Secure and Privacy-Preserving Fog-Assisted Vehicular Crowdsensing

by

Sultan Basudan

A thesis
presented to the University of Ontario Institute of Technology
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

University of Ontario Institute of Technology
Oshawa, Ontario, Canada
August, 2018

© Sultan Basudan, 2018
THESIS EXAMINATION INFORMATION

Submitted by: Sultan Basudan

Ph.D. in Computer Science

Thesis title: Secure and Privacy-Preserving Fog-Assisted Vehicular Crowdsensing

An oral defense of this thesis took place on August 31, 2018 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee  Dr. Shahram Heydari
Research Supervisor  Dr. Xiaodong Lin
Research Co-supervisor  Dr. Karthik Sankaranarayanan
Examine Committee Member  Dr. Ying Zhu
Examine Committee Member  Dr. Patrick Hung
University Examiner  Dr. Min Dong
External Examiner  Dr. Lian Zhao, Ryerson University

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.
I hereby declare that I am the sole author of this thesis. Any published (or unpublished) ideas and/or techniques from the work of others are fully acknowledged in accordance with the standard referencing practices. I understand that my thesis may be made electronically available to the public.

Sultan Basudan, 2018
Abstract

Vehicular crowdsensing (VCS) is an emerging paradigm where vehicles use onboard sensors to collect and share data with the aim of measuring phenomena of common interest. Great attention has been recently directed towards road surface condition monitoring systems (RSCMS). Such an endeavor is of critical importance in transportation infrastructure management. As a response, multiple recommendations have been proposed. These recommendations make use of mobile sensing, more specifically contemporary applications and architectures that are used in both crowdsensing and vehicle-based sensing. This has allowed for automated control as well as analysis of road surface quality. These innovations have thus encouraged and shown the importance of the cloud in providing reliable transport services to clients. Nonetheless, these initiatives have not been without challenges, ranging from mobility support, location awareness, low latency and geo-distribution. In order to address these challenges, a new term, known as fog computing, has been coined as a novel paradigm. Therefore, this present work exploits the advantages of VCS and fog computing paradigms in order to propose a promising framework, which is referred to as fog-assisted vehicular crowdsensing (FVCS). Although FVCS has addressed the aforementioned challenges, it may encounter various security threats and privacy concerns that could jeopardize public safety and become the main barrier to the acceptance of such a new technology.

This thesis presents the proposal of a secure and privacy-preserving framework for FVCS. The objective of the proposed framework is to allow vehicles to share their resources while preserving their privacy by preventing private information from being disclosed. A thorough search of the relevant literature suggests that the proposed framework is the first work that attempts to address critical security and privacy challenges in FVCS. Attention is first focused on investigating the threat towards the data generated by vehicles, which is then forwarded to cloud servers and organizations by roadside units (RSUs). The generated data can be exploited by an adversary to reveal vehicle privacy. Protecting the privacy of participants is essential to the success of FVCS applications. Therefore, this work presents a privacy-preserving protocol for enhancing security in a VCS-based road surface condition monitoring system using fog computing. This protocol presents a highly efficient certificateless aggregate signcryption scheme (CLASC). On the basis of the proposed CLASC scheme, a data transmission protocol for monitoring road surface conditions is designed with security aspects such as information confidentiality, mutual authenticity,
integrity, privacy and anonymity. In analyzing the system, the ability of the proposed protocol to both achieve the set objectives and exercise higher efficiency with respect to computational and communication abilities, in comparison to existing systems, is also considered.

Furthermore, in order to revoke compromised users from the system, this work offers a novel secure and efficient revocable privacy-preserving protocol in FVCS. The proposed protocol is distinguished by using a binary tree structure to address scalability concerns and achieve an efficient revocation function. Based on the CLASC scheme, this protocol is designed with security properties that include report confidentiality, integrity, privacy, revocation functionality and key escrow resilience. Extensive simulations are conducted in order to validate the proposed protocol. It is demonstrated that the proposed protocol achieves a much better performance than its counterparts in terms of scalability, user revocation and signature verification.

In addition to the above countermeasures in FVCS, this work also presents an efficient deduplicated reporting scheme in order to ensure that vehicles are free from security risks and privacy threats while sharing their resources with semi-trusted nodes. The proposed scheme is characterized by employing homomorphic property to provide secure computations on ciphertext. In addition, the proposed scheme provides a promising approach for improving storage and communication overheads while maintaining contents’ privacy. Specifically, RSUs as fog nodes are able to detect and remove replicate crowdsensing reports without learning information about their contents. Furthermore, the proposed scheme achieves fairness between vehicles whose reports are reduplicated and deleted. Based on the CLASC scheme, this strategy is designed with security properties such as report confidentiality, integrity, mutual authenticity, privacy, anonymity, secure data deduplication and key escrow resilience. To conclude, this work demonstrates both the achievement of the proposed scheme’s secure data deduplication property and its efficiency due to low computational and communication overheads.

Moreover, attention is also given to how the present work can be further developed by exploring the important business perspective related to FVCS technology. In order to introduce a competitive product that outperforms its competitors, this study investigates business-related aspects such as technology management and its importance, strategic analysis, technology recommendations, technology forecasting, cost-efficient FVCS deployment and stakeholders.
**Acknowledgements**

I would like to thank all the people who made this possible. This thesis would not have been possible without the help and support of my supervisors, my thesis committee members, and my colleagues in Information Forensics and Security Laboratory (IFS Lab). During my Ph.D. research I have learned many new things, and without the people surrounding me I could not enjoy from this period of my life.

First of all, I gratefully acknowledge my supervisor, Professor Xiaodong Lin. He made available his support and aid in a number of ways. He always does care about his students, and I had this opportunity to discuss the obstacles encountered me in my study and research openly with him. He not only helps me to develop the academic skills, but also guides me to strive for excellence.

Also, I would like to thank my co-supervisor Professor Karthik Sankaranarayanan for his constant support, availability and constructive suggestions, which were determinant for the accomplishment of the work presented in this thesis.

In addition, I would like to thank all the members of my examining committee, Dr. Ying Zhu, Dr. Patrick Hung, Dr. Min Dong and Dr. Shahram Heydari, for the time and efforts to read my thesis. In spite of their busy schedules, all have been readily available for advice, reading and encouragement. I would also like to thank Prof. Lian Zhao as my thesis external examiner and sharing his invaluable insight on computer and communication security with me.

I would like to extend my appreciation to all the IFS Lab group for continuing support and warm working atmosphere. Special thanks go to Dr. Khalid Alharbi, Dr. Aiqing Zhang and Dr. Abdulrahman Alamer, for day-and-night discussion and continuous support throughout my Ph.D. study. Furthermore, grateful acknowledgments are made for Jazan University and the Saudi Arabian Cultural Bureau in Ottawa for providing me with a full scholarship during my Ph.D. study.

I would never get this far without the support of my mother, sisters and brothers. I think them for always believing in me and supporting me. Their love and encouragement have been and will always be a great source of inspiration in my life. Finally, my special thanks go to my wife for the loving support and patience she has for me to fulfill my career goals.
# Table of Contents

**List of Tables** xi

**List of Figures** xii

**List of Acronyms** xiii

1. **Introduction** 1
   1.1 Motivation ............................................. 1
   1.2 Objectives and Contributions .............................. 6
   1.3 Organization of the Thesis ..................................... 8

2. **Background and Literature Review** 9
   2.1 An Overview of Vehicular Crowdsensing ......................... 9
   2.2 Network Infrastructure ........................................ 10
   2.3 An Overview of Fog Computing .................................. 11
     2.3.1 Networking Architecture ................................. 11
     2.3.2 Fog Computing Features .................................. 12
   2.4 FVCS Architecture ........................................... 13
   2.5 FVCS Applications ........................................... 16
2.6 FVCS Security and Privacy Challenges ........................................ 17
2.7 Applied Cryptography to FVCS .................................................... 18
  2.7.1 Certificatless Public Key Cryptography (CL-PKC) ....................... 19
  2.7.2 Symmetric and Asymmetric Cryptosystems ............................... 20
  2.7.3 Signcryption Technique .......................................................... 20
  2.7.4 Homomorphic Encryption Technique ....................................... 21
2.8 Related Work ........................................................................... 21
  2.8.1 Road Surface Condition Monitoring System ............................... 22
  2.8.2 Certificateless Aggregate Signcryption Schemes ......................... 23
  2.8.3 Compromised User Revocation ................................................. 24
  2.8.4 Secure Deduplication .............................................................. 26

3 A Privacy-Preserving Protocol for Fog-Assisted Vehicular Crowdsensing 28
  3.1 Introduction ............................................................................ 28
  3.2 System Models and Design Goals .............................................. 30
    3.2.1 System Model ................................................................. 30
    3.2.2 Threat Model ................................................................. 32
    3.2.3 Design Goals ................................................................. 33
  3.3 Preliminaries .......................................................................... 34
    3.3.1 Bilinear Maps ................................................................. 34
    3.3.2 Complexity Assumptions .................................................. 35
    3.3.3 Framework of Certificateless Aggregate Signcryption ............... 35
    3.3.4 Security Model of CLASC ............................................... 36
  3.4 Proposed CLASC scheme .......................................................... 37
3.5 Proposed Privacy-Preserving Protocol ................................................. 40
   3.5.1 System Initialization ................................................................. 41
   3.5.2 Data Formulation and Sending .................................................. 42
   3.5.3 SRER Aggregated Verification .................................................. 43
   3.5.4 Data Receiving ................................................................. 44
3.6 Security Analysis ................................................................. 44
3.7 Performance Evaluation ................................................................. 45
   3.7.1 Computational Cost ................................................................. 46
   3.7.2 Communication Overhead .......................................................... 48
3.8 Summary ................................................................. 49
3.9 Limitation ................................................................. 49

4 Efficient Compromised Node Revocation in Fog-Assisted Vehicular Crowdsensing 51
   4.1 Introduction ................................................................. 51
   4.2 System Models and Design Goals .................................................. 53
   4.3 Construction of the proposed protocol ........................................... 56
      4.3.1 Preliminaries ................................................................. 56
      4.3.2 KUNode Algorithm Definition ............................................... 57
      4.3.3 Proposed revocable privacy-preserving scheme ......................... 58
   4.4 Security Analysis ................................................................. 65
   4.5 Performance Evaluation ................................................................. 66
      4.5.1 Computational Cost ................................................................. 66
      4.5.2 Communication Overhead .......................................................... 68
      4.5.3 KGC Overhead ................................................................. 68
   4.6 Summary ................................................................. 69
   4.7 Limitation ................................................................. 69
## Efficient Deduplicated Reporting in Fog-Assisted Vehicular Crowdsensing

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>70</td>
</tr>
<tr>
<td>5.2 System Models and Design Goals</td>
<td>73</td>
</tr>
<tr>
<td>5.3 Preliminaries</td>
<td>75</td>
</tr>
<tr>
<td>5.4 Proposed Privacy-Preserving Data Deduplication Scheme</td>
<td>76</td>
</tr>
<tr>
<td>5.5 Security Analysis</td>
<td>81</td>
</tr>
<tr>
<td>5.6 Performance Evaluation</td>
<td>82</td>
</tr>
<tr>
<td>5.7 Summary</td>
<td>86</td>
</tr>
<tr>
<td>5.8 Limitation</td>
<td>86</td>
</tr>
</tbody>
</table>

## Conclusions and Future Work

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Contributions</td>
<td>87</td>
</tr>
<tr>
<td>6.2 Future Work</td>
<td>88</td>
</tr>
<tr>
<td>6.2.1 Differential Privacy in Fog-Assisted Vehicular Crowdsensing</td>
<td>89</td>
</tr>
<tr>
<td>6.2.2 A Fog-Assisted Vehicular Crowdsensing Framework: A Technology Management Perspective</td>
<td>89</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Cryptographic operations comparison with other CLASC schemes .......................... 47
3.2 Cryptographic operations running time ................................................................. 47
3.3 Computational cost and communication overhead analysis ...................................... 49

4.1 Data sending, receiving and batch verification comparison of different privacy-
preserving protocols ................................................................. 67
4.2 Computational and communication overhead analysis ......................................... 68

5.1 Computational cost of the proposed scheme ......................................................... 83
# List of Figures

1.1 Cloud-based architecture ...................................................... 3
1.2 An example of detected results .............................................. 4
1.3 Fog-based architecture ........................................................ 5

2.1 Architecture of fog-assisted vehicular crowdsensing ......................... 14

3.1 System model ........................................................................... 31
3.2 Threat model ........................................................................... 32
3.3 Efficiency comparison with other CLASC schemes ............................... 48

4.1 System model ........................................................................... 54
4.2 An example of compromised node revocation in KUNodes algorithm ....... 57
4.3 Time consumption comparison .................................................... 67

5.1 Crowdsensing reports formulation .............................................. 72
5.2 System model ........................................................................... 74
5.3 Discarding redundant copies ...................................................... 79
5.4 Bandwidth overhead ................................................................ 84
5.5 Communication overhead comparison .......................................... 85

6.1 Stakeholders ............................................................................. 98
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCS</td>
<td>Vehicular Crowdsensing</td>
</tr>
<tr>
<td>RSCMS</td>
<td>Road Surface Condition Monitoring System</td>
</tr>
<tr>
<td>FVCS</td>
<td>Fog-Assisted Vehicular Crowdsensing</td>
</tr>
<tr>
<td>CL-PKC</td>
<td>Certificateless Public Key Cryptography</td>
</tr>
<tr>
<td>RSUs</td>
<td>Roadsode Units</td>
</tr>
<tr>
<td>CLSC</td>
<td>Certificateless Signcryption</td>
</tr>
<tr>
<td>CLASC</td>
<td>Certificateless Aggregate Signcryption</td>
</tr>
<tr>
<td>MCS</td>
<td>Mobile Crowdsensing</td>
</tr>
<tr>
<td>OBUs</td>
<td>On-Board Units</td>
</tr>
<tr>
<td>VANETs</td>
<td>Vehicular Ad-Hoc Networks</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>ITS</td>
<td>Intelligent Transportation System</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>CA</td>
<td>Certification Authority</td>
</tr>
<tr>
<td>ID-PKC</td>
<td>Identity-Based Public Key Cryptography</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>PKG</td>
<td>Private Key Generator</td>
</tr>
<tr>
<td>CSR</td>
<td>Crowdsensing Report</td>
</tr>
<tr>
<td>P2</td>
<td>Pothole Patrol</td>
</tr>
<tr>
<td>CRL</td>
<td>Certificate Revocation List</td>
</tr>
<tr>
<td>CRT</td>
<td>Certificate Revocation Tree</td>
</tr>
<tr>
<td>OCSP</td>
<td>Online Certificate Status Protocol</td>
</tr>
<tr>
<td>RC2RL</td>
<td>Revocation Compressed Certificate Revocation List</td>
</tr>
<tr>
<td>RTPD</td>
<td>Revocation of the Tamper Proof Device</td>
</tr>
<tr>
<td>DRP</td>
<td>Distributed Revocation Protocol</td>
</tr>
<tr>
<td>KGC</td>
<td>Key Generator Center</td>
</tr>
<tr>
<td>SEM</td>
<td>Security Mediator</td>
</tr>
<tr>
<td>MLE</td>
<td>Message-Locked Encryption</td>
</tr>
<tr>
<td>CE</td>
<td>Convergent Encryption</td>
</tr>
<tr>
<td>CC</td>
<td>Control Center</td>
</tr>
<tr>
<td>CDH</td>
<td>Computational Diffie-Hellman</td>
</tr>
<tr>
<td>DBDH</td>
<td>Decisional Bilinear Diffie-Hellman</td>
</tr>
<tr>
<td>SRER</td>
<td>Secure Road Event Report</td>
</tr>
<tr>
<td>NRU</td>
<td>Non-Revoked Users</td>
</tr>
<tr>
<td>TF</td>
<td>Technological Forecasting</td>
</tr>
<tr>
<td>MA</td>
<td>Morphological Analysis</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Mobile crowdsensing (MCS) is an emerging paradigm that integrates sensors and embedded computing devices to allow individuals to cooperatively collect and share data and extract information to measure and map phenomena of common interest with the use of sensing and communication technologies [35], [53]. MCS has now become a foundation for a number of sensing applications. As an example, smart phones are able to sense the environment with several embedded sensors, including a camera, global positioning system (GPS), and an accelerometer, to generate and share sensing reports with interested organizations. With the advance of vehicular technology, vehicular crowdsensing (VCS) extends the concept of MCS, where modern vehicles are equipped with a massive number of sophisticated on-board sensors and powerful computational on-board units (OBUs) [36], which provide the fundamental capability and feasibility of the VCS paradigm. It is worth indicating that MCS can support drivers by using their mobile devices to collect and contribute data. However, mobile devices suffer from sensor limitations, which are insufficient for providing accurate and valuable data for different applications such as parking navigation and road surface monitoring.

