networks including the CAU from seeing the true identities of other vehicles.

The SLIM scheme is composed of three phases: *Registration*, *Inner-zone Authentication* and *P*eer-to-Peer Communication. During the registration phase, a vehicle will contact Identity Management Center (IDMC) to be verified and then obtain a global identity that does not reveal the vehicle's real identity. During the authentication phase, vehicles will send its global identity to the CAU to obtain a local identity. This local identity is later used for communication among vehicles in the same moving zone. In what follows, we elaborate the detailed algorithms for generating the global and local identities.

5.2.1 Registration

Procedure 1 presents the registration phase of our proposed scheme. This phase is executed only once for each new vehicle joining the Internet of Vehicles networks.

Procedure 1 Registration

Each Vehicle v_i executes the following steps

Generate global key pair $Gpubkey_i$ and $Gprikey_i$

Encrypt $ID_i = \{Identity_i, VIN\}_{DMV_{pubkey}}$

Generate signature $rs_i = Sign(ID_i, Gpubkey_i)_{Gprikey_i}$

$v_i \xrightarrow{\{ID_i||Gpubkey_i||rs_i\}_{IDMC_{pubkey}}} IDMC$

IDMC executes the following steps

Decrypt using $IDMC_{prikey}$

Verify signature rs_i using $Gpubkey_i$

Verify v_i's identity ID_i with DMV

IF v_i's identity is verified

Generate a random number r_i

Generate signature $s_i = Sign(r_i, R_i, Gpubkey_i)_{IDMC_{prikey}}$

Generate $GIT_i = \langle r_i, R_i, Gpubkey_i, s_i \rangle$

$IDMC \xrightarrow{\{GIT_i\}_{Gpubkey_i}} v_i$

ELSE Reject Request

Each Vehicle v_i executes the following steps

Verify signature s_i using $IDMC_{pubkey}$ and obtain GIT_i

The first time that a vehicle v_i logs onto the Internet of Vehicles networks, it will communicate with the IDMC to obtain a global identity token GIT. Specifically, before logging onto the Internet of Vehicles networks, v_i need to generate a pair of global keys $Gpubkey_i$ and $Gprikey_i$, encrypt its ID_i using DMV_{pubkey} and generates a digital signature rs_i. The first time that v_i enters, it sends a encrypted registration request to IDMC.

When receives the registration request, the IDMC decrypts it and verifies v_i's signature rs_i to make sure that the message is sent by v_i who owns $Gprikey_i$. Then the IDMC verifies the received encrypted identity information ID_i with DMV (Department of Motor Vehicles). Since the verification message can only be decrypted by DMV, the IDMC will only know whether v_i has a valid identity but don't know what this true identity is. In this way, the vehicles' privacy is also protected against the IDMC. Only if the validation result is true, for v_i, the IDMC generates a global identity token GIT_i. Upon receiving the GIT_i, v_i decrypts and verifies it to ensure that the GIT_i was issued by the IDMC and has not been altered. At this point, v_i

Procedure 2 Joining Existence Zone j

Each Vehicle v_i **executes the following steps**

Generate local key pair $Lpubkey_i^j$ and $Lprikey_i^j$

Generate signature $vs_i = Sign(GIT_i, Lpubkey_i^j)_{Gprikey_i^j}$

$v_i \xrightarrow{\{GIT_i||Lprikey_i^j||vs_i\}_{CAU^j_{pubkey}}} CAU^j$

CAU^j **executes the following steps**

Decrypt using CAU^j_{prikey}

Verify IDMC's signature on GIT_i

Verify signature vs_i using $Gpubkey_i^j$

IF verified

Generate timestamp T_c

Generate signature $cs_i = Sign(R_i, T_c, Lpubkey_i^j)_{CAU^j_{prikey}}$

Generate $LIT_i^j = \langle R_i, r_i, Lpubkey_i^j, cs_i \rangle$

$CAU^j \xrightarrow{\{LIT_i^j\}_{Lpubkey_i^j}} v_i$

ELSE Reject Request

Each Vehicle v_i **executes the following steps**

Verify timestamp T_c and signature cs_i using CAU_{pubkey}

Obtain LIT_i^j

has a global identity token that does not reveal any sensitive information about its actual identity.

5.2.2 Inner-Zone Authentication

After vehicle v_i obtains the global identity token, it can use this token to be authenticated in any moving zone that it belongs to during the movement. Specifically, when v_i joins a new moving zone Z_j, it will contact the captain authentication unit CAU^j to obtain a local identity token LIT_i^j. This local identity LIT_i^j will only be used within this zone. When v_i moves to another zone, it will need to seek another local identity so that it would not be easily tracked by observers. Procedure 2 illustrates how the local identity tokens are issued.

In Procedure 2, vehicle v_i first randomly generates a pair of local keys $Lpubkey_i^j$ and $Lprikey_i^j$ during any free time before v_i wants to enter a new zone so that the generation procedure would not affect the authentication time. Then, v_i computes a

Procedure 3 Peer-to-Peer Communication (v_i, v_k) within Zone j

Vehicle v_i executes the following steps

$v_i \xrightarrow{LIT_i^j} v_k$

Vehicle v_k executes the following steps

Verify CAU^j's signature on LIT_i^j

IF Verified

Generate session key $key_{i,k}$

Generate signature $ts_k = Sign(LIT_k^j,\ key_{i,k})_{Lprikey_k^j}$

$v_k \xrightarrow{\{LIT_k^j,\ key_{i,k},\ ts_k\}_{Lpubkey_i^j}} v_i$

ELSE Reject Request

Vehicle v_i executes the following steps

Decrypt using $Lprikey_i^j$ and extract LIT_k^j

Verify CAU^j's signature on LIT_k^j

IF Verified

v_i and v_k authenticate each other and both share the session key $key_{i,k}$

ELSE Reject Request

digital signature vs_i and sends a join request to CAU_j.