VCS has been successfully adopted to enable sensing applications such as road surface condition monitoring systems (RSCMS) [64], [29], [85]. The condition of road surfaces is considered
as a major indicator of the quality of roads. In fact, the classification of whether a road is safe or
dangerous is most commonly determined by its surface condition. Parameters such as potholes,
bumps and slipperiness are conventionally considered as the distinguishing features of the qual-
ity of road surfaces [64]. Also notable is the fact that the surface condition of roads is one of the
major causes for vehicle damage and premature aging. In Ontario (Canada), the winter climate
is known to generate weather conditions such as snow, sleet, ice and freezing rain. When acting
alongside poor roadside surface conditions, this creates situations that are potentially dangerous
to motorists, vehicles, pedestrians and properties [89]. As a result, this is an area where systems
capable of monitoring road conditions are critical to the improvement of safety in roads, the low-
ering of accident rates and protection of vehicles from damage as a result of poor surface road
conditions.

Municipalities worldwide spend millions of dollars on the maintenance and repair of road
surfaces [29]. Traditionally, municipalities engage patrol crews that perform physical exami-
nation of road surface conditions with the aim of identifying problems such as potential skid
spots and potholes. Nonetheless, with the use of advanced vehicular technologies, especially
vehicular communication combined with sensing technologies, road anomalies can be easily
identified and managed. This is achieved using an advanced system for monitoring road surface
conditions [85]. Advances in sensing technologies have already allowed the use of sensors in
gathering useful information from the environment [64], [29], [85].

The technological strides made in VCS, such as the advent of smart vehicles, have aided the
collection of information regarding the environment. For example, a vehicle embedded with on-
board sensors gathers data to measure traffic congestion on a specific road. The emphasis that
is placed on contemporary applications/architectures for both crowdsensing and vehicle-based
sensing, alongside advances in cloud computing, actually allows for data collection, analysis,
storage, processing and transmission, in an efficient manner.

Cloud-based architecture, as shown in Figure 1.1, is used by various applications such as the
smart city [59], which consists of sensors that could be embedded in a vehicle to sense and gener-
ate data that is forwarded to cloud servers via roadside units (RSUs). These on-board sensors are
used to collect data when the vehicle encounters anomalies when, for example, hitting a pothole
on the road, as displayed in Figure 1.2. The data is then transferred to a centralized cloud system
from where it is processed. The cloud-based facility acts as an efficient means through which
the integrated system remains up to date while maintaining privacy and security. RSUs as base stations help in relaying data to the cloud for processing and to provide recommendations [84]. For any application, the approaching cars require real-time data processing in order to be able to offer instant recommendations with regard to road surface conditions. Nonetheless, solutions that are cloud-based and used in dealing with VCS applications present a number of issues, such as the transmission of extensive real-time data to centralized cloud servers, which are prone to time delays and elevated cost of bandwidth.

Furthermore, cloud servers are able to release spatial tasks. Collecting data from specific locations is based on spatial task information. Cloud servers also have the ability to recruit vehicles to sense and generate crowdsensing reports based on the released spatial tasks [67]. In addition, cloud servers are responsible for collecting various aspects of information about vehicles (e.g., locations) based on the spatial tasks, which have the unique requirements of sensing tasks and user mobility [88]. As an example, in order to measure traffic congestion in a downtown area, the cloud server should recruit the vehicles being driven on the roads in that area. However, since the cloud server has no knowledge of network connections with vehicles, it is difficult to guarantee whether the potential vehicles will receive the assigned sensing tasks and upload the sensing reports on time. Specifically, in order to perform sensing tasks, some vehicles have to
travel to particular locations with a certain impact on time and travel. Therefore, an efficient solution to help improve the cloud server in terms of accuracy of task allocation and recruitment of appropriate vehicles should be recommended.

A computer paradigm, referred to as fog or edge computing, has recently emerged. This computing model, which stretches cloud computing and related services to the network edge, as presented in Figure 1.3, offers low latency, position awareness, a large node, extensive geodistribution, increased mobility and real-time application processes [19]. In contrast, with globally centralized cloud-based systems, once the vehicles sense and generate data, the data is then transmitted to the closest RSU, i.e., a fog device [84]. The RSU then does real-time computations in addition to taking local decisions, as shown in Figure 1.2. The results together with recommendations can also be transmitted to other approaching vehicles heading towards the affected region. This system thus achieves low latency as well as a reduction in bandwidth cost. Thus, with the use of fog computing, there can be envisioned a system for measuring road surface conditions. This system will allow applications to operate as reasonably as possible with the vast amount of sensed information collected via sensors.

In addition, deploying RSUs as fog nodes helps with reducing the burden on the cloud server as well as improving the accuracy of task allocation. In particular, rather than the cloud server
recruiting vehicles to perform tasks, fog node RSUs can carry out the recruiting process and determine the relevant vehicles to perform the task based on requirements. The cloud server firstly assigns the spatial tasks to the connected fog nodes (e.g., RSU). The RSU, which is physically located in the intended sensing area, can then recruit the appropriate vehicles to perform the tasks. Therefore, the fog computing paradigm helps to minimize the overhead of the cloud server as well as enhance the accuracy of task allocation. Moreover, RSUs act as geography-related local servers that have complete knowledge about the vehicles in their coverage area. Motivated by the promising features provided by VCS and fog computing, this present work proposes a fog-assisted vehicular crowdsensing (FVCS) framework as a foundation for RSCMS. The proposed FVCS framework is described in more detail in Chapter 2. The main objective of this work is to enhance security and privacy in data transmission when vehicles outsource the task of crowdsensing reports.

![Fog-based architecture](image)

Figure 1.3: Fog-based architecture

Security and privacy issues need to be addressed before implementation in FVCS. Although the majority of previous research studies focused on the transmission of data in vehicular ad hoc networks (VANETs) [69], [50], [45], [97], [101], the security challenges related to methods of ensuring authenticity and confidentiality regarding a reported road event are yet to be explored. In particular, there still exist several challenges that may affect the development of RSCMS.
The first challenge is not just the message confidentiality that needs to be addressed but also the authenticity and integrity of transmitted messages. Furthermore, it is important to protect user-related data, including user ID and position. Another challenge is whether the system can revoke the compromised user where detected as being malicious. Moreover, the participants will have definite concerns about their private information if there is a semi-trusted node involved in the system (e.g., RSU).

In order to address these challenges, the main motivation of this study is to propose a novel framework for privacy-preserving in order to enhance security and privacy in FVCS. The objective is to defend against various security and privacy issues in FVCS. The proposed protocols and schemes utilize a combination of cryptographic protocols and techniques as their foundation; i.e. they use certificateless public key cryptography (CL-PKC), bilinear pairing, signcryption and homomorphic encryption. Thus, they aid in protecting user privacy and satisfying the security requirements in FVCS.

1.2 Objectives and Contributions

The main objective of this research is to design a privacy-preserving framework for FVCS. This includes a privacy-preserving protocol in data transmission for enhancing security, aiming to fulfill confidentiality, integrity and authentication for crowdsensing reports, and ensure that participants are free from any risk of privacy disclosure. Moreover, an efficient revocable privacy-preserving protocol is proposed in order to achieve the security requirement, which is compromised user revocation (e.g. compromised vehicle or RSU). This work concludes by proposing an efficient deduplicated reporting scheme in FVCS to enable a semi-trusted RSU to find replicated crowdsensing reports by performing homomorphic calculations on the ciphertext without disclosing vehicle privacy. Specifically, the main contributions of this study are as follows:

- In order to form a new and promising FVCS framework, the advantages of two different paradigms, namely VCS and fog computing, are exploited. Specifically, all the reports generated by vehicles are processed near the end user rather than being processed in the centralized cloud. Therefore, the proposed framework can take advantage of fog computing to act as a foundation for real-time VCS applications. In considering that vehicle
privacy may be disclosed during a report formulation and generation, this present work proposes a novel privacy-preserving protocol for enhancing security in the FVCS [11]. At the onset, a highly efficient certificateless aggregate signcryption (CLASC) scheme is presented. On the basis of the CLASC scheme, a privacy-preserving protocol for monitoring road surface conditions is designed. The proposed protocol combines CL-PKC and signcryption technique in order to protect vehicle privacy from being disclosed during the generating of reports. The proposed protocol is much more efficient in terms of computational cost and communication overhead compared to existing schemes [31], [51].

- An efficient revocable privacy-preserving protocol is also proposed. This protocol will enable the system (e.g., the key generation centre) to have the ability to revoke any detected malicious users from the system [10]. The proposed protocol implements the revocation technique, which takes place when a legitimate user is discovered as being malicious. This may happen whether the user’s private key is expired or not. The proposed protocol makes use of a combination of a binary tree structure with a certificateless signcryption technique to ensure compromised users are revoked, thus preserving the entire system from being breached. Specifically, the protocol protects crowdsensing reports that are processed by RSUs from being accessed by a non-legitimate user (e.g., compromised RSU). Extensive simulations demonstrate efficiency with regard to the computational cost and ciphertext size of the proposed protocol. In terms of scalability, compromised user revocation, signature verification process and key escrow problem evasion, the proposed protocol outperforms existing competing schemes [98], [68], [92].

- Inspired by the fact that there are inevitably some duplicates in the crowdsensing reports generated by vehicles at the same location, and that gateways as RSUs may be corrupted, an efficient deduplicated reporting scheme in FVCS [9] is designed in order to address these challenges. The proposed scheme integrates a homomorphic concept with a signcryption technique to allow the semi-trusted gateways to process and analyze encrypted crowdsensing reports. The proposed scheme also supports the deduplication process on the reports without revealing any sensitive information related to the participating vehicles. Furthermore, the proposed scheme is much more efficient and guarantees fairness between vehicles whose reports are reduplicated and deleted. A detailed performance anal-
ysis demonstrates the achievement of secure data deduplication property and efficiency in terms of computational cost, communication overhead and bandwidth overhead.

1.3 Organization of the Thesis

This thesis is organized as follows. An overview of vehicular crowdsensing (VCS) and fog computing, associated with FVCS security and privacy issues, applied cryptography to FVCS and related work are introduced in Chapter 2. In Chapter 3, a privacy-preserving protocol for enhancing security in the FVCS is presented, followed by security analysis and performance evaluation. Chapter 4 introduces a revocable privacy-preserving protocol against compromised users in FVCS and provides a security analysis and performance evaluation. In Chapter 5, an efficient deduplicated reporting scheme in FVCS is described, followed by security analysis and performance evaluation. Finally, conclusions and future work are described in Chapter 6.
Chapter 2

Background and Literature Review

2.1 An Overview of Vehicular Crowdsensing

Recently, a new business model has been made aware of the scientific community regarding the sensing phenomenon, referred to as crowdsensing paradigm (also referred to as mobile crowdsensing MCS) [71]. It allows a massive number of mobile sensors to be used for exchanging information. Moreover, the applications of MCS may further benefit and enable our society. In other words, it indicates to the sharing of sensor data to measure a community phenomenon. For instance, a large group of individuals may have mobile devices capable of sensing and computing such as smartphones. These devices are able to measure, map, analyze or predict any processes of common interest by sharing data and extracting information. In terms of MCS applications, they are particularly attractive to organizations because they can provide them valuable data without the need to make significant investments. However, mobile devices suffer from sensor limitations, which are insufficient for providing valuable data for different applications such as parking navigation and road surface monitoring. Therefore, MCS is a very charming solution for organizations to collect important data in terms of an intelligent transportation system (ITS). In this study, we consider RSCMS as the application scenario we have implemented to investigate from different perspectives including efficiency, accuracy, security and privacy.

VCS paradigm has attracted more and more attention in recent years, which can help com-
panies, data analysts or communities to collect and share large amounts of data with the aim of measuring phenomena of common interest. Modern vehicles are also equipped with on-board sensors and wireless communication devices, such as cameras, GPS, acceleration sensors, and OBUs [54], [44]. The essential capability of VCS is provided by the equipped sensors and devices. Vehicles can periodically report the driving information (e.g., location, real-time speed, and driving video) and also provide traffic conditions, road conditions, and weather conditions for transportation planning, road system design, traffic signal control, and so on [64], [29], [85]. Recently, the design of vehicular communication and sensing has been growing. A majority of some applications, such as traffic monitoring, transportation management and data collection, requires the vehicle to act as a sensor. It mainly relies on the sensing capabilities of vehicles and the communication channels. For example, a camera could be embedded to a vehicle to do traffic monitoring, and then transmit the captured data to a control center for further processing. Obviously, vehicles equipped with various sensors have become ubiquitous. These sensors could be utilized to measure; noise via the microphone, movement via the accelerometer and location via GPS [44]. Therefore, these sensors can collect vast quantities of data that may be useful in a variety of ways. For example, locating potholes on the roads in cities by processing GPS and accelerometer data. In order to adopt and implement VCS applications, a promising computing network infrastructure should be provided.

2.2 Network Infrastructure

An efficient network infrastructure is significantly needed to implement large-scale applications, i.e., RSCMS. In this system, smart vehicles are equipped with various sensors such as accelerometer and GPS. Thus, these vehicles deal with massive amounts of generated data. In the application scenario, the approaching vehicles require real-time data processing in order to be able to offer instant recommendations with regard to the road surface conditions. A network infrastructure that can efficiently handle such large volume of real-time data is essential. In parallel to these developments, cloud computing as a network infrastructure has experienced significant improvements in recent years in terms of both coverage and performance [79]. It allows transmitting, storing, and processing large amounts of data in an efficient manner. However, adopting cloud-based solutions to use in VCS applications presents a number of issues. Latency and a
high cost of bandwidth are the potential challenges to centralized cloud servers during receiving extensive real-time data. Furthermore, the accuracy of assigning spatial tasks by a cloud server has a burden of its direct knowledge of network connections in terms of recruiting proper vehicles located at specific location regarding spatial task information. Therefore, fog computing paradigm has been adopted in order to address the aforementioned challenges.

2.3 An Overview of Fog Computing

The explosive growth of the Internet of Things (IoT) brought millions of devices and sensors connected to the Internet [32]. These devices and sensors are generating more data every day. Moving all that data to a central data center, i.e., cloud for analysis presents latency, bandwidth, security, and reliability challenges. Actually, there are several emerging IoT applications such as industrial automation, transportation, and networks of sensors. These applications demand real-time processing or risk a longer time delay. Hence, fog computing has been introduced to support these IoT applications on billions of connected devices to run directly at the network edge [23]. It is a generic platform for edge computing and focuses on the localized service applications and computational requests. Many IoT applications require both fog localization and cloud globalization for analytics and big data. This new distributed computing allows applications to run as close as possible to large quantities of sensed data. Thus, fog computing can manage big data more efficiently. Also, fog data services run directly on the network edge. The first goal of these services is to convert raw data generated from sensors into smooth information, i.e., sensitive or insensitive data. While insensitive data can travel to the cloud for long term storage and further historical analysis, sensitive data is stored and analyzed at the network edge. The second goal of these services is to filter data based on metadata, aggregation and detection of events, and efficient encryption of plaintext sensor data.

2.3.1 Networking Architecture

Fog networking is a new architecture that provides storage, communication, control, configuration, measurement and management between terminal devices and the Internet with features,
including location awareness, geographic distribution and low response latency [84], [19]. In fog networking, a huge number of decentralized mobile devices or vehicles can self-organize to communicate and potentially collaborate with each other via a fog node located at the edge of the Internet. There are several dimensions in fog architecture in terms of the current standard practice [21]. The essential data is stored near the end user rather than being stored in data centers. Moreover, instead of all traffic routed through the backbone network, fog performs a substantial amount of communication at or near the end-user. Furthermore, a fundamental amount of management, including network measurement, control and configuration, at or near the end-user is carried out. Each node in the fog network must be able to act as a router for its neighbors and be flexible to node mobility. More precisely, data collected by sensors are sent to devices like network edge, routers, access point for processing, and not sent to cloud server thus fog computing paradigm reduces bandwidth traffic issues. Also, fog computing improves the quality of service and minimizes latency. Therefore, fog computing plays an important role by reducing the traffic of data to the cloud and not delaying the computation and communication due to its placement near the data source.

2.3.2 Fog Computing Features

The fog computing paradigm provides a number of features that make the fog is a significant extension of the cloud.

- **Low latency and location awareness.** Applications with low latency requirements such as RSCMS can be supported by the fog at the network edge. In order to fulfill better latency, fog provides the computation close to the vehicles who sense and generate the data.

- **Widespread geo-distribution.** In contrast to the centralized cloud, fog is widely distributed. For instance, fog will play a significant role in delivering high quality streaming to connected vehicles through proxies such as RSUs positioned close to each other.

- **Large-scale.** Because of the wide geo-distribution, there could be numerous fog nodes in a local region such as a city. These fog nodes have the possibility to collaborate between each other to provide fog services to different end users.
• **Mobility.** Fog supports this promising feature that each fog-based application can immediately interact with its nodes, i.e., vehicles during moving.

• **Real-time application processing.** Fog-based applications (e.g., RSCMS) require real-time interaction for fast and sophisticated services. While patch processing is a helpful characteristic that cloud computing paradigm provides, fog computing supports an efficient way by processing real-time data.

### 2.4 FVCS Architecture

FVCS architecture inherits the advantages of VCS and fog computing paradigms. They are integrated together to form the new proposed framework that has unique characteristics, including real-time processing, location awareness, geo-distribution, and communication efficiency. RSUs as fog nodes are upgraded to have computational capabilities and storage spaces for offering computational and storage services to vehicles. Furthermore, they act as geography-related local servers to recruit a set of mobile users to perform the tasks. Therefore, RSUs as fog nodes are much powerful than RSUs in the conventional VANET, which is considered as a self-organized network to facilitate inter-vehicle communications, vehicle-to-roadside communications, and the Internet access with relay by RSUs. The proposed framework reduces the overhead of the cloud servers and also improves the accuracy of task allocation. FVCS is mainly composed of an organization, vehicles, RSUs as fog nodes and cloud servers.

• **Organization:** An organization releases their vehicular crowdsensing tasks on the cloud server in order to help perform these tasks because the organization does not have sufficient resources to accomplish these tasks individually.

• **Vehicles:** Each vehicle is equipped with various sensors and a powerful OBU. It can communicate with nearby vehicles and fog RSUs. The computations are performed by the OBU, which can collect data from on-board sensing devices and upload them to the nearby fog nodes.
• **RSUs as fog nodes:** RSUs are placed on the edge of the network and close to end users. Specifically, they are deployed along the road-side or at critical points. Therefore, they have complete knowledge about the vehicles in their coverage area and one-hop connection with these vehicles. Unlike RSUs in traditional VANETs, RSUs as fog nodes are equipped with storage space, computational and communication devices.

• **Cloud servers:** They have enormous capabilities for storage and computational, which can provide various services to the entire system. They communicate with an organization such as an insurance company for releasing spatial crowdsensing tasks and delivering results as well. Hence, deploying RSUs as fog nodes helps reducing the burden on the cloud server.
and performing the computations at the edge. RSUs also send the results to the cloud and the registered vehicles. Therefore, the collaboration between cloud and fog avoids sending all the data generated by vehicles to the cloud for processing, and thus achieves low bandwidth and better latency.

As shown in Figure 2.1, cloud, fog and vehicle layers have formed the architecture of FVCS. These layers cooperate with each other within the FVCS framework.

In the vehicle layer, vehicles can perform the tasks using its own mobile devices with capabilities of data sensing, processing and communications. In fact, the vehicles can collect road condition information from their on-board sensors during driving and then submit their crowdsensing reports to the local RSUs.

In the fog layer, RSUs are responsible to recruit a set of vehicles to perform the tasks. They use short range communication devices to communicate with the driving-through vehicles in their coverage regions. Thus, RSUs collect and process crowdsensing reports outsourced by vehicles. In addition, they distribute the results to the registered vehicles and the cloud. In particular, RSUs perform computations and make decisions close to the end users.

In the cloud layer, when the organization releases spatial crowdsensing tasks on a cloud server, the cloud server then assigns these tasks to fog nodes RSUs based on the spatial information of tasks, e.g., the sensing areas. For example, to measure the traffic congestion in the downtown area, the cloud server should assign the fog nodes located in the downtown area. In addition, the cloud can receive the results from the RSUs as historic information to be utilized later via the organization. Furthermore, the organization can collect generated data by vehicles from the cloud that is capable to analyze certain data collected by RSUs.

In this thesis, we focus our attention to enhancing security and privacy in data transmission where vehicles generate crowdsensing reports in the proposed framework. Meanwhile, we assume that the spatial tasks are protected since there have been many studies concerning security and privacy for spatial tasks assignments either in cloud-based or fog-based [63], [4], [6], [5].
2.5 FVCS Applications

Since modern vehicles are equipped with sophisticated sensors and on-board units (OBUs), VCS increasingly becomes targeted and ideal for several applications. There are some applications using MCS-based via supporting drivers to collect and contribute data using sensing and computing mobile devices (i.e. smartphones) such as Google maps and Waze. Nevertheless, the data obtained from mobile device sensors are not accurate enough to estimate the road condition due to sensing capability limitations in mobile devices [82]. In addition, MCS-based is prone to a high cost of bandwidth and time delay issues. Therefore, the sensing capabilities in smart vehicles can improve generating accurate and efficient data to assess the road and traffic condition. Meanwhile, deploying RSUs as fog nodes helps to tackle the cloud drawbacks. Motivated by the various applications found in current literature [64], [29], [85], we consider that the safety-related application RSCMS is the application scenario we integrate into our proposed FVCS framework.

In RSCMS, the detection of road surface abnormalities (e.g., potholes, bumps, ice) and their locations contribute to the improvement of road conditions and drivers’ safety. Road quality assessment has been identified as an important issue related to the possibility of making drivers and passengers more comfortable and safe more efficiently. The presence of road damage or abnormalities also worsens the energy efficiency of vehicles during driving since it determines an increase in fuel and consumption of vehicles’ components, especially brakes and suspensions.

The sensing devices equipped on vehicles such as GPS, accelerometer, and camera offer the possibility of obtaining real-time information about road features. Thus, the vehicles can upload road condition reports to fog nodes. Then, the organizations (e.g., transportation agencies or municipalities) can query the road surface abnormalities in the region of their jurisdiction to the cloud, which can automatically recognize the road problems for prioritizing road repair according to the data and results provided by fog nodes located in that region. Despite the proposed FVCS is a promising solution as an infrastructure for adopting real-time application in VCS, the public may not accept our proposed framework and threaten themselves if it lacks security measures to ensure privacy, integrity, and authenticity of the data they contribute.
2.6 FVCS Security and Privacy Challenges

Obviously, the objective of developing RSCMS on the basis of FVCS is to improve road quality and safety, e.g., reducing accident rates and protection vehicles from getting damaged. Nevertheless, the design of FVCS applications brings many challenges in terms of security and privacy. In fact, security and privacy in FVCS applications should be considered as important as securing other networks in computing. Because of the unique features of the VCS networks, such as high mobility and an extremely large amount of network entities (i.e., the vehicles and RSUs), the issues on security and privacy in FVCS applications become more challenging. Indeed, authorized users and adversaries may exist in the same environment and share the same privileges. Consequently, the utilization of these privileges such as accessing to confidential data or even tampering with integrity of data can be exploited by adversaries for further malicious intentions, and fatal to other users. However, the vehicle owners concern about their private information such as identity and location. Also, they may not trust the reports transmitted from another vehicle. Thus, they may not be willing to participate and contribute in FVCS applications while there is no guarantee for protecting their private data. Hence, solving security and privacy challenges in FVCS applications has top priority and is necessary for any vehicle to participate. To be more precise, there are a number of possible security and privacy attacks in FVCS. These are discussed in a detail as follows:

- Personal information leakage.

  In the FVCS applications (e.g. RSCMS), a malicious vehicle may be interested in the road event reports that might have sensitive information generated by other vehicles. Consequently, we take this potential attack into account as disclosing the source of private information. For instance, during a vehicle journey, it records a road event report, which could include important information about the vehicles or the report itself, and then sends it to the local RSU. Therefore, a malicious vehicle is able to eavesdrop on the reports and reveals the confidentiality of the data such as vehicle’s sensitive information. Moreover, malicious RSU can easily reveal the sensitive information of reports generated by vehicles. In addition, vehicle’s identity and location are a major critical issue in terms of the privacy protection [49]. In fact, the majority of VCS applications needs vehicles identity and location that can be easily disclosed by an attacker. To be more precise, the generated reports
from the surrounding environment may be related to some aspects of the drivers or even passengers and their social setting. For example, where drivers usually go, head, visit, or which activity they prefer to do in vehicles. As a result, these reports have to be encrypted in order to protect the sensitive information and thus address these possible issues.

- **Report modification attack.**
  A road event report can be forged by an attacker who modifies the report and forwards it to the RSU. This crucial attack may cause damages; for example, neither malicious vehicle nor RSU may fabricate the reports during report transmission. If the RSU accepts forged reports generated by malicious vehicles, it then performs computations on these reports and thus provides false results to the entire network. Therefore, integrity and mutual authentication should be achieved. Thus, only the original messages from legitimate users are accepted and the receiver can verify the sender’s report. This insures that the sensitive information has not been modified by an unauthorized user.

- **Impersonation attack.**
  The adversary may pretend to be another vehicle or even an RSU to deceive the others by sending bogus information to meet his own purpose. In order to address this attack, mutual authentication property should be implemented and achieved between system participants in the FVCS application.

Therefore, in order to tackle the aforementioned challenges, a set of novel and promising mechanisms for achieving security and privacy in the FVCS applications is developed. Specifically, cryptography techniques can be used to achieve security requirements including confidentiality, authentication, integrity, and privacy.

### 2.7 Applied Cryptography to FVCS

In this section, an overview of various cryptographic techniques most used in many technologies and academic works for FVCS are provided. Due to the features of FVCS, several security and
privacy issues have been emerging in this framework. For instance, the proposed framework offers services such as widespread geo-distribution, scalability and mobility to a massive number of vehicles. Thus, this may lead to critical issues in terms of authentication and protecting vehicles privacy during data transmission. In fact, we exploit the advantage of using some certain cryptographic techniques such as CL-PKC, symmetric/asymmetric, signcryption and homomorphic encryption to guarantee that the data traffic through the system is secure and achieves privacy preservation. These techniques are used in this study, and their basic concepts are introduced in more detail later.

2.7.1 Certificatless Public Key Cryptography (CL-PKC)

The deployment and management of infrastructure is significant to support the authenticity of cryptographic keys. Therefore, there is a need to provide an assurance to the user about the relationship between a public key and the identity of the holder of the corresponding private key. This assurance is carried out in a traditional Public Key Infrastructure (PKI) especially in the form of certificate that is used to prove the ownership of a public key by a certification authority (CA) [1]. However, there are issues associated with certificate management, including revocation, storage, distribution and the computational cost of certificate verification. These are particularly acute in computational or bandwidth limited environments [24].

Identity-based public key cryptography (ID-PKC) was first proposed by Shamir [78] who improved the way of authenticity of keys in PKI. In ID-PKC, each user uses his identity as the public key while his private key will be generated by a trusted third party such as private key generator (PKG). The first fully practical and secure ID-PKC encryption scheme was presented by Boneh et al. [17] by using elliptic curves with bilinear pairings. However, the dependence on a PKG to generate private keys introduces a key escrow inherent problem. For instance, if the PKG is compromised, it can access any ciphertext generated by any user in the system.