When receives the join request, the CAU^j decrypts it using its private key, extracts v_i's global identity token GIT_i and verifies IDMC's signature s_i in GIT_i to validate this global identity. The CAU^j also verifies v_i's signature vs_i to ensure that this GIT_i belongs to v_i. Only if the verification results are true, the CAU^j generates a randomized number r_i, issues a local identity LIT_i^j and sends this local identity to v_i.

Once receives the response from CAU^j, vehicle v_i will extract and verify the authenticity and integrity of this response. At this point, v_i has obtained a local identity token LIT_i^j until it leaves current moving zone.

5.2.3 Peer-to-Peer Communications

After vehicle v_i obtains the local identity LIT_i^j, it is now ready to securely communicate with any other vehicles in the same zone. As illustrated in Procedure 3, in particular, when v_i intends to establish a fresh session communication channel with another vehicle (say v_k), the first step is to generate a session key between them. For

this, v_i first send a session request along with its local identity LIT_i^j to v_k. When receives this request, v_k first verify the validity of v_i's local identity by checking the CAU^j signature in LIT_i^j and generate a random session key $key_{i,k}$ and a signature ts_k. Then, encrypts the following message using v_i's local public key so that attackers can neither eavesdrop or modify it: $\{LIT_k^j, key_{i,k}, ts_k\}$. After that, sends it to v_i. Once receives this response, v_i will decrypt the message and verify the identity of v_k in the same way that v_k just did.

After the above peer-to-peer authentication, v_i and v_k are able to communicate securely by encrypting the messages using the session key in the following form: $\{LIT_{v_i}^j, msg\}_{key_{i,k}}$. It is worth noting that as long as v_i and v_k stay communicating with each other, the peer-to-peer authentication between these two vehicles just need to conducted once. If more security is desired, the two vehicles can change the session keys over time.

To sum up, the SLIM scheme involves one-time communication between the IDMC and the vehicle, and vehicles can have different local identities in different moving zones for privacy preserving.

5.3 SECURITY ANALYSIS

In this section, we analyze the reactions of our proposed SLIM scheme to common attacks.

Eavesdropping Attack: With our SLIM scheme in place, any outside attacker cannot obtain any sensitive identity information of vehicles by eavesdropping the Internet of Vehicles networks. When sending the registration request to IDMC, the vehicle's identity information was encrypted by DMV_{pubkey}, and the whole request was encrypted by $IDMC_{pubkey}$ too. It is impossible for any attacker to decrypt the registration message because they do not have the required private keys. For the same reason, outside attackers cannot eavesdrop any valuable private information during

the peer-to-peer authentication and communication.

Considering inside attackers, the IDMC can only verify v_i's identity with DMV without knowing any detail personal information because only DMV can extract the private information from ID_i. Moreover, the $CAUs$ cannot eavesdrop their member vehicles' communication either. This is because CAUs do not know the session keys established between member vehicles.

Impersonation Attack: In SLIM, a vehicle v_i cannot be impersonated because no other vehicles knows v_i's $Gprikey_i$ or $Lprikey_i$. Thus, it is impossible for other vehicles to generate v_i's signature or decrypt the messages received by v_i. More specifically, during the peer-to-peer communication, suppose that an attacker knows v_i's LIT_i and plans to impersonate v_i. When the attacker sends this local identity to another vehicle v_k in the same moving zone, v_k will generate a session key encrypted using vehicle v_i's $Lpubkey_i$ and send it back to the attacker. Since the attacker does not possess vehicle v_i's local private key, it would not be able to decrypt the message received from v_k and hence cannot pretend to be v_i.

Movement Tracking: As previously mentioned, any outside attacker cannot see sensitive ID information by eavesdropping the network that is using the SLIM scheme. Thus, outsiders would not be able to find out the traveling routes of vehicles. Considering the insider attacks, we separate the cases of CAU and member vehicles. Any member vehicle only knows the local identities of vehicles in the same zone that communicates with it, but does not know the global identity of these vehicles. Thus, member vehicles may only be able to track the vehicles who are communicating with it within the same zone, but will not be able to keep tracking the same vehicle which has moved to another zone. Note that member vehicles even do not know if they are communicating with the same vehicle that they have met in the past since the same vehicle will use a different local identity in a differen zone.

As for CAUs who know the global identities of its member vehicles, the CAU

may be able to track the same vehicle whenever the vehicle enters its moving zone. However, this risk can be mitigated by a proper CAU election which forbids a vehicle to serve as a CAU continuously and frequently. This can be achieved since member vehicles know the CAU's global identity and they can verify if the same vehicle wants to serve CAU again when they move along together from one zone to another. On the other hand, a normal CAU may not want to serve as CAU frequently either since in that way it exposes its global identities for a long time for others to track.

Message Replay Attack: In our system, if an attacker replays a registration or inner-zone authentication request sent by vehicle v_i, it would not be able to decrypt the response messages from IDMC or CAU witout konwing the private keys obtained by v_i. Also, if an attacker replays a message sent by v_i to v_j, it would not be able to know the content of the response sent back by v_j since the attacker does not know the session key used by v_i and v_j. As a result, the attacker would not be able to continue meaningful conversation with v_j further.