In order to address these issues, Alriyami et al. [3] introduced a new paradigm for public key cryptography, which is CL-PKC. It solves the key escrow problem in ID-PKC while maintaining its good certificate free property. Ordinarily, a key generation centre (KGC) is given complete control and implicitly trusted to generate the keys. The key generation process is divided between the KGC and a user in order to prevent a complete breakdown of the system in the case of KGC
being compromised. The private key has two parts; one is the partial private key generated by
the KGC; the second part of the key is a random secret value generated by the user and is never
revealed to anyone, including the KGC.

2.7.2 Symmetric and Asymmetric Cryptosystems

A cryptosystem is a pair of algorithms that take a number of keys in order to convert plaintext
to ciphertext [83]. These keys are called encryption keys. In particular, the system that uses the
same cryptographic key for both encryption of plaintext and decryption of ciphertext is referred
to as symmetric key cryptography. On the other hand, the system that uses different keys for
encryption and decryption is referred to as a public key or asymmetric cryptosystem. Menezes et
al. [56] have well studied the pros and cons of each of these cryptosystems. Extensive compar-
isons in terms of key length, hash function, digital signature and computational performance are
provided in [56]. In the literature, asymmetric and symmetric cryptosystems methods have not
widely been used in VCS. Therefore, on the base of those cryptosystems, we design secure and
privacy-preserving mechanisms in FVCS with a variety of security purposes, e.g., authentication,
confidentiality, batch verification and privacy-preserving techniques.

2.7.3 Signcryption Technique

A signature and encryption approach is a conventional way to guarantee the confidentiality and
integrity of a message [7]. In particular, this approach is digitally signing a message and then
encrypting it. This approach is done in two steps. However, the main disadvantage of signature
and encryption approach is that the cost involved is the sum of the signing the message added to
the cost of encrypting it. It consumes more machine cycles and bloats the message by introducing
extended bits to it. Hence, decrypting and verifying the message at the receiver side could spend
a large amount of computational power [7].

In order to address this issue, a promising paradigm in the public key cryptography named
signcryption technique [100] has been introduced. It simultaneously fulfills both the functions
of digital signature and public key encryption in a logically single step. Although the signcryp-
tion technique performs the signature and encryption simultaneously, the computational costs
and communication overhead are much lower compared to the signature and encryption approach [7]. Multiple CL-PKC schemes have been proposed and exploited the advantage of sign-cryption technique [8], [93], [95], [46]. Therefore, based on this promising technique, we build our certificateless aggregate signcryption (CLASC) scheme.

2.7.4 Homomorphic Encryption Technique

Homomorphic encryption is used to perform arithmetic computations on ciphertexts without knowing the private key (without decryption) and generate a new encrypted result. When the ciphertext is decrypted, the result will give the same result when doing arithmetic on a plaintext [87]. The purpose of using the homomorphic encryption technique in this work is to help facilitating the analysis and detecting of the replicated data among the encrypted messages. For example, given $n$ vehicles and their encrypted crowdsensing reports $(CSR_1, ..., CSR_n)$ where $CSR_i$ denotes encrypted crowdsensing report of vehicle $i$ and so on. As a result, we can detect a set of users who have the same crowdsensing report while they are encrypted.

In this work, the homomorphic concept and signcryption technique are integrated based on CL-PKC to achieve an efficient privacy-preserving mechanism in order to protect the system from any security threats and privacy risks.

2.8 Related Work

Extensive studies have been conducted to address the challenges caused by poor surface road conditions, and tackle security and privacy issues in FVCS. This section provides a brief review according to the existing systems for RSCMS and the most recent privacy-preserving schemes with respect to CLASC. Furthermore, we investigate the applied cryptographic approaches (e.g., compromised user revocation and secure deduplication) that are part of this work.
2.8.1 Road Surface Condition Monitoring System

Modern devices especially mobile devices have made sensing capabilities possible through the use of multiple powerful embedded sensors including accelerometers, gyroscopes and GPS systems, among others. We thus evaluate multiple scenarios/applications where mobile sensors are used in detection and reporting road surface conditions.

Eriksson et al. proposed Pothole Patrol (P2) [29], a mobile sensing application used in detection and reporting of road surface condition. In this system, they used a taxi cabinet in which multiple accelerometer sensors were placed and used in the collection of multiple predefined patterns associated with road surface anomalies via manual labelling. In the experiment, Eriksson et al. equipped taxis with an embedded Linux computer system and were able to detect more than 90% of potholes.

In a similar system used in traffic sensing and communication, Mohan et al. [61] proposed Nericell that can detect and report road conditions using the built-in sensors (e.g., accelerometers, GPS) in mobile phones. The information was further collected into traffic maps to be shared by the public.

Further, Mednis et al. [55] improved on the P2 system using a customized embedded gadget and extended the approach using vehicular sensor networks. It is operated using wireless sensor networks with the help of smartphones hardware platform for sensing road surface conditions [85]. Specifically, they have proposed CarMote that is a promising road monitoring platform. For instance, accelerometer and camera sensors are used for capturing photos and videos that further analyzed to extract road features.

Although these RSCMS MCS-based help provide promising solutions to enhance the road quality and safety, MCS-based still has critical challenges in terms of collecting accurate data about the roads. Specifically, mobile devices do not have high powerful computation and also suffer from sensor limitations such as finding nails on the roads. Furthermore, a majority of those RSCMS is designed based on cloud computing architecture. Therefore, in order to address these challenges and being different from the above works, VCS and fog computing paradigms are integrated to propose an efficient and novel FVCS framework to be a foundation for RSCMS.
2.8.2 Certificateless Aggregate Signcryption Schemes

In the proposed framework, we design various privacy-preserving schemes based on CL-PKC. Furthermore, we utilize a signcryption technique in order to fulfill confidentiality, integrity, and authenticity simultaneously, which presents high efficiency in terms of computational cost and communication overhead. The proposed privacy-preserving schemes mainly rely on the aggregation signcryption technique. The concept of signcryption technique was first introduced by Zheng [100]. On the other hand, Boneh et al. [18] proposed the aggregation concept, which is a digital signature scheme. To be more precise, the focus of this work will be on existing certificateless aggregate signcryption schemes literature.

Certificateless public key cryptography was first proposed by Al-Ryami and Paterson [3] as a way of overcoming the challenges associated with key escrow as applied in cryptography approaches that are identity-based. In particular, the private key is divided into two parts. One is generated by the trusted third party in the form of a partial private key while the second part is computed by a user himself in the form of a secret value. These two parts totally create the full private key for the user. There are several schemes proposed in encryption [77], [25], digital signature [40], [41], and signcryption [8], [93], [95], [46], certificateless cryptography.

Since we are using certificateless aggregate signcryption, we first evaluate multiple aggregate signcryption as used in identity-based aggregate schemes of signcryption [75], [43]. However, these schemes have security issue which is a key escrow problem. In order to solve this problem, Lu et al. [51] and Eslami et al. [31] are the only works focusing on designing certificateless aggregate signcryption in the literature. They argued in favour of CLASC as a secure system. CLASC is emphasized in [51] as an appropriate secure model as has been proven in its use in the random oracle model [14].

Nonetheless, the scheme as currently constituted requires significant improvements over pairing maps that can potentially lead to a promising low computational scheme in addition to lowering time consumption. Therefore, we propose a new and efficient CLASC scheme by building on the random oracle model [14].
2.8.3 Compromised User Revocation

This subsection begins by investigating some of the existing VANETs revocation schemes, revocation in public key cryptosystem and revocation in CL-PKC setting.

Revocation in VANETs

The family of standards IEEE 1609 describes the use of a PKI in VANETs. Jean et al. [42] analyzed a proposal for the use of a PKI to protect messages and mutually authenticate entities in VANETs. Lin et al. [48] investigated some certificate revocation protocols introduced in the traditional PKI architecture. They concluded that the most commonly adopted certificate revocation scheme is through certificate revocation list (CRL), using central repositories prepared in certificate authorities (CAs). Based on such centralized architecture, alternative solutions to CRL could be used for certificate revocation system like certificate revocation tree (CRT) and the online certificate status protocol (OCSP) where the common requirements for these schemes is high availability of the centralized CAs. Frequent data transmission with OBUs to obtain timely revocation information may cause significant overhead. Thus, with the high-speed mobility and extremely large amount of network entities in FVCS, the centralized CRL architecture may be far from realistic.

To tackle the problem, Raya et al. [69] discussed the current methods of revocation and its weaknesses, and proposed a novel protocol for certificate revocation, including CRL, revocation using compressed certificate revocation lists (RC2RL), revocation of the tamper proof device (RTPD) and distributed revocation protocol (DRP), stating the differences among them. They also made a simulation on the DRP protocol concluding that the DRP protocol is the most convenient one which used the bloom filter. Also, Jason et al. [38] proposed the use of Bloom filter to store the revoked certificate, and dedicate the CRL just to sign the revocation key for each vehicle. Using bloom filter will increase the speed for searching in it while it is a probabilistic function, and may give wrong information. For example, the certificate may be in the list, and the result shows that it is not in the list. Zhang et al. [98] and Wang et al. [92] proposed the efficient privacy-preserving protocols involved with CRL methods, however they still use the idea of CRL that leads to high computational overhead and is not suitable for implementing in FVCS.
In order to solve the problem caused by the management of valid public-key certificates, we propose a revocable privacy-preserving protocol CL-PKC-based.

2.8.3.1 Revocation in public key cryptosystem

The revocation mechanism is significantly needed in order to revoke the malicious users from the system. The revocation problem is resolved in PKI [2], [28] by using CRL and OCSP [62]. PKI uses a certificate to bind a public key with its user identity [27], [72]. However, the issues associated with certificate management are complicated and expensive. The certificate based PKI management suffers from long computational delay in the RCL checking in case of applying to a scalable environment, i.e., VCS applications. To get rid of the complex and costly certificate management problem in the traditional PKI, ID-PKC effectively uses a user identity information as its public key [78]. Boneh and Franklin [17] suggested a method that the PKG generates private keys for all non-revoked users periodically. Boldyreva et al. [16] utilized a binary tree to present the first scalable revocable identity-based encryption scheme, which was later improved by Libert and Vergnaud [47]. Many ID-PKC revocation schemes [76], [94] are proposed, but these schemes have an inherent problem of key escrow. Alriyami et al. [3] introduced CL-PKC that solves the key escrow problem in ID-PKC while remaining its good certificate free property.

2.8.3.2 Revocation in CL-PKC setting

Up to now, one of the available solutions to revocation in CL-PKC is to employ an online trusted third party that is different from Key Generator Center (KGC), called Security Mediator (SEM) [22], [96]. The KGC divides a user partial private key into two components. One component is sent to the user while the other is delivered to the SEM. Both communications require secure channels. A user cannot perform decryption or signing without the online help of the SEM. To revoke a compromised user, the SEM just simply stops supplying online help for that user. However, this inevitably increases the complexity of the system and makes the system inconvenient for practical applications. Another serious drawback of the revocation mechanism using SEM lies in the fact that the SEM must keep a large amount of secret key component of users. This not only brings a heavy burden to the SEM but also introduces a bottle neck
Another existing revocation mechanism is to require the KGC to generate users’ partial private keys at predetermined regular time periods [80], [90], [86]. To revoke a user, the KGC just stops sending fresh partial private keys of a new time period for that user. In the proposed protocol, the KGC is in charge to produce users’ partial private keys at predetermined time periods.

### 2.8.4 Secure Deduplication

In terms of secure deduplication, extensive studies have been conducted recently under the consideration of cloud storage environments not specific to fog storage environments. Several secure deduplication schemes [99], [39] have been proposed under the assumption that the servers are fully-trusted. In particular, the end users outsource plaintext reports that the server-side performs the data deduplication on these reports. There is a crucial challenge in case the server-side is considered as a semi-trusted, so that the end users have concerns about their private information from being disclosed. In order to address this issue, Douceur et al. [26] proposed convergent encryption (CE) to provide data confidentiality in deduplication. In this study, users derive a convergent key $K$ from each original data copy $M$ and encrypt the data copy with $K$ to get the ciphertext $C$. Also, they derive a tag for the data copy such that tag will be used to detect duplicates. Following this concept of deterministic key derivation from the data itself, Bellare et al. [13] presented a generalized framework, called message-locked encryption (MLE). MLE is categorized into four particular schemes and assessed accurately according to levels of integrity and security guarantees. These schemes have addressed duplicate-faking attacks for space efficient secure outsourced storage.

One supposed scenario of the proposed framework scenarios is in case that RSUs are considered as semi-trusted. Therefore, the privacy concerns about vehicles private information is taking into account to provide a secure and efficient way to manage replicated contents without learning any information about vehicles. We exploit the advantage of fog computing to improve efficiency and latency, and CE to support deduplication feature in the ciphertexts. The proposed scheme is a hybrid scheme including secure deduplication and signcryption techniques. Since we adopt CE in our work, we do not consider the local brute force attack because Bellare et al. [12] proposed dupLESS that prevents this kind of attack and offered a secure, easily-deployed
solution for efficient outsourced storage supporting data deduplication. To the best of our knowledge and different from the existing works, we propose a first deduplicated reporting scheme based on the CLASC to delete the repeated data without scarifying the confidentiality, integrity and authenticity of the reports.
Chapter 3

A Privacy-Preserving Protocol for Fog-Assisted Vehicular Crowdsensing

3.1 Introduction

Although road surface condition information is seen as a useful system in transportation infrastructure, security and privacy issues need to be addressed before its implementation into FVCS. In reality, crowdsensing report transmissions experience major challenges due to the privacy sensitivity of road event information, as well as the unauthentic interconnection of vehicles and the corresponding road infrastructure, including RSUs. A number of issues need to be addressed in the design of a security protocol, including a guarantee that a road event report is not accessed at the time of transmission by unauthenticated users as well as consideration for its scalability. It is supposed that the generated reports remain encrypted, hence the system should not only be able to verify but also to simultaneously decrypt reports while providing low computational and communication costs. Additionally, the protocol should attain mutual authentication among vehicles and RSUs. Further, the protocol should be lightweight as a result of constraints due to the dynamic nature of VCS. The protocol also needs to retain its robustness when there is a threat; for instance, in a case where the authentication keys remain exposed. Furthermore, a key generation center (KGC) may be represented by a commercial organization that can naturally misbehave by illegally collecting and accessing vehicle sensitive information. Thus, a KGC cannot be fully
trusted to generate the users’ private keys.

In order to successfully address the aforementioned issues, CL-PKC [3] is used in pursuing the security objectives. CL-PKC avoids the often experienced key escrow problem that is associated with ID-PKC since the user’s private key is not only offered by the KGC but by a combination of the KGC’s partial private key and the user’s secret value. As a result, the KGC lacks information about the user’s full private key. Furthermore, CL-PKC successfully evades certificate management with regard to certificate-based public key cryptography, such as revoking, distributing and storing data. In order to achieve efficiency in terms of computational cost and communication overhead, this present work adopts a signcryption technique to accomplish both encryption and signature in one logical step.

To adjust the current work by adopting a signcryption technique, certificateless signcryption (CLSC) schemes are used in capturing communication with respect to both confidentiality and unforgeability. The first CLSC scheme was proposed by Barbosa and Farshim [8] using a formal security analysis as evidenced in the random oracle model. The proposed scheme is premised on a process of aggregation, which lowers the volume of exchanged information and signature verification as well as massive data unsigncryption, thus attaining scalability, and lower computational and communication costs. These can be achieved with a single step of particular importance to low communication network bandwidths as well as computationally restricted environments. Although Eslami et al. [31] and Lu et al. [51] proposed certificateless aggregate signcryption (CLASC), these schemes are realized using many pairing operations that may lead to high computational cost and time consumption if there is an increase in the number of vehicles.

This chapter presents an efficient privacy-preserving protocol to solve the aforementioned security and privacy challenges in FVCS. A review of previous research suggests that, in building a CLASC scheme, this is the most efficient work to date. The main contributions of this chapter are as follows:

- A new and efficient CLASC scheme is proposed. This scheme offers a significant improvement over the pairings required by aggregate signature verification and unsigncryption that are needed in order to protect crowdsensing reports from being disclosed.

- Based on the proposed CLASC scheme, a privacy-preserving protocol for enhancing security and privacy in FVCS road surface condition monitoring systems is also proposed.
• The proposed scheme resists various security and privacy threats. It also preserves user privacy while achieving lightweight aggregation. Furthermore, as a control center (CC) cannot be fully trusted, the proposed scheme shows a CC cannot reveal private keys even if compromised.

• Extensive simulations were conducted in terms of computational cost and communication overhead. The simulation results well demonstrate that the proposed scheme is much more efficient compared to existing schemes [31], [51] and outperforms them.

The rest of this chapter is organized as follows. In Section 3.2, the system models and design goals are presented followed by the preliminaries in Section 3.3. In Section 3.4, the CLASC scheme is introduced in detail. The proposed privacy-preserving protocol is described in Section 3.5. Security analysis is given in Section 3.6 followed by performance evaluation in Section 3.7. A summary of the chapter is presented in Section 3.8 while the limitation of this chapter is discussed in Section 3.9.

3.2 System Models and Design Goals

This section presents the system model, threat model and design goals.

3.2.1 System Model

Motivated by the various applications found in the current literature, a road surface condition monitoring system comprised of a CC, vehicles, RSUs as fog devices, and cloud servers, is considered (see Figure 3.1).

• A CC is a powerful entity in charge of the entire system and responsible for initializing the system. In the proposed scheme, the CC works as the key generation center. As the function of the CC may be undertaken by a commercial organization, it cannot be fully trusted. Therefore, to avoid the key escrow problem, it only generates a partial private key for the participants and is blocked from accessing both vehicle and RSU sensitive data. It is assumed that the CC is powered with sufficient computational and storage capabilities.
• Vehicles generate a significant volume of data, including time, location and road event indicators, such as potholes or slippery surfaces (see Figure 3.1)

• An RSU is considered to be an efficient computational and storage device that can extend the cloud services to the edge. RSUs have the ability to react and make decisions close to the end users. All the real-time data sensed by the vehicles are sent to the RSU for immediate processing. As an example, once processed, the RSUs can send an alert regarding road hazards at a specific location.

• Cloud servers are the data centers of the system. System data, such as historic information, are stored in the cloud to be later utilized. The advantage of a fog device is that instead of sending all the data generated by the vehicles to the cloud for processing (which can lead to high bandwidth cost and high latency), RSUs perform the computation at the edge and only send the results to the cloud and the connected vehicles.
3.2.2 Threat Model

The threat model assumes that the connection between RSUs and cloud servers is secure and that they are fully-trusted components. In this present work, attention is focused on the threat to data generated by vehicles, which is then forwarded to the RSUs. As shown in Figure 3.2, road event reports that are devoid of content-oriented privacy may result in eavesdroppers disclosing the road event report of the source, causing the receiver to be given false road event reports. Malicious attackers may modify or fabricate the data for their own purposes. Particularly, the adversary can control the entire communication channel and monitor all the data passing through the channel. The adversary can also tamper with the message, drop some packets or even replace the original message. Furthermore, all the data transmitted to/through RSUs and vehicles can be
intercepted and analyzed by the adversary. In particular, a vehicle could become malicious by generating false reports for its own benefit; for example, it could gain credits for contributing to a crowdsensing task. Ultimately, the CC as a key generation center cannot be fully trusted because it may be represented by a commercial organization. By considering the commercial benefits, it is natural for a CC to misbehave in ways such as illegally collecting and accessing vehicle sensitive information. It may also disclose users' authentication keys and fabricate road event reports. Therefore, in order to prevent an adversary or malicious vehicles from violating vehicle privacy during the generating of crowdsensing reports, the following security goals should be satisfied.

3.2.3 Design Goals

The present work aims to achieve the following security and performance objectives, based on the system model and potential threats.

1. Security objectives

   • *Data confidentiality and integrity.* All accepted crowdsensing reports should be delivered unaltered, and the origin of these reports should be protected from revealing private and sensitive information.

   • *Mutual authentication.* The vehicles and the RSU should authenticate each other in order to guarantee that the data is from the source and, once received, is unaltered.

   • *Anonymity.* Vehicle identity should be hidden from a normal message receiver during the authentication process in order to protect the sender's private information.

   • *Key escrow resilience.* The key generation center does not have the users' full private keys. Therefore, this proposal ensures that an adversary will not gain access to users' full private keys if a CC is compromised.

2. Performance objectives

   • *Low communication overhead and fast verification.* The security scheme should be efficient in terms of communication overhead and acceptable processing latency.
large number of report signatures should be first verified and then unsigncrypted within a short interval.

- **Robustness.** The crowdsensing reports generated via vehicles should not be accessed in case part of the private keys is infiltrated.

## 3.3 Preliminaries

This section starts with basic concepts and portrays the necessary complexity assumptions. Then, the framework and security model of CLASC are presented.

### 3.3.1 Bilinear Maps

We recall the bilinear pairing technique, which serves as the basis of our proposed CLASC. Let $G$ be an additive group of large prime order $q$, and $G_T$ be a multiplicative group of the same large prime order and $P$ be a generator of $G$. An admissible bilinear pairing $\hat{e}: G \times G \rightarrow G_T$ is a map with the following properties.

- **Bilinearity:** for all $P,Q \in G$ and $a, b \in \mathbb{Z}_q^*$, we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$

- **Non-degeneracy:** $\hat{e}(P, Q) \neq 1_{G_T}$ where $1_{G_T}$ denotes the identity element of group $G$.

- **Computability:** There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for $P,Q \in G$. An admissible bilinear pairing $\hat{e} : G \times G \rightarrow G_T$ can be implemented by the modified Weil/Tate pairings over elliptic curves [17].

**Definition 1:** Bilinear Parameter Generator. A bilinear parameter generator $Gen$ is a probabilistic algorithm that takes a security parameter $k$ as input, and outputs a 5-tuple $(G, G_T, \hat{e}, P, q)$ where $q$ is a $k$-bit prime number, $G$ and $G_T$ are two groups with order $q$, $P \in G$ is a generator, and $\hat{e}$ is a non-degenerated and efficiently computable bilinear map.
3.3.2 Complexity Assumptions

We recall the following intractability assumptions related to the security of our scheme.

**Definition 2:** Computational Diffie-Hellman (CDH) problem. Given \( P, aP, bP \in G, \forall a, b \in \mathbb{Z}_q^* \), the CDH problem is to compute \( abP \in G \) probability within polynomial time.

**Definition 3:** Decisional Bilinear Diffie-Hellman (DBDH) problem. Given \( P, aP, bP, cP \in G, \forall a, b, c \in \mathbb{Z}_q^* \) and \( x \in G_T \), DBDH problem is to decide whether \( x = \hat{e}(P, P)^{abc} \).

3.3.3 Framework of Certificateless Aggregate Signcryption

Based on Eslami et al. [31] and Lu et al. [51], we first define the participants involved in a framework of a CLASC scheme. They are composed of four parties which are: a KGC, an aggregating set \( ID_i \) of \( n \) users with an identity \( \{ID_i\}_{i=1}^n \), a receiver with an identity \( ID_R \) and an aggregate signcryption generator. The framework of a CLASC scheme is defined by the following seven probabilistic polynomial time (PPT) algorithms.

- **Setup:** This algorithm takes a security parameter \( k \) as input and outputs system parameters \( params \) and a master private key \( s \), a corresponding master public key \( P_{pub} \). Then, the KGC carries out the algorithm and publishes \( params \). The key \( s \) is kept secure.

- **Partial-Private-Key-Extract:** Given the system parameters \( params, s \) and identity \( ID_i \) of an entity \( i \). It returns a partial private key \( D_i \). Then, the KGC calculates the algorithm to generate \( D_i \) that is sent to the corresponding user \( i \) through a secure channel.

- **User-Key-Generate:** This algorithm is run by each user and takes \( params \) and user’s identity \( ID_i \) as input. It returns a randomly chosen secret value \( x_i \) and a corresponding public key \( Y_i \) for the entity. Then, the user generates his own public key and then publishes it.

- **Signcrypt:** This algorithm runs by each user \( ID_i \) in an aggregating set of \( n \) users \( \{ID_i\}_{i=1}^n \). It takes \( params \) and some state information \( \Delta \). All of the users must use the same unique state information in the signcryption algorithm for an aggregating set, a message \( M_i \), user’s identity \( ID_i \) with corresponding public key \( Y_i \) and private key \( (x_i, D_i) \), the receiver identity \( ID_R \) with corresponding public key \( Y_R \) as input. This algorithm returns a ciphertext \( C_i \).
• Aggregate: This algorithm is run by the aggregate signcryption generator and takes an aggregating set \( ID_i \) of \( n \) users \( \{ID_i\}_{i=1}^n \), \( \Delta \), user’s identity \( ID_i \) of each sender with corresponding public key \( Y_i \) and \( C_i \) on a message \( M_i \) as input. The message is ciphered with the state information \( \Delta \) with the receiver identity \( ID_R \) with corresponding public key \( Y_R \). It outputs an aggregated ciphertext \( C \) on messages \( \{M_i\}_{i=1}^n \).

• Aggregate-Verify: This algorithm is performed by the receiver \( ID_R \) and takes as input an aggregating set of \( n \) users \( \{ID_i\}_{i=1}^n \), user’s identity \( ID_i \) of each sender with corresponding public key \( Y_i \), the receiver identity \( ID_R \) with corresponding public key \( Y_R \), state information \( \Delta \), and an aggregated ciphertext \( C \). If the aggregate signcryption is valid, algorithm returns true otherwise false.

• Unsigncrypt: The receiver \( ID_R \) performs this algorithm that takes as input an aggregated ciphertext \( C \), state information \( \Delta \), the receiver full private key \( (x_R, D_R) \), his identity \( ID_R \) and public key \( Y_R \), and the senders identities \( \{ID_i\}_{i=1}^n \) with their corresponding public keys \( \{Y_i\}_{i=1}^n \). It returns a set of \( n \) plaintexts \( \{M_i\}_{i=1}^n \).

### 3.3.4 Security Model of CLASC

A certificateless cryptography may be subject to two types of adversaries [3]. Type I adversary may request entities public keys and replace keys with values of its choice but is not allowed to access the master private key. Type II adversary on the other hand may access the master private key but is not allowed to replace the public key of the entities. The CLASC scheme has two security objectives which are: confidentiality for the signcryption and encryption mode; and unforgeability for signcryption and signature mode. There exists an interactive game between a challenger \( C \) and an adversary \( \mathcal{A} \) to prove the security of a CLASC scheme. There are four games for confidentiality and unforgeability between \( C \) and type I, type II adversary respectively. Eslami et al. [31] provide details for the four games. Thus, to avoid reinventing the wheel, we refer to their work for the security model of a CLASC scheme and also, provide the definitions based on the games as declared in their work.

**Definition 4:** Confidentiality of CLASC. A CLASC scheme is semantically secure under adaptively chosen ciphertext attacks (IND-CCA2) if no PPT adversary (of either Type) has a
non-negligible advantage in Game I or Game II. As the adversaries can access the private keys of all of the senders, therefore; this definition assures that confidentiality is preserved even if these keys are compromised and insider security is guaranteed.

Definition 5: Unforgeability of CLASC. A CLASC scheme is existentially unforgeable under adaptively chosen message attacks (EUF-CMA) if no PPT adversary (of either Type) has a non-negligible advantage in the Game III or the Game IV. As the adversaries can access the private key of the receiver, therefore this definition assures that unforgeability is preserved even if this key is compromised and insider security is guaranteed.

3.4 Proposed CLASC scheme

In this section, we propose an efficient CLASC scheme that serves as the design basis for our privacy-preserving protocol.

We propose a solid and promising CLASC scheme that outperforms the existing schemes [31] [51]. They utilize the bilinear map that is an efficient way of pairing. However, their schemes suffer from high computational complexity because of the number of pairing operations for signcryption, aggregate, aggregate verification and unsigncryption. Therefore, we address this problem by reducing pairing operations and thus achieving low computational and communication cost. The proposed CLASC scheme is composed by the following six algorithms.

1. Setup: Given the security parameters $k$, and this algorithm is performed by the KGC as follows.

   - Chooses a cyclic additive group $G$ of prime order $q$ on elliptic curve, and $P$ is an arbitrary generator of $G$.
   - Chooses a cyclic multiplicative group $G_T$ of the same order $q$ and a bilinear map $\hat{e} : G \times G \to G_T$.
   - Randomly selects a master private key $s \in Z_q^*$ and compute the master public key $P_{pub} = sP$. 

• Selects four secure hash functions $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$, $H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^n$ here $n$ is the bit-length of plaintexts, $H_3 : \{0, 1\}^* \rightarrow G$ and $H_4 : \mathbb{Z}_q^* \rightarrow G$.

• Publishes the system parameter $\text{params} = (G, G_T, \hat{e}, P, q, P_{pub}, H_1, H_2, H_3, H_4)$ and the master private key $s$ will be kept secure by the KGC.

2. **Key-Generation**: This algorithm is interactively performed by the user $ID_i$ and KGC as follows.

   • The user $ID_i$ randomly chooses $x_i \in \mathbb{Z}_q^*$ as the secret value and computes a partial public key $Y_{ib} = x_i P$.

   • The user sends its identity and partial public key $(ID_i, Y_{ib})$ to the KGC.

   • The KGC then randomly selects $y_i \in \mathbb{Z}_q^*$ and compute another partial public key for the user $Y_{ia} = y_i P$, so the full public key for the user is $(Y_{ib}, Y_{ia})$.