Man-In-The-Middle Attack: All the messages in our SLIM scheme are either signed or encrypted, which prevents attackers to modify or reuse. Specifically, the global identity GIT_i cannot be modified by other vehicles because it's signed by the IDMC. Vehicle v_i's inner-zone authentication request can only be verified by $Gpubkey_i$ which is included in GIT_i. Thus, any other entity cannot modify this request and regenerate the signature without knowing v_i's $Gprikey_i$. Also, attackers cannot put itself into the communication between vehicles. When v_i communicates with the IDMC, its message is encrypted using the IDMC's public key and hence only the IDMC can open it. When the IDMC responds to v_i, the message is encrypted using v_i's public key and hence only v_i can open the message. The case with the CAU is similar.

During the peer-to-peer communication, when v_k received the local identity LIT_i^j from v_i, a possible attack that it may conduct is to pass this local identity to another

v_l and try to play a middle role in this communication. However, the v_l's response will be encrypted by $Lpubkey_i^j$. Since v_k does not know the local private key of v_i, v_k would not be able to decrypt the message sent back by v_l and obtain the session key inside the message. Also, v_k cannot generate new response to v_l since v_k is not able to produce v_i's signature.

Denial of Service (DoS) Attack: In the SLIM system, outside attackers' messages can be filtered because they do not have valid identity tokens. When they try to replay the registration or inner-zone authentication request, the IDMC or CAUs can reject those messages because the *Gpubkey* or *Lpubkey* have been used in the previous requests. The inside attackers also will eventually be caught as they have been authenticated and will leave all these malicious behavior in records.

5.4 PERFORMANCE STUDIES

We now move to evaluate SLIM's efficiency in the authentication process. We compare its performance with the most related vehicle-to-vehicle-based authentication scheme – PAIM [27]. The implementations are conducted using a machine equipped with an Intel Core i7 at 2.6 GHz with 16 GB of RAM running UNIX system. Each procedure in the program has been run 1000 times and the mean values are reported in milliseconds.

The network simulation was conducted using the Network Simulator NS-3 (version 3.26) and vehicular mobility simulator SUMO (version 0.23.0). Vehicles' movements along with the main roads of three real maps: Manhattan (4.5 km x 5.5 km), Chicago (6 km x 7km) and Los Angeles (5 km x 4.5 km). Vehicles' speed ranging from 30 to 60 miles/hour. In NS-3, the maximum transmission range is set to 100 meters, the network delay is 10ms, and the wireless transmission rate is 6Mbps. Unless noted, otherwise we use the Manhattan map and set the number of vehicles to 800. The simulation was run for 15 seconds to insert all vehicles, then begin registration phase.

After 60 seconds, at random time, each vehicle become group manager respectively, select up to 10 vehicles over a range of 80 meters and start Inner-Zone Authentication. The simulation time is 120 seconds.

5.4.1 Registration Phase Performance

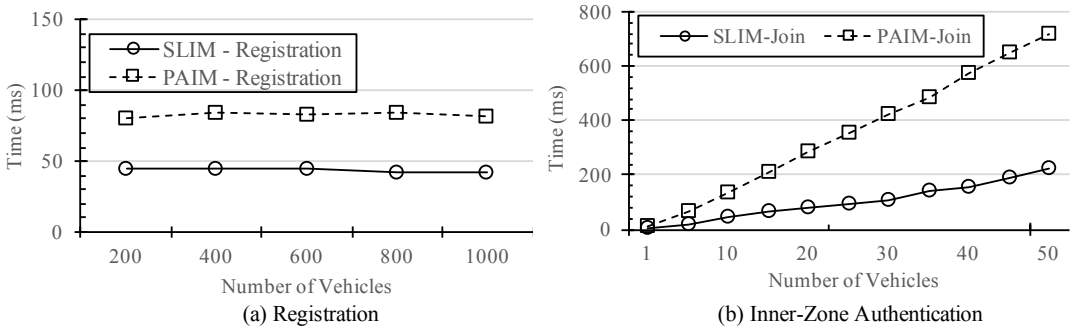


Figure 5.1: Time Performance During Registration

In the first round of experiments, we measure the average time needed for a vehicle to register at the IDMC using the SLIM and the PAIM scheme respectively. As shown in Fig. 5.1(a), the average registration time per vehicle under SLIM is about 40 ms, which was faster than PAIM's 80 ms. This could be attributed to the efficient protocol of SLIM which does not need extra rounds to establish a session key between the IDMC and the vehicle. Note that the vehicles' private/public key pairs in SLIM scheme can be gerentated during the vehicle's free time and hence would not affect any authentication performance.

5.4.2 Inner-Zone Authentication Phase Performance

Next, we measure the performance of the inner-zone authentication for both the SLIM and the PAIM schemes. Fig. 5.1(b) shows the total inner-zone authentication time at the CAU side when the number of vehicles in its zone varies from 1 to 50. Observe that SLIM is clearly faster than the PAIM. With the increase of the number of vehicles in the zone, the performance gap between the two approaches widened.

Specifically, when there are 50 vehicles, our proposed SLIM scheme is more than 3 times faster than PAIM. In Fig. 5.2, with the increasing of the number of vehicles, the time raises due to more packets, larger network delay and heavier workload, and our SLIM protocol is obviously performs better than PAIM. This is because the SLIM scheme requires much fewer rounds of message exchanges to generate a local identity for a vehicle as shown in Fig. 5.3.