   • The KGC computes the partial private key $D_i = y_i + s \cdot Q_i$ where $Q_i = H_1(ID_i)$, and $D_i$ is sent securely to the user $ID_i$.

   • The user $ID_i$ judges the validity of the partial private key by checking,

     $$D_i P = Y_{ia} + P_{pub} Q_i.$$ 

Notably, these procedures finish three different algorithms which are, *set-secret-value*, *partial-private-key-extract* and *set-public-key* of the proposed scheme. These algorithms generate public key $(Y_{ib}, Y_{ia})$ that is kept in the public tree by the KGC, and the full private key $(x_i, D_i)$ is kept secret by the user.

3. **Signcrypt**: This algorithm is performed by a sender $ID_i$ to signcrypt the message $m_i$ with $ID_R$ as a receiver. $ID_i$ performs the algorithm as follows.

   • $ID_i$ randomly selects $r \in \mathbb{Z}_q^*$ and computes,

     • $T_i = r P$.

     • $Z_b = r Y_{rb}$.

     • $Z_a = r (Y_{ra} + P_{pub} Q_i)$. 

38
• $h_a = H_2(Q_R||Y_{ra}||Y_{rb}||\Delta||T_i||Z_b||Z_a)$.
• $K_i = h_a \oplus m_i$.
• $h_b = H_3(Q_R||Y_{ra}||Y_{rb}||\Delta||T_i||K_i||Q_i||Y_{ib}||Y_{ia})$.
• $h_c = H_4(\Delta)$.
• $\alpha_i = D_i h_c + rh_b + x_i h_c$.
• Returns the ciphertext $C_i = (T_i, K_i, \alpha_i)$

4. **Aggregate**: This algorithm is performed by aggregator signcryption generator on the receiver $Q_R$ as follows.

  • Computes $\alpha = \sum_{i=1}^{n} \alpha_i$.
  • This algorithm outputs the aggregate ciphertexts $C = (T_1...T_n, K_1...K_n, \alpha)$.

5. **Aggregate-Verify**: This algorithm is run by a receiver $Q_R$ and computes the following.

  • $h_b = H_3(Q_R||Y_{ra}||Y_{rb}||\Delta||T_i||K_i||Q_i||Y_{ib}||Y_{ia})$, for $i = 1...n$.
  • $h_c = H_4(\Delta)$.
  • $Q_R$ verifies $\hat{e}(\alpha, P) = \hat{e}(\sum_{i=1}^{n} Y_{ia} + P_{pub} Q_i, h_c) \hat{e}(\sum_{i=1}^{n} T_i, h_b) \hat{e}(\sum_{i=1}^{n} Y_{ib}, h_c)$. If this equation holds, this algorithm outputs true otherwise false.

6. **Unsigncrypt**: If the output of Aggregate-Verify algorithm is true, this algorithm is performed by the receiver $Q_R$ who computes the following.

  • $Z_b'l = x_i T_i$.
  • $Z_a'l = D_i T_i$.
  • $h_{a'i} = H_2(Q_R||Y_{ra}||Y_{rb}||\Delta||T_i||Z_{bl}||Z_{al})$.
  • $m_i'l = K_i \oplus h_{a'i}$.
  • This algorithm outputs $\{m_i\}_{i=1}^{n}$. 

39
• The correctness of our signature scheme is as follows.

\[ \hat{e}(\alpha, P) = \hat{e}(\sum_{i=1}^{n} \alpha_i, P) \]
\[ = \hat{e}(\sum_{i=1}^{n} (D_i h_c + r h_b + x_i h_c), P) \]
\[ = \hat{e}(\sum_{i=1}^{n} D_i h_c, P) \hat{e}(\sum_{i=1}^{n} r P, h_b) \hat{e}(\sum_{i=1}^{n} x_i P, h_c) \]
\[ = \hat{e}(\sum_{i=1}^{n} D_i P, h_c) \hat{e}(\sum_{i=1}^{n} T_i, h_b) \hat{e}(\sum_{i=1}^{n} Y_{ib}, h_c) \]

• The correctness of our decryption scheme is as follows.

\[ m_i' = K_i \oplus h_{a'} \]
\[ = H_2(Q_i||Y_{ia}||Y_{ib}||\Delta||T_i||Z_b||Z_a) \oplus m_i \oplus h_{a'} \]
\[ = h_a \oplus m_i \oplus h_{a'} \]
\[ = m_i \]

### 3.5 Proposed Privacy-Preserving Protocol

In this section, we present the details of our privacy-preserving protocol. In this application scenario, RSUs are considered as fog devices, which are able to aggregate the crowdsensing reports, verify them and then perform decryption. Our certificateless aggregate signcryption scheme is introduced in the protocol to fulfill the design objectives. The proposed protocol consists of four steps: system initialization, data formulation and sending, secure road event report (SRER) aggregated verification, and data receiving.
3.5.1 System Initialization

The vehicles and RSUs register to the CC to generate their full public and private keys. Moreover, CC determines the format of road event report that is generated by the vehicles.

Given the security parameter $k$, the CC first generates the bilinear parameters $(G, G_T, \hat{e}, P, q)$ by running $Gen(k)$. Then, the CC randomly selects $s \in \mathbb{Z}_q^*$ as its master secret key and computes its master public key $P_{pub} = sP$. Additionally, the CC chooses four secure hash functions: $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$, $H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^n$ where $n$ is the bit-length of plaintexts, $H_3 : \{0, 1\}^* \rightarrow G$ and $H_4: \mathbb{Z}_q^* \rightarrow G$. After that, the system parameters params will be published, which include $(G, G_T, \hat{e}, P, q, P_{pub}, H_1, H_2, H_3, H_4)$.

A significant task of the setup procedure is to determine the format of secure road event report $SRER_{ij}$. For a road event $RE_i$, the vehicles $Sen_j$ will generate the data where $Data_i = (Time_{ij}, Location_i, Signals_i)$ and the $SRER_{ij}$ will securely forward to the RSU in the format $SRER_{ij} = (Q_j, Signcrypt(Data_i))$ where,

- $Time_{ij}$ denotes the time when the vehicle $j$ makes the claim on a road event $i$.
- $Location_i$ denotes the place where the road event takes place.
- $Q_j$ denotes the pseudo identity of the vehicle that generates the report.
- $Data_i$ denotes a report generated by a vehicle about road event.
- $Signcrypt_{ij}$ denotes the signcryption generated by the vehicle $Sen_j$ on the road event $RE_i$ that sends to RSU. Vehicles and RSUs can join the system by performing the following Steps.

- A vehicle $Sen_j$ can randomly choose $x_j \in \mathbb{Z}_q^*$ as its secret value and compute its partial public key $Sen_{jb} = x_jP$.
- $Sen_j$ sends its identity and partial public key $(Sen_j, Sen_{jb})$ to the CC for registration.
- The CC randomly selects $y_j \in \mathbb{Z}_q^*$ and compute another partial public key for the mobile sensor $Sen_{ja} = y_jP$.
- The CC then computes the partial private key $D_j = y_j + s \ast Q_j$, where $Q_j = H_1(Sen_j)$, for the register $Sen_j$ with partial public key $Sen_{jb}$.
- $D_j$ is sent to the $Sen_j$ via a secure channel. The full public key $(Sen_{jb}, Sen_{ja})$ is kept in the public tree by the CC.
• The vehicle $\text{Sen}_j$ receives the partial private key $D_j$ and concatenates with its secret value $x_j$ to form its full private key $(D_j, x_j)$.

3.5.2 Data Formulation and Sending

This part is performed by the source with a vehicle $Q_j$. A road event $RE_i$ is sensed by one or multiple vehicles and then $Data_i$, which include $(\text{Time}_{ij}, \text{Location}_i, \text{Signals}_i)$, is discovered. After that, $Q_j$ with encrypted $Data_i$ as a $SRER_{ij}$ sends to the RSU as fog device receiver. Then, $Q_j$ utilizes the certificateless signcryption algorithm on $Data_i$ as follows.

• $\text{Sen}_j$ randomly selects $r \in Z_q^*$ and computes the following,

  • $T_j = rP$.
  • $Z_b = rPK_{rb}$.
  • $Z_a = r(PK_{ra} + P_{pub}Q_j)$.
  • $h_a = H_2(Q_R || PK_{ra} || PK_{rb} || \Delta || T_j || Z_b || Z_a)$.
  • $K_j = h_a \oplus Data_i$.
  • $h_b = H_3(Q_R || PK_{ra} || PK_{rb} || \Delta || T_j || K_j || Q_j || \text{Sen}_ja || \text{Sen}_jb)$.
  • $h_c = H_4(\Delta)$.
  • $\alpha_j = D_jh_c + rh_b + x_jh_c$.

The ciphertext $C_j = (T_j, K_j, \alpha_j)$ is attached to SRER in the format as $SRER_{ij} = (Q_j, \text{Signcrypt}(Data_i))$, where $\text{Signcrypt}(Data_i) = C_j$. 

42
3.5.3 SRER Aggregated Verification

Notably, this application scenario is based on vehicles to infrastructure communication (V2I) which means vehicles can directly communicate with the RSUs. Once a road event $RE_i$ is sensed by one or multiple vehicles, they then generate a road event report $SRER_{ij}$ that includes accurate information such as time, location and the type of event. Suppose that RSCMS is deployed on the highway, that massive of objects can pass through. Therefore, a bunch of data will be generated by the various vehicles and sent to the closest RSU. If the RSU receives each ciphertext separately to verify the signature and then usingcrypt it, this process will have a long time that may lead to long delay. We exploit an advantage of fog devices, which are efficient in computational cost and bandwidth. Therefore, our protocol provides the aggregation property that the RSUs can aggregate all the ciphertexts generated by the multiple vehicles. This process provides a sufficient amount of efficiency over sending each ciphertext separately. Whenever receiving a SRER, the RSU can perform the SRER aggregation and SRER batch verification operations as follows.

1. SRER aggregation

   This algorithm is used to aggregate multiple SRERs into a single SRER. For a road event $RE_i$, given $n$ SRERs $SRER_{ij} = (Q_j, \text{Signcrypt}(Data_i))$ generated by vehicles $Sen_1, ..., Sen_n$, we can obtain $SRER_{agg} = (Q_1...Q_n, \text{Signcrypt}(Data_i)_1^{n})$. This algorithm is performed by the RSU as follows.

   - This algorithm takes a collection of individual ciphertexts $C_j = (T_j, K_j, \alpha_j)_j^{n}$ generated by vehicles with $(Q_j)_j^{n}$ to a receiver with identity $Q_R$ under the same state information $\Delta$, which is considered as a secret value to insure the aggregation phase.

   - It aggregates the signature parts of ciphertexts and then computes the signature aggregation $\text{sig}_{agg} = \sum_{j=1}^{n} \alpha_j$.

   - It outputs the aggregate ciphertexts $SRER_{agg} = ((Q_j)_j^{n}, T_1...T_n, K_1...K_n, \text{sig}_{agg})$.

2. SRER batch verification

   This step performs signature batch verification for all the ciphertexts simultaneously. Given the signature aggregation $\text{sig}_{agg}$, the report sets $(SRER_{ij})_j^{n}$, corresponding public keys
\((Sen_{ja}, Sen_{jb})\) for all the vehicles and a receiver’s identity \(ID_R\), and its corresponding public key \((PK_{ra}, PK_{rb})\) using the same state information \(\Delta\). In order to verify the signature, this algorithm computes the following.

- \(h_b = H_5(Q_R||PK_{ra}||PK_{rb}||\Delta||T_j||K_j||Q_j||Sen_{ja}||Sen_{jb}), \) for \(j = 1, ..., n\).
- \(h_c = H_4(\Delta)\).

The signature aggregation \(Sig_{agg}\) accept if

\[
\hat{e}(Sig_{agg}, P) = \hat{e}(\sum_{j=1}^{n} sen_{ja} + P_{pub} Q_j, h_c)\hat{e}(\sum_{j=1}^{n} T_i, h_b)\hat{e}(\sum_{j=1}^{n} sen_{jb}, h_c)
\]

If the batch verification holds, the RSU will accept SRERs in list \(V\) as a valid SRERs. Then, the aggregated SRER \(SRER_{agg}\) in \(V\) will be forwarded to complete unsigncryption step. Once a road event report SRER is verified valid, RSU pursues the next unsigncryption step.

### 3.5.4 Data Receiving

The RSU decrypts the SRERs when the signature verification outputs true. The RSU continues to complete the decryption phase and computes the follows.

- \(Z_b' = x_r T_j\).
- \(Z_a' = D_r T_j\).
- \(h_a' = H_2(Q_R||PK_{ra}||PK_{rb}||\Delta||T_j||Z_b'||Z_a')\).
- \(Data_i' = K_j \oplus h_a'\).

### 3.6 Security Analysis

In this section, the security of the proposed protocol has been analyzed according to the security objectives described in section 3.2.
A. The proposed protocol achieves road report $Data_j$ confidentiality and integrity.
The vehicles signcrypt $Data_j$ as $C_j = (T_j, K_j, \alpha_j)$, where $T_j$ and $K_j$ fulfill the encryption part and $\alpha_j$ achieves digital signature in one logical step. Only the RSU unsigncrypts $Data_j'$ by computing $T_j$, $K_j$ and $\alpha_j$. Therefore, according to Definition 4 and Definition 5 the encryption and signature achieve confidentiality and unforgeability under CDH problem.

B. The proposed protocol can achieve the mutual authentication.
RSU is authenticated by the signcryption on the road report $Data_j$ that is generated by the vehicle. Particularly, in the proposed protocol, in order to verify and decrypt the crowdsensing report, only the RSU that holds the private key $(D_R, x_R)$ is able to perform these procedures. The vehicle computes $Z_a$ and $Z_b$ through the signcryption algorithm to establish the mutual authentication. RSU authenticates the source road report by verifying the signcryption on the $Data_j$. Therefore, according to Definition 5 we deduce that the adversary cannot forge the signature on the message without the full private key under DBDH problem in the signcryption unforgeability theorem.

C. The proposed protocol achieves anonymity.
CC computes a pseudo identity $Q_i$ for users in order to fulfill anonymity. This pseudo identity is calculated from the user’s real identity. It is used during the entire road report transmission processes. Therefore, all the malicious parties can not reveal the real identity of the requesting user either vehicle or RSU.

D. The proposed protocol achieves key escrow resilience.
Because the proposed protocol relies on CL-PKC, the CC can only generate the partial private key $D_i$ for the user who is then able to compute the full private key $(D_j, x_j)$ after randomly selecting its secret value $x_j$. Therefore, even the CC is compromised, we insure that the adversary cannot get user’s full private keys.

3.7 Performance Evaluation

In this section, we evaluate the performance of the proposed privacy-preserving protocol in terms of the computational cost and communication overhead. To demonstrate the efficiency of pro-
posed protocol, we compare proposed CLASC scheme with the existing schemes [31], [51], which suffer from computational complexity and communication cost due to the fact that pairing and exponentiation operations take much more computation time.

### 3.7.1 Computational Cost

To the best of our knowledge, we compare the efficiency of our scheme with the certificateless aggregate signcryption scheme available in the literature [31], [51]. As the operations scalar multiplication in $G$, exponentiation in $G_T$ and pairing dominate the computational cost, we consider those three operations in computing the time consumption. We denote $t_p$ the time consumption of pairing, $t_m$ the time consumption of a scalar point multiplication in $G$ and $t_e$ the time consumption of an exponentiation in $G_T$.

In the proposed CLASC scheme, each sender signcrypts the data separately unlike the receiver that is able to aggregate verify all the signature parts of ciphertexts and then unsigncrypt them. The signcryption algorithm takes six multiplication operations in $G$ to compute both signature and encryption. On the other hand, the unsigncrypt algorithm needs four pairing operations and two scalar multiplication operations to aggregate verify the signature and unsigncrypt the ciphertexts.

On the receiver side, verification of signatures can be performed in a single step rather than verifying each signature separately. The computational cost in the receiver side is more efficient than existing schemes. Therefore, efficiency of aggregate signcryption schemes can be evaluated by computing the number of cryptographic operations, including $t_p$, $t_m$ and $t_e$. The comparison of the computational cost among schemes are demonstrated in Table 3.1. While the proposed CLASC in Table 3.1 is implemented without exponentiations, we demonstrate that the existing CLASC schemes have utilized three operations on pairing, multiplication and exponentiation.
Table 3.1: Cryptographic operations comparison with other CLASC schemes

<table>
<thead>
<tr>
<th></th>
<th>Signcrypt</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>schemes</td>
<td>$t_p$</td>
<td>$t_m$</td>
<td>$t_e$</td>
</tr>
<tr>
<td>Lu et al. [51]</td>
<td>1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Ziba et al. [31]</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Proposed</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>UnSigncrypt</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>schemes</td>
<td>$t_p$</td>
<td>$t_m$</td>
<td>$t_e$</td>
</tr>
<tr>
<td>Lu et al. [51]</td>
<td>10</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Ziba et al. [31]</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Proposed</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

In order to evaluate the computation of efficiency of the proposed protocol, an MNT curve [60] with the pairing $\hat{e} : G \times G \rightarrow G_T$ defined over this curve will be employed, where the embedding degree of the curve is 6 and $q$ is 160 bit. The implementation was executed on an Intel Pentium IV 3.0 GHZ machine [74]. The running time is shown in Table 3.2.

Table 3.2: Cryptographic operations running time

<table>
<thead>
<tr>
<th>Operation</th>
<th>Running Time</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_m$</td>
<td>0.6 ms</td>
<td>The time for a scalar point multiplication</td>
</tr>
<tr>
<td>$t_p$</td>
<td>4.5 ms</td>
<td>The time for one pairing operation</td>
</tr>
</tbody>
</table>

Figure 3.3 shows the comparison of computational cost between the existing CLASC schemes and our proposed scheme. It demonstrates that our proposed CLASC scheme needs much fewer computation of time than other CLASC schemes because of the fact that the pairing and exponentiation operations take much longer computation time than the multiplication operation. Our proposed scheme needs four pairing operations while the scheme in [31] has six pairing operations with one exponentiation operation and [51] has eleven pairing operations. Therefore, our
proposed CLASC scheme outperforms the competitive schemes because it has fewer number of pairing operations and does not perform exponentiation operations. Based on the running time results in [74], the computational cost in the whole scheme \( T_k = 8t_m + 4t_p = 8 \times 0.6 + 4 \times 4.5 = 22.8 \) ms. However, there are unique features in the VCS such as mobility. Specifically, the moving vehicle acts as a sender in the assumed application scenario. Consequently, the proposed scheme provides a lightweight signcryption that its time consumption \( T_s = 6t_m = 3.6 \) ms. On the other hand, the receiver RSU, is a fog device that has a high computational capability and the time consumption for unsigncryption \( T_u = 2t_m + 4t_p = 19.2 \) ms, which is an efficient reasonable time consumption including aggregate ciphertexts, batch verification, and unsigncryption.

![Efficiency comparison with other CLASC schemes](image)

### 3.7.2 Communication Overhead

In the proposed CLASC scheme, the communication cost is determined by the size of the aggregated ciphertext length \( SRE_{R_{agg}} \), which is mainly due to report aggregation and batch verification. However, it is not possible to reduce the communication overhead of a CLASC scheme to a constant value because two parts of each ciphertext are needed for decryption. In contrast, the aggregated ciphertext \( SRE_{R_{agg}} \) has \( n + 1 \) elements in \( G \) for achieving the security level. There-
fore, we have an efficient protocol that has much fewer computational time than other schemes, without increasing the communication cost, as shown in Table 3.3. Thus, the proposed protocol is suitable for narrow bandwidth and terminals with limited resources.

Table 3.3: Computational cost and communication overhead analysis

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Computational Cost</th>
<th>Communication Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lu et al. [51]</td>
<td>$11t_p + 7t_m$</td>
<td>$(n + 1)</td>
</tr>
<tr>
<td>Ziba et al. [31]</td>
<td>$6t_p + 5t_m + t_e$</td>
<td>$(n + 1)</td>
</tr>
<tr>
<td>Proposed</td>
<td>$4t_p + 8t_m$</td>
<td>$(n + 1)</td>
</tr>
</tbody>
</table>

3.8 Summary

In this chapter, a novel efficient certificateless aggregate signcryption (CLASC) scheme is proposed. Then, a privacy-preserving protocol in FVCS is designed based on the proposed CLASC scheme. In addition, the proposed privacy-preserving protocol meets the security requirements such as data confidentiality and integrity, mutual authentication, anonymity and key escrow resilience. Extensive comparisons of computational cost and communication overhead show that the proposed scheme can achieve much better efficiency than the existing schemes.

3.9 Limitation

The present study has some limitations within which the findings need to be interpreted carefully. The present study was on vehicular networks that have unique features such as vehicle mobility. This is because of the nature and characteristics of vehicular networks. It is worth pointing out that the proposed protocol uses only a pseudo identity in order to achieve identity privacy. However, this approach may lead to a privacy concern by using one pseudo identity during mobility [73]. Furthermore, the reported locations in the future traffic information from a vehicle can be used to link pseudo identity that may discover a vehicle’s real-world identity [34].
In this work, we do not elaborate vehicle’s pseudo identities change technique due to the fact that this problem is well-studied in the past. In the proposed scheme, we can adopt the mix-zone technique that provides promising ways to avoid this challenge. For instance, when all the vehicles approaching an intersection where there is an RSU deployed, they coordinate with each other and change their pseudo identities at the same time. Also, their public and private keys are updated accordingly with the involvement of CC through the RSU. CC will update the public tree with the vehicles new public keys as well.
Chapter 4

Efficient Compromised Node Revocation in Fog-Assisted Vehicular Crowdsensing

4.1 Introduction

Chapter 3 introduces a secure and efficient privacy-preserving protocol in FVCS. However, a legitimate user may become compromised if detected as a malicious user whether its subscription expires or not. Unfortunately, the proposed privacy-preserving protocol is not suitable for addressing this issue. Therefore, this chapter presents a solution to the problem from a different perspective by allowing a KGC to revoke a detected malicious user from the system.

A comprehensive set of security mechanisms integrated into the VCS applications is critical for their deployment, especially because of the life-critical nature of the vehicular network operation. The security architecture will make use of the KGC or authority to manage the identities, credentials, and cryptographic keys of all network nodes [70]. Therefore, instead of an ad-hoc or web-of-trust method [69], this approach is considered appropriate in the field of VCS.

Generally, security and privacy challenges need to be addressed because road event reports may be disclosed by compromised users. Furthermore, a vehicle could generate false reports for his/her own benefits. For example, giving a notice for traffic jam at a specific location while this location is being clear. Moreover, the KGC cannot be fully trusted since it is usually acted by a
commercial organization. In particular, if the KGC is being compromised, the vehicles private information will be disclosed. In this chapter, we focus our attention to the threat of existing malicious users after detection.

User revocation issue is crucial and should be addressed. The following are some of the serious challenges FVCS applications may encounter. First, each legitimated user either vehicle or RSU has subscription expiry for their private keys during participation. For example, if the vehicle or RSU has expired time for their legal participation and at the same time are detected as a malicious user, they are then considered as a compromised user who may breach the system. One more challenge is from different perspective when a participated user is detected as a malicious user even its private key is still valid. Furthermore, RSUs as a compromised node may generate results for undesired purposes. As a result, the immediate road condition result generated by the malicious RSUs can be accessed by a legitimate vehicle. Therefore, it is logical to provide a privacy-preserving protocol with revocation functionality in order to address the aforementioned issues.

There are also several privacy-preserving designs with revocation functionality in a large-scale environment in the VANETs [98], [68], [92]. The majority of these protocols use CRL method and some utilizes certificate free protocols such as ID-PKC. However, these protocols suffer from high computational cost and time consumption, and key escrow problems. We exploit the advantage of CL-PKC, which solves key escrow problem in ID-PKC. We adopt CL-PKC as the basis of our solution. While there are some designs of revocation approaches for CL-PKC [86], [80], [90], incorporating the revocation functionality on the CLASC scheme may be difficult. Furthermore, the workload of KGC on compromised user revocation becomes a bottleneck once the overhead increases linearly as the number of users in the FVCS applications increases. Thus, designing a privacy-preserving protocol based on CL-PKC with an efficient revocation is a non-trivial task.

In this chapter, we aim to design an efficient revocable privacy-preserving protocol for FVCS in order to successfully address the aforementioned challenges. To the best of our knowledge, this work is considered as the first privacy-preserving scheme that supports efficient revocation mechanism based on the CLASC scheme. In the proposed scheme, a compromised user will be revoked from the system by the KGC to guarantee the system is free from any security threats and privacy risks. To be more specific, the main contributions of this chapter are as follows:
Based on the proposed CLASC scheme [11], a data transmission protocol, which implements compromised user revocation mechanism for enhancing security and privacy in FVCS, is designed [10].

Then, the proposed scheme is based on a binary tree structure in order to achieve an efficient revocation functionality.

The designed scheme assures that the workload of the KGC only increases logarithmically with the number of users.

The proposed scheme has the lowest computational cost and avoids the key escrow problem compared to the counterparts [98], [68], [92].

The rest of this chapter is organized as follows. In Section 4.2, the system models and design goals are presented. In Section 4.3, the proposed revocable privacy-preserving protocol is presented in detail. Security analysis is given in Section 4.4 followed by performance evaluation in Section 4.5. Section 4.6 summarizes the chapter while the limitation of this chapter is discussed in Section 4.7.

4.2 System Models and Design Goals

In this section, we define system model, threat model, and identify the design goals.

A. System Model

As shown in Figure 4.1, the system model comprises of KGC, vehicles, RSUs as fog nodes, and cloud servers. Vehicles should be registered with KGC and preloaded with the public parameters before it generates road event reports. For safety reasons, a vehicle must be revoked from the system in case KGC detects the misbehavior of that vehicle by declaring it as compromised. RSUs have the ability to react and make decisions close to the end users. RSUs should also be registered with KGC and preloaded with public parameters before they provide results. The revocation of the RSUs is necessary as well in case the RSUs violate system rules. KGC is in charge of the enrollment and eviction of vehicles and RSUs.
Moreover, in order to renew the private keys for the users (e.g., vehicle, RSU), KGC first checks if that user is discovered as malicious or not. Thus, KGC can take the decision to revoke that user or just broadcast the time key update for him to resume the system. Cloud servers are the data centers of the system. The system data such as historic information is stored in the cloud to be utilized later.

![System model](image)

**Figure 4.1: System model**

### B. Threat Model

In this threat model, we focus our attention to the threat of legitimate users (e.g., vehicles or RSUs) in case they become compromised. As we assumed in the previous chapter, RSUs are considered as fully-trusted nodes but they may be detected as a misbehaved user or their private keys are compromised. In particular, any legitimate user in the system has a certain time period for its private key to participate. If the private keys for legal users have expired and detected as malicious users, they then become compromised users and thus can tamper the system. One of the threat challenges, a compromised vehicle can still use the keys
and thus fabricate the crowdsensing reports for his own purposes. As a result, this compromised vehicle can send bogus information to the RSU for processing. Furthermore, a compromised RSU is also capable to decrypt crowdsensing reports generated by legal vehicles. Consequently, it can disclose the privacy of vehicles and can create false results for certain benefits. Finally, as the KGC may be represented by a commercial organization, it can not be fully trusted. By considering the commercial benefits, it is natural for KGC to misbehave such as illegally collecting and accessing vehicles sensitive information. Therefore, in order to protect the system from any these challenges, the following security goals should be achieved.

C. Design Goals

The design goals are to propose an efficient revocable privacy-preserving protocol in FVCS. The proposed protocol combines an efficient revocation mechanism based on binary tree structure with the CLASC scheme to address the possible above threats. Specifically, the proposed protocol should achieve the following design goals.

- **Revocation functionality.** The revocation mechanism takes place against any user whose is detected as malicious and his subscription may be expired. Thus, it is considered as a compromised user. Therefore, the KGC should revoke the compromised users from the system. in addition, when the detected malicious user may still have valid time period for participation, KGC can also revoke him by distributing key update keys for renew the subscription of legal users except the malicious one. As a result, achieving this significant goal leads to fulfilling the following design goals.

- **Data confidentiality and integrity.** Compromised vehicles cannot modify the crowdsensing reports that should be protected from any tampering. On the other hand, a compromised RSU can not modify the result for his own purposes. Furthermore, compromised RSUs are not able to decrypt the crowdsensing reports and obtain the data from the legitimate vehicles.

- **Key escrow resilience.** The KGC only generates partial private keys for the registers to avoid key escrow problem. As a result, it is blocked to access the vehicles private information.
4.3 Construction of the proposed protocol

In this section, we initially present the basic concepts and the necessary complexity assumptions of the proposed protocol. Then, we define a KUNodes algorithm [16] which is used to achieve the efficient revocation function, e.g., reducing the key update costs. Furthermore, we propose an efficient privacy-preserving scheme for compromised node revocation in FVCS.

4.3.1 Preliminaries

I. Bilinear Maps.

Let $G$ be a cyclic multiplicative group with the generator $g$ of large prime order $q$, and $G_T$ be another cyclic multiplicative group of the same large prime order $q$. An admissible bilinear pairing $\hat{e} : G \times G \rightarrow G_T$ is a map with the following properties.

- Bilinearity. For all $u, u', v, v' \in G$, $\hat{e}(uu', v) = \hat{e}(u, v)\hat{e}(u', v)$ and $\hat{e}(u, vv') = \hat{e}(u, v)\hat{e}(u, v')$.
- Non-degeneracy. There exist $g, g_1 \in G$ such that $\hat{e}(g, g_1) \neq 1_{G_T}$.
- Computability. For all $g, g_1 \in G$, there is an efficient algorithm to compute $\hat{e}(g, g_1)$.