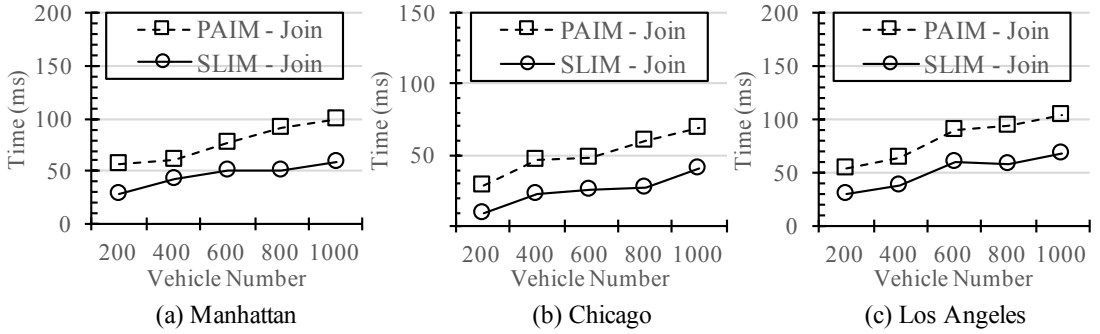


Figure 5.2: Time Performance During Inner-Zone Authentication on Three Maps

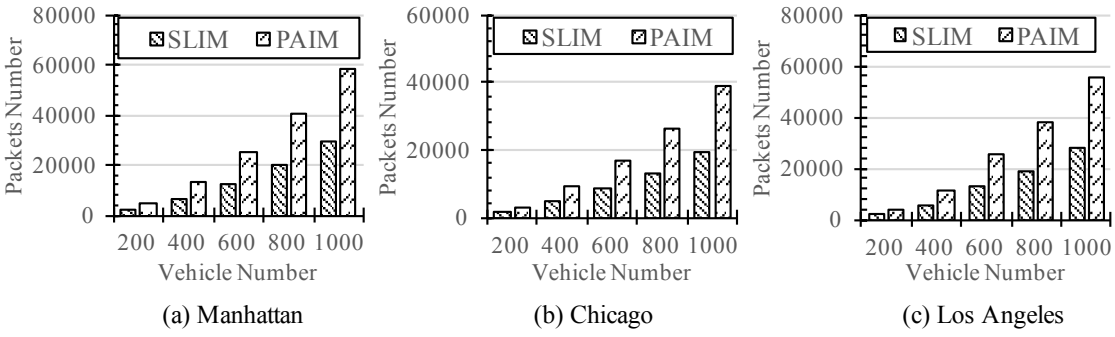


Figure 5.3: Communication Cost During Inner-Zone Authentication

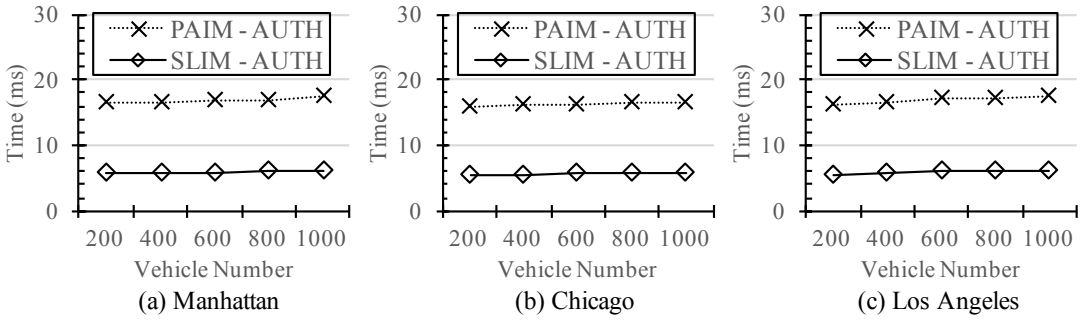


Figure 5.4: Communication Cost During Peer-to-Peer Communication

5.4.3 Peer-to-Peer Communication Performance

Finally, we compare the efficiency of the two approaches in terms of peer-to-peer communication. Fig 5.4 presents the time performance of these two protocols on three maps. In SLIM scheme, the time taken for two vehicles to mutually validate each other's local identity is only 3.5ms excluding network delay. However, in PAIM, since two vehicles need to conduct the zero-knowledge proof which could take as long as 13.6ms, it is clearly much slower than the SLIM scheme.

5.5 SUMMARY

In this chapter, we proposed a lightweight privacy preserving vehicular authentication protocol SLIM, which alleviates the reliance on infrastructure support. The SLIM scheme leverages the PKI in an efficient way to create anonymous global identity and then local identities for vehicles to preserve their privacy when communicating with other vehicles. The SLIM is not only robust against various types of attacks but also very efficient as compared to the state-of-the-art.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

This dissertation firstly presents a novel moving-zone-based architecture and a corresponding routing protocol for message dissemination in the Internet of Vehicles networks by using vehicle-to-vehicle communications only (i.e., without using vehicle-to-infrastructure communications). To the best of our knowledge, this is the first study that applies moving object techniques to vehicular networks. The moving object modeling and indexing techniques have been leveraged in various tasks including zone construction and maintenance as well as information dissemination. The proposed approach greatly reduces communication overhead and improves message delivery rate compared to other existing approaches, which makes it possible for vehicles to exchange important data such as traffic conditions, vehicle status, and travel plans with each other.

On the basis of the message routing protocol, this dissertation proposes a highly efficient randomized authentication protocol, RAU+, which leverages homomorphic encryption and enables individual vehicles to easily generate a new randomized identity for each newly established communication while each authentication server would not know their real identities. The RAU+ is very efficient since it does not require any pre-generation of a long list of pseudonyms, or the server to generate pseudonyms every time. It prevents vehicles from being tracked by any single party including peer

vehicles, service providers, authentication servers, and other infrastructure. Meanwhile, the proposed protocol also provides traceability in case of any dispute. Both security analysis and experimental study have been conducted to demonstrate the superiority of the proposed protocol compared to other existing works.

Aiming at minimizing the reliance on infrastructure support, this dissertation further proposes a Secure and Lightweight Identity Management (SLIM) mechanism for vehicle-to-vehicle communications. The proposed approach is built upon self-organized groups of vehicles that take turns to serve as captain authentication units to provide temporary local identities for member vehicles. While ensuring the vehicles' identities are verifiable to each other, the proposed approach can also prevent any vehicle in the Internet of Vehicles networks including the captain authentication unit from seeing the true identities of other vehicles. The proposed authentication protocols leverage the public key infrastructure in a way that the key generation workload is distributed over time and hence achieves authentication efficiency during vehicle-to-vehicle communication. Compared to the previous related work, the proposed SLIM mechanism is more secure in that it can defend against more types of attacks in the Internet of Vehicles networks, and is more efficient in that it requires a much shorter response time for identity verification between vehicles.

6.2 FUTURE WORK

The underlying foundation of the information management and security protection for the Internet of Vehicles is the organization of continuously moving vehicles to form a network that enables these independent vehicles to collaborate with one another efficiently, securely, and stably. Although many approaches have been proposed to form vehicle groups and support collaborations across vehicles, research is needed to enhance the security of the Internet of Vehicles. When joining the Internet of Vehicles environment, vehicles should be able to exchange hello messages with neighboring

vehicles, register themselves with cluster head / RSUs / TA, and obtain necessary information such as pseudonyms, key pairs, random seeds, etc. Vehicles may need to join groups formed by RSUs or other vehicles, basic topology needs to be built, and important secure connections should be established. Lots of work are still need to be done to provide well-designed architectures to support the management operations of the Internet of Vehicles. In the meantime, it is essential to preserve privacy while designing the architectures, protocols, and mechanisms of the Internet of Vehicles environment. If the sensitive information is not well protected, the attacker may be able to steal them and perform attacks not only to the Internet of Vehicles systems but also to other systems. In traditional scenarios, there are many existing methods to reduce the risk of privacy leakage. However, in the Internet of Vehicles systems, the infrastructures might be expensive and not expected to be deployed widely soon. To reduce the reliance on the infrastructures, the authentication procedure should be carried out via pure peer-to-peer communication, or at least, by relying on the connectivity provided by pure peer-to-peer communication. Thus, it is hard to ensure all the nodes are honest. To address these issues, in future work, we will extend the research to design the privacy preservation mechanism to reduce the risk of sensitive information leakage while ensuring the usability and reliability of the Internet of Vehicles systems.

BIBLIOGRAPHY

[1] A. Boukerche and E. Robson. "Vehicular cloud computing: Architectures, applications, and mobility". In: *Computer Networks* (2018).

[2] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar. "Broadcasting in VANET". In: *2007 mobile networking for vehicular environments*. IEEE. 2007, pp. 7–12.

[3] V. Naumov and T. R. Gross. "Connectivity-aware routing (CAR) in vehicular ad-hoc networks". In: *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. IEEE. 2007, pp. 1919–1927.

[4] W. Viriyasitavat, O. K. Tonguz, and F. Bai. "UV-CAST: an urban vehicular broadcast protocol". In: *IEEE Communications Magazine* 49.11 (2011), pp. 116–124.

[5] O. K. Tonguz, N. Wisitpongphan, and F. Bai. "DV-CAST: A distributed vehicular broadcast protocol for vehicular ad hoc networks". In: *IEEE Wireless Communications* 17.2 (2010), pp. 47–57.

[6] P. Samar, M. R. Pearlman, and Z. J. Haas. "Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks". In: *IEEE/ACM Transactions On Networking* 12.4 (2004), pp. 595–608.

[7] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein. "Geographic routing in city scenarios". In: *ACM SIGMOBILE mobile computing and communications review* 9.1 (2005), pp. 69–72.

[8] C. Shea, B. Hassanabadi, and S. Valaee. "Mobility-based clustering in VANETs using affinity propagation". In: *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*. IEEE. 2009, pp. 1–6.

[9] M. Hadded, R. Zagrouba, A. Laouiti, P. Muhlethaler, and L. A. Saidane. "A multi-objective genetic algorithm-based adaptive weighted clustering protocol in vanet". In: *2015 ieee congress on evolutionary computation (cec)*. IEEE. 2015, pp. 994–1002.

[10] Y. Peng, Z. Abichar, and J. M. Chang. "Roadside-aided routing (RAR) in vehicular networks". In: *2006 IEEE international conference on communications*. Vol. 8. IEEE. 2006, pp. 3602–3607.

[11] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar. "Broadcast storm mitigation techniques in vehicular ad hoc networks". In: *IEEE Wireless Communications* 14.6 (2007), pp. 84–94.

[12] I. Tal and G.-M. Muntean. "User-oriented cluster-based solution for multimedia content delivery over VANETs". In: *IEEE international symposium on broadband multimedia systems and broadcasting*. IEEE. 2012, pp. 1–5.

[13] Y. Shi, L. H. Zou, and S. Z. Chen. "A mobility pattern aware clustering mechanism for mobile vehicular networks". In: *Applied Mechanics and Materials*. Vol. 130. Trans Tech Publ. 2012, pp. 317–320.

[14] C. S. Jensen, D. Lin, and B. C. Ooi. "Continuous clustering of moving objects". In: *IEEE transactions on knowledge and data engineering* 19.9 (2007), pp. 1161–1174.

[15] M. Gerla and M. Gruteser. "Vehicular networks: Applications, protocols, and testbeds". In: *Emerging Wireless Technologies and the Future Mobile Internet* (2011), pp. 201–241.

[16] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe. "Parknet: drive-by sensing of road-side parking statistics". In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM. 2010, pp. 123–136.

[17] W. Viriyasitavat, F. Bai, and O. K. Tonguz. "Toward end-to-end control in vanets". In: *Vehicular Networking Conference (VNC)*. IEEE. 2011, pp. 78–85.

[18] D. Lin, E. Bertino, R. Cheng, and S. Prabhakar. "Location privacy in moving-object environments". In: *Transactions on Data Privacy* 2.1 (2009), pp. 21–46.

[19] M. H. Eiza and Q. Ni. "An evolving graph-based reliable routing scheme for VANETs". In: *IEEE Transactions on Vehicular Technology* 62.4 (2013), pp. 1493–1504.

[20] W. Jiang, F. Li, D. Lin, and E. Bertino. "No one can track you: Randomized authentication in Vehicular Ad-hoc Networks". In: *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*. IEEE. 2017, pp. 197–206.

[21] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen. "ECPP:Efficient Conditional Privacy Preservation Protocol for Secure Vehicular Communications". In: *Proc. of IEEE Conference on Computer Communications*. 2008, pp. 1229–1237.

[22] A. Squicciarini, D. Lin, and A. Mancarella. "Paim: Peer-based automobile identity management in vehicular ad-hoc network". In: *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual*. IEEE. 2011, pp. 263–272.

[23] C. Zhang, R. Lu, X. Lin, P. Ho, and X. Shen. "An efficient identity-based batch verification scheme for vehicular sensor networks". In: *INFOCOM 2008. The 27th Conference on Computer Communications*. IEEE. 2008, pp. 246–250.

[24] M. Raya and J.-P. Hubaux. "Securing vehicular ad hoc networks". In: *Journal of computer security* 15.1 (2007), pp. 39–68.

[25] K.-A. Shim. "CPAS: An Efficient Conditional Privacy-Preserving Authentication Scheme for Vehicular Sensor Networks". In: *IEEE Transaction on Vehicular Technology*. 2012, pp. 1874–1883.

[26] D. Lin, J. Kang, A. Squicciarini, Y. Wu, S. Gurung, and O. Tonguz. "MoZo: A moving zone based routing protocol using pure V2V communication in VANETs". In: *IEEE transactions on mobile computing (TMC)* 16.5 (2016), pp. 1357–1370.

[27] A. Squicciarini, D. Lin, and A. Mancarella. "PAIM: Peer-Based Automobile Identity Management in Vehicular Ad-Hoc Network". In: *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual*. IEEE. 2011, pp. 263–272.

[28] J. Kang, D. Lin, W. Jiang, and E. Bertino. "Highly efficient randomized authentication in VANETs". In: *Pervasive and Mobile Computing* 44 (2018), pp. 31–44.

[29] J. Kang, Y. Elmehdwi, and D. Lin. "Slim: Secure and lightweight identity management in vanets with minimum infrastructure reliance". In: *International Conference on Security and Privacy in Communication Systems*. Springer. 2017, pp. 823–837.

[30] J. Bernsen and D. Manivannan. "Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification". In: *Pervasive and Mobile computing* 5.1 (2009), pp. 1–18.

[31] J. Kakarla, S. S. Sathya, B. G. Laxmi, et al. *A Survey on Routing Protocols and its Issues in VANET*. 2011.

[32] S. Singh and S. Agrawal. "VANET routing protocols: Issues and challenges". In: *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*. IEEE. 2014, pp. 1–5.

[33] T. Song, W. Xia, T. Song, and L. Shen. "A cluster-based directional routing protocol in VANET". In: *2010 ieee 12th international conference on communication technology*. IEEE. 2010, pp. 1172–1175.

[34] P. M. Ruiz, V. Cabrera, J. A. Martinez, and F. J. Ros. "Brave: Beacon-less routing algorithm for vehicular environments". In: *The 7th ieee international conference on mobile ad-hoc and sensor systems (ieee mass 2010)*. IEEE. 2010, pp. 709–714.

[35] P. Fan, J. G. Haran, J. Dillenburg, and P. C. Nelson. "Cluster-based framework in vehicular ad-hoc networks". In: *International Conference on Ad-Hoc Networks and Wireless*. Springer. 2005, pp. 32–42.

[36] M. Aoki and H. Fujii. "Inter-vehicle communication: Technical issues on vehicle control application". In: *IEEE Communications Magazine* 34.10 (1996), pp. 90–93.

[37] O. Kayis and T. Acarman. "Clustering formation for inter-vehicle communication". In: *2007 IEEE Intelligent Transportation Systems Conference*. IEEE. 2007, pp. 636–641.

[38] J. Chen, C. Lai, X. Meng, J. Xu, and H. Hu. "Clustering moving objects in spatial networks". In: *International conference on database systems for advanced applications*. Springer. 2007, pp. 611–623.

[39] Z. Wang, L. Liu, M. Zhou, and N. Ansari. "A position-based clustering technique for ad hoc intervehicle communication". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.2 (2008), pp. 201–208.

[40] W. Liu, J. Feng, Z. Wang, and H. Shan. "Multi-thresholds Clustering Objects in a road network". In: *2008 International Conference on Computer Science and Software Engineering*. Vol. 1. IEEE. 2008, pp. 686–689.

[41] K. Ibrahim and M. C. Weigle. "CASCADE: Cluster-based accurate syntactic compression of aggregated data in VANETs". In: *2008 IEEE Globecom Workshops*. IEEE. 2008, pp. 1–10.

[42] M. A. Alawi, R. A. Saeed, and A. A. Hassan. "Cluster-based multi-hop vehicular communication with multi-metric optimization". In: *2012 International Conference on Computer and Communication Engineering (ICCCE)*. IEEE. 2012, pp. 22–27.

[43] K. Mershad, H. Artail, and M. Gerla. "We can deliver messages to far vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 13.3 (2012), pp. 1099–1115.

[44] T. D. Little, A. Agarwal, et al. "An information propagation scheme for VANETs". In: *Proc. IEEE Intelligent Transportation Systems*. 2005, pp. 155–160.

[45] R. Goonewardene, F. Ali, and E. Stipidis. "Robust mobility adaptive clustering scheme with support for geographic routing for vehicular ad hoc networks". In: *IET Intelligent Transport Systems* 3.2 (2009), pp. 148–158.

[46] Y. Luo, W. Zhang, and Y. Hu. "A new cluster based routing protocol for VANET". In: *2010 second international conference on networks security, wireless communications and trusted computing*. Vol. 1. IEEE. 2010, pp. 176–180.

[47] Y. Ohta, T. Ohta, and Y. Kakuda. "An autonomous clustering-based data transfer scheme using positions and moving direction of vehicles for VANETs". In: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2012, pp. 2900–2904.

[48] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve. "A routing strategy for vehicular ad hoc networks in city environments". In: *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683)*. IEEE. 2003, pp. 156–161.

[49] J. Tian, L. Han, and K. Rothermel. "Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks". In: *Proceedings of the 2003 IEEE international conference on intelligent transportation systems*. Vol. 2. IEEE. 2003, pp. 1546–1551.

[50] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, and K.-K. Lee. "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications". In: *International conference on research in networking*. Springer. 2004, pp. 989–999.

[51] I. Leontiadis and C. Mascolo. "GeOpps: Geographical opportunistic routing for vehicular networks". In: *2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. Ieee. 2007, pp. 1–6.

[52] N. Kumar and M. Dave. "BIIR: A beacon information independent VANET routing algorithm with low broadcast overhead". In: *Wireless Personal Communications* 87.3 (2016), pp. 869–895.

[53] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, and J.-P. Hubaux. "Certificate revocation in vehicular networks". In: *Laboratory for computer Communications and Applications (LCA) School of Computer and Communication Sciences, EPFL, Switzerland (2006)* (2006).

[54] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy. "Efficient and robust pseudonymous authentication in VANET". In: *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*. ACM. 2007, pp. 19–28.

[55] Y. Sun, R. Lu, X. Lin, X. Shen, and J. Su. "An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications". In: *Vehicular Technology, IEEE Transactions on* 59.7 (2010), pp. 3589–3603.

[56] A. Studer, E. Shi, F. Bai, and A. Perrig. "TACKing together efficient authentication, revocation, and privacy in VANETs". In: *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*. SECON'09. Rome, Italy: IEEE Press, 2009, pp. 484–492. ISBN: 978-1-4244-2907-3.

[57] J. Sun, C. Zhang, Y. Zhang, and Y. Fang. "An identity-based security system for user privacy in vehicular ad hoc networks". In: *IEEE Transactions on Parallel and Distributed Systems* 21(9) (2010), pp. 1227–1239.

[58] C. Zhang, P.-H. Ho, and J. Tapolcai. "On batch verification with group testing for vehicular communications". In: *Wireless Networks* 17(8) (2011), pp. 1851–1865.

[59] J. Zhang, Y. Cui, and Z. Chen. "SPA: Self-certified PKC-based Privacy-preserving Authentication Protocol for Vehicular Ad Hoc Networks". In: *International Journal of Security and Its Applications* 6.2 (2012), pp. 409–414.

[60] J. Li, H. Lu, and M. Guizani. "ACPN: a novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs". In: *IEEE Transactions on Parallel and Distributed Systems* 26.4 (2015), pp. 938–948.

[61] K. Sha, Y. Xi, W. Shi, L. Schwiebert, and T. Zhang. "Adaptive privacy-preserving authentication in vehicular networks". In: *Communications and Networking in China, 2006. ChinaCom'06. First International Conference on*. IEEE. 2006, pp. 1–8.

[62] C. D. Jung, C. Sur, Y. Park, and K.-H. Rhee. "A Robust Conditional Privacy-Preserving Authentication Protocol in VANET". In: *Social Informatics and Telecommunications Engineering* 17 (2009), pp. 35–45.

[63] Y. Wang, H. Zhong, Y. Xu, and J. Cui. "ECPB: Efficient Conditional Privacy-Preserving Authentication Scheme Supporting Batch Verification for VANETs". In: *International Journal of Network Security* 18.2 (2016), pp. 374–382.

[64] Y. Hao, C. Yu, C. Zhou, and W. Song. "A distributed key management framework with cooperative message authentication in VANETs". In: *Selected Areas in Communications, IEEE Journal on* 29.3 (2011), pp. 616–629.

[65] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu. "2FLIP: A Two-Factor Lightweight Privacy-Preserving Authentication Scheme for VANET". In: *IEEE Transactions on Vehicular Technology* 65.2 (2016), pp. 896–911.

[66] L. Yeh, Y. Chen, and J. Huang. "PAACP: A portable privacy-preserving authentication and access control protocol in vehicular ad hoc networks". In: *Computer Communications* 34.3 (2011), pp. 447–456.

[67] Z. Tan. "A lightweight conditional privacy-preserving authentication and access control scheme for pervasive computing environments". In: *Journal of Network and Computer Applications* 35.6 (2012), pp. 1839–1846.

[68] C. Gamage, B. Gras, B. Crispo, and A. Tanenbaum. "An identity-based ring signature scheme with enhanced privacy". In: *Securecomm and Workshops*. 2006, pp. 1–5.

[69] S. Zeng, Y. Huang, and X. Liu. "Privacy-preserving communication for vanets with conditionally anonymous ring signature". In: *International Journal of Network Security* 17.2 (2015), pp. 135–141.

[70] U. Rajput, F. Abbas, J. Wang, H. Eun, and H. Oh. "CACPPA: A Cloud-Assisted Conditional Privacy Preserving Authentication Protocol for VANET". In: *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*. 2016, pp. 434–442.

[71] J. Camenisch and A. Lysyanskaya. "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation". In: *EUROCRYPT*. 2001, pp. 93–118.

[72] J. Camenisch and E. Van Herreweghen. "Design and implementation of the idemix anonymous credential system". In: *Proceedings of the 9th ACM conference on Computer and communications security*. ACM. 2002, pp. 21–30.

[73] J. Camenisch and T. Groß. "Efficient attributes for anonymous credentials". In: *Proceedings of the 15th ACM conference on Computer and communications security*. ACM. 2008, pp. 345–356.

[74] E. Brickell, J. Camenisch, and L. Chen. "Direct anonymous attestation". In: *Proceedings of the 11th ACM conference on Computer and communications security*. ACM. 2004, pp. 132–145.

[75] E. Cesena, H. Löhr, G. Ramunno, A.-R. Sadeghi, and D. Vernizzi. "Anonymous authentication with TLS and DAA". In: *Trust and Trustworthy Computing*. 2010, pp. 47–62.

[76] C. Wachsmann, L. Chen, K. Dietrich, H. Löhr, A.-R. Sadeghi, and J. Winter. "Lightweight anonymous authentication with TLS and DAA for embedded mobile devices". In: *Information Security*. 2011, pp. 84–98.

[77] J. Camenisch and A. Lysyanskaya. "Dynamic accumulators and application to efficient revocation of anonymous credentials". In: *Advances in Cryptology —-CRYPTO 2002*. Springer, 2002, pp. 61–76.

[78] G. Di Crescenzo, J. Katz, R. Ostrovsky, and A. Smith. "Efficient and non-interactive non-malleable commitment". In: *Advances in Cryptology——Eurocrypt* (2001), pp. 40–59.

[79] M. Fischlin and R. Fischlin. "Efficient non-malleable commitment schemes". In: *Advances in Cryptology, Crypto 2000. Lecture Notes in Computer Science, vol. 1880 (Springer, Berlin, 2000)*. Crypto 2000. 2000, pp. 414–432.

[80] M. Fischlin and R. Fischlin. "Efficient non-malleable commitment schemes". In: *Journal of cryptology* 24.1 (2011), pp. 203–244.

[81] T. Pedersen. "Non-interactive and information-theoretic secure verifiable secret sharing". In: *Advances in Cryptology—CRYPTO'91*. Springer. 1992, pp. 129–140.

[82] T. P. Pedersen. *Non-interactive and information-theoretic secure verifiable secret sharing*. 1998.

[83] *Standard specification for telecommunications and information exchange between roadside and vehicle systems*. https://www.astm.org/Standards/E2213.htm. 2018.

[84] J. Lee. "Design of a network coverage analyzer for roadside-to-vehicle telematics networks". In: *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*. IEEE. 2008, pp. 201–205.

[85] S. Goldwasser, S. Micali, and C. Rackoff. "The Knowledge Complexity of Interactive Proof Systems". In: *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*. Providence, Rhode Island, U.S.A., 1985, pp. 291–304.

[86] P. Paillier. "Public Key Cryptosystems Based on Composite Degree Residuosity Classes". In: *Advances in Cryptology - Eurocrypt '99 Proceedings, LNCS 1592*. Prague, Czech Republic: Springer-Verlag, 1999, pp. 223–238.

[87] Y. Zhang, M. M. A. Kabir, Y. Xiao, D. D. Yao, and N. Meng. "Automatic Detection of Java Cryptographic API Misuses: Are We There Yet". In: *IEEE Transactions on Software Engineering* (2022).

[88] D. Wang, X. Zhang, Z. Zhang, and P. Wang. "Understanding security failures of multi-factor authentication schemes for multi-server environments". In: *Computers & Security* 88 (2020), p. 101619.

[89] X. Lin, X. Sun, P.-H. Ho, and X. Shen. "GSIS: A Secure and Privacy-Preserving Protocol for Vehicular Communications". In: *Vehicular Technology, IEEE Transactions on* 56.6 (Nov. 2007), pp. 3442–3456. ISSN: 0018-9545.

[90] S. Mohanty, D. Jena, and S. K. Panigrahy. "A secure RSU-aided aggregation and batch-verification scheme for vehicular networks". In: *Intemational Conference on Soft Computing and its Applications (ICSCA2012), pp174-178*. 2012.

VITA

Jian Kang is currently a PhD candidate at Electrical Engineering & Computer Science Department, University of Missouri. He received the B.S. degree in Computer Science from Fuzhou University in 2013, and the Master's degree in Computer Software and Theory in Fuzhou University in 2016. His general research interests are information management, security and privacy in cyber-physical systems.