An admissible bilinear pairing $\hat{e} : G \times G \rightarrow G_T$ can be implemented by the modified Weil/Tate pairings over elliptic curves [17].

II. Complexity Assumptions.

We recall the following intractability assumptions related to the security of our scheme.

Definition 1 (Computational Diffie-Hellman (CDH) Problem): Given a tuple $(g, g^a, g^b) \in G$ for some $a, b \in \mathbb{Z}_q^*$, the CDH problem in $G$ is to compute the element $g^{ab}$.

Definition 2 (Decisional Bilinear Diffie-Hellman (DBDH): Given a tuple $(g, g^a, g^b, g^c) \in G$ for some $a, b, c \in \mathbb{Z}_q^*$ and $x \in G_T$, the DBDH problem is to decide whether $x$ holds $\hat{e}(g, g)^{abc}$ or not.
4.3.2 KUNode Algorithm Definition

In order to overcome scalability concerns, we define the KUNode algorithm that applies a binary tree structure, where each leaf node $\eta_i$ represents a user $u_i$. This algorithm is used to compute the minimal set of nodes when a compromised node must be revoked. It generates key updates for these nodes. As a result, only non-revoked vehicles at time $T$ are able to signcrypt crowdsensing reports and non-revoked RSUs can unsigncrypt ciphertexts. Basically, this algorithm takes as input a binary tree $BT$, revocation list $RL$, and time $T$, and outputs a set of nodes for the key update. If a compromised user $\eta_i$ is revoked within the time period $T_i$, then the item $(\eta_i, T_i)$ is inserted into the $RL$, $(\eta_i, T_i) \in RL$. $\text{Path}(\eta_i)$ denotes the set of nodes on the Path inclusively from the leaf node $\eta_i$ to the root node $R$.

![Figure 4.2: An example of compromised node revocation in KUNodes algorithm](image)

Figure 4.2: An example of compromised node revocation in KUNodes algorithm
This algorithm works as follows: In case no user has been detected as malicious, the root node will be returned by the KUNode algorithm. On the other hand, as shown in Figure 4.2, if a user $u_6$, who is assigned to the leaf node $x_{13}$, has been compromised, the set of nodes $\text{Path}(x_{13}) = \{x_{13}, x_6, x_3, R\}$ will be inserted into the set $X$ that has the ancestors of the revoked user. The KUNode algorithm then selects sibling nodes according to $\text{Path}(x_{13})$. For example, choosing $x_{12}$ which is sibling with revoked node $x_{13}$, $x_7$ through its revoked sibling $x_6$ and $x_2$ that is sibling with revoked node $x_3$. Thus, the KUNode algorithm returns a new set $Y = \{x_{12}, x_7, x_2\}$ that does not contain any ancestors of revoked users. Consequently, all users except the revoked user $u_6$, have a node $y \in Y$ that is contained in the set of nodes on the path from their assigned node to the root node. Therefore, $u_1, u_2, u_3$ and $u_4$ have the node $x_2 \in Y$, $u_5$ has the node $x_{12} \in Y$, and $u_7$ and $u_8$ have the node $x_7 \in Y$, whereas $Y \cap \text{Path}(x_{13}) = \emptyset$ as illustrated in Figure 4.2.

When a user, for instance an RSU or a vehicle, joins the system, KGC first assigns a random leaf node $\eta_i$ of a complete $BT$ to this user, and then issues a set of initial partial private keys, i.e., identity component of partial private key to this user, where each key in this key set corresponds to each node on Path. To get rid of revoked users, the KGC distributes the key updates i.e., time component of partial private key for a set $\text{KUNode}(BT, RL, T)$ at time period $T$. Then, only non-revoked users have at least one key corresponding to a node in $\text{KUNode}(BT, RL, T)$ and are able to generate the full partial private key.

The advantage of adopting KUNode algorithm is to reduce the size of key update costs from linear $O(n)$ to logarithmic $O(\log_2 n)$ when the number of users increases. Particularly, instead of KGC requires to compute key updates for each node separately, it just needs to broadcast them to a set of nodes (i.e., $x_2, x_7, x_{12}$) at time period. Evidently, it is computationally better in terms of efficient revocation due to efficiency improvements at the stage of key updates. By exploiting this advantage, we integrate KUNode algorithm with the CLASC scheme in order to design an efficient revocable privacy-preserving protocol.

### 4.3.3 Proposed revocable privacy-preserving scheme

We present the details of our revocable privacy-preserving protocol, which consists of six phases: system initialization, time update key, data formulation and sending, secure road event report
(SRER) aggregated verification, data receiving and user revocation.

1. **System Initialization**

The KGC chooses a security parameter $k$, which ensures the security level of the system and determines the prime order $q$ of the bilinear groups. In general, $k = 256$ or $160$ bits. In this phase, the vehicles and RSUs register with the KGC to generate their public and full private keys. The KGC first selects a cyclic multiplicative group $G$ with the generator $g$ of large prime order $q$, and $G_T$ be another cyclic multiplicative group of the same order $q$ and a bilinear map $\hat{e}: G \times G \rightarrow G_T$. Then, the KGC chooses a random master secret key $s \in \mathbb{Z}_q^*$ and a random arbitrary generator $g \in G$. It also computes its master public key $P_{pub} = g^s$ with picking a random element $g_1 \in G$. Additionally, the KGC chooses three secure hash functions: $H_1: \{0,1\}^* \rightarrow G$, $H_2: \mathbb{Z}_q^* \rightarrow G$ and $H_3: G \times G_T \rightarrow \{0,1\}^*$. The KGC sets an initially empty revocation list $RL = \emptyset$ and, a state $st = BT$ where $BT$ denotes to a $BT$ with $N$ leaves. The KGC then publishes the system parameter $\text{params} = (G, G_T, e, g, g_1, q, P_{pub}, H_1, H_2, H_3, RL, st)$ and the master secret key $s$ will be kept secure by the KGC. The KGC also determines the format of $SRER_{ij}$. For a road event $RE_i$, the vehicles $Sen_j$ will generate the data where $Data_j = (Time_{ij}, Location_{i}, Signals_i)$ and the $SRER_{ij}$ will securely forward to the RSU in the format $SRER_{ij} = (Q_j, \text{Signcrypt}(Data_j))$. Also, a binary tree will be constructed like the one shown in Figure 4.2, where leaf nodes (users) are vehicles or RSUs.

Vehicles and RSUs can join the system by performing the following steps:

- Before RSUs provide the results to the vehicles, A vehicle with identity $Sen_j$ randomly selects $sID \in \mathbb{Z}_q^*$ as its secret value and computes its public key $Sen_{PK} = g^{sID}$.
- The KGC receives $Sen_j$’s identity and public key $(Sen_j, Sen_{PK})$ for registration. In order to keep identity private, KGC computes $Q_j = H_1(sen_j)$ as its pseudo identity.
- Then, the KGC computes the initial partial private key (identity component of partial private key) by performing the following:
– When a \( Sen_j \) provides its identity \( Q_j \), KGC first checks whether the identity \( Q_j \) exists in the \( RL \) or not. IF this \( Q_j \) has already been revoked, the KGC aborts initial partial private algorithm.

– Otherwise, KGC then stores the identity \( Q_j \) in the leaf node \( \eta_{Q_j} \), which is randomly chosen from the \( BT \).

– For each node \( \theta \in \text{Path}(\eta_{Q_j}) \), KGC selects \( s_{\theta_1}^{Q_j} \in Z_q^* \) and stores \( (s_{\theta_1}^{Q_j}, s_{\theta_2}^{Q_j}) \) in the node \( \theta \) that satisfies \( s_{\theta_1}^{Q_j} + s_{\theta_2}^{Q_j} = s \mod q \) in case this node has not been initialized yet.

– If the node \( \theta \) is already initialized, KGC retrieves \( s_{\theta_1}^{Q_j} \) from node \( \theta \) and selects \( z_{\theta}^{Q_j} \in Z_q^* \).

– KGC computes \( (u_{\theta_1}^{Q_j}, u_{\theta_2}^{Q_j}) = (g_{\theta_1}^{s_{\theta_1}^{Q_j}} \cdot H_1(Sen_j)^{Q_j}, g^{z_{\theta}^{Q_j}}) \).

– KGC returns the initial partial private key \( DID = \{(u_{\theta_1}^{Q_j}, u_{\theta_2}^{Q_j})\}_{\theta \in \text{Path}(\eta_{Q_j})} \) to the corresponding user via a secure channel.

– Likewise, RSU with the identity \( ID_{RSU} \) is provided the initial partial private key \( DID_{RSU} = \{(u_{\theta_1}^{ID_{RSU}}, u_{\theta_2}^{ID_{RSU}})\}_{\theta \in \text{Path}(\eta_{RSU})} \) and its public key with secret value \( (PK_{RSU}, s_{RSU}) \).

2. Time Update Key

This algorithm first finds a minimal set of nodes, which contains no ancestors of revoked users. It then computes the t-component of partial private key. For each node \( \theta \in \text{KUNode}(BT, RL, T) \), KGC generates the time update key and broadcasts it to all the non-revoked users (NRU) using the following steps:

– KGC recalls \( s_{\theta_2}^{Q_j} \) from the node \( \theta \), and chooses \( r_{\theta}^{Q_j} \in Z_q^* \).

– KGC computes the time update key \( (w_{\theta_1}^{Q_j}, w_{\theta_2}^{Q_j}) = (g_{\theta_1}^{s_{\theta_2}^{Q_j}} \cdot H_2(T_i)^{Q_j}, g^{r_{\theta}^{Q_j}}) \), and sends it to the corresponding NRU.

– KGC returns the time update key \( DID_t = \{(w_{\theta_1}^{Q_j}, w_{\theta_2}^{Q_j})\}_{\theta \in \text{KUNode}(BT, RL, T)} \) to the corresponding NRU.

– Once a non-revoked vehicle with identity \( Q_j \) receives the time update key \( DID_t \), it can generate the full partial private key, by finding components of \( DID \) and \( DID_t \),
which were computed on the same node. In case $KUNode(BT, RL, T) \cap \text{Path}(\eta_{Q_j}) = \emptyset$, the generating of full partial private key algorithm will abort. Otherwise, a vehicle performs the follows:

- Selects $\theta \in KUNode(BT, RL, T)$.
- Chooses $z_{Q_j}, r_{Q_j} \in \mathbb{Z}_q^*$.  
- Computes the following components to calculate the full partial private key.

$$DID^1_{Q, t} = (u_{\theta_1}^{Q_j} \cdot w_{\theta_1}^{Q_j} \cdot H_1(Sen_j)^{z_{Q_j}} \cdot H_2(T_i)^{r_{Q_j}}).$$  
$$DID^2_{Q, t} = (u_{\theta_2}^{Q_j} \cdot g^{z_{Q_j}}).$$  
$$DID^3_{Q, t} = (w_{\theta_2}^{Q_j} \cdot g^{r_{Q_j}}).$$

- A vehicle with identity $Q_j$ returns the full partial private key.

$$PSK_{Q_j, T} = (DID^1_{Q, t}, DID^2_{Q, t}, DID^3_{Q, t})_{\theta \in \text{Path}(\eta_{Q_j})}.$$  

- Likewise, RSU with the identity $ID_{RSU}$ is provided the full partial private key.

$$PSK_{RSU, T} = (DID^1_{RSU, t}, DID^2_{RSU, t}, DID^3_{RSU, t})_{\theta \in \text{Path}(\eta_{RSU})}.$$  

- Finally, a vehicle with identity $Q_j$ computes the full private key $SK_{Q_j, t}$, which is expressed as $(DID^1_{Q, t}, DID^2_{Q, t}, DID^3_{Q, t}, sID)$. Likewise, RSU is provided $SK_{RSU, t} = DID^1_{RSU, t}, DID^2_{RSU, t}, DID^3_{RSU, t}, s_{RSU}).$

3. Data Formulation and Sending

This part is performed by a non-revoked vehicle $Q_j$. A road event $RE_i$ is sensed by one or multiple vehicles and then $Data_j$ is discovered. After that, $Q_j$ with encrypted $Data_j$ as a $SRER_{ij}$ sends to the RSU as fog device receiver. Then, the vehicles, associated with the identity $Q_j$ and its public key $Sen_{PK}$ within the time $T_i$, signcrypts $Data_j$ in time period $T_i$ with a receiver $ID_{RSU}$. $Q_j$ performs the following steps:

- Chooses a random value $r \in \mathbb{Z}_q^*$ and computes,

$$U = g^r.$$  
$$K = \hat{e}(P_{pub}, g_1)^r.$$  
$$V = (PK_{RSU})^r.$$
• $CID = H_1(ID_{RSU})^r$.
• $CIDt = H_1(sen_j)^r$.
• $TID = H_2(T_i)^r$.
• $C_{ENC} = H_3(K,V) \oplus data_j$.
• $Z = H_1(C_{ENC}, ID_{RSU}, Q_j, T_i, PK_{RSU}, Sen_PK, U)^{sID}$.
• $C_{SIGN1} = (DID_{Q_jt}^1 \cdot CIDt \cdot TID \cdot Z)$.
• $C_{SIGN2} = (DID_{Q_jt}^2 \cdot U)$.
• $C_{SIGN3} = (DID_{Q_jt}^3 \cdot U)$.
• $C_{SIGN} = (C_{SIGN1}, C_{SIGN2}, C_{SIGN3})$.
• Returns $C = (U, C_{ENC}, C_{SIGN}, CID, TID)$.

The ciphertext $C = (U, C_{ENC}, C_{SIGN}, CID, TID)$ is attached to secure road event report in the format as $SRER_{ij} = (Q_j, \text{Signcrypt}(Data_j))$, where $\text{Signcrypt}(Data_j) = C$.

It is worth pointing out that we can adopt the mix-zone technique [34] to avoid the problem when vehicle uses only one pseudo identity during mobility, which can be easily predicted and inadequate to preserve driver privacy [73].

4. SRER Aggregated Verification

The RSU performs the SRER aggregation and SRER batch verification operations as follows:

i. SRER Aggregation

Aggregate SRER is used to aggregate multiple SRERs into a single SRER. For a road event $RE_i$, given $n$ SRERs $SRER_{ij} = (Q_j, \text{Signcrypt}(Data_j))$ by vehicles $Sen_1, ..., Sen_n$, we can obtain,

$$SRER_{agg} = (Q_1...Q_n, \text{Signcrypt}(Data_j)_1^j...\text{Signcrypt}(Data_j)_n^n).$$

This algorithm takes a collection of individual ciphertexts,

$$C = (U, C_{ENC}, C_{SIGN}, CID, TID)_{j=1}^n$$
generated by vehicles with \((Q_j)_{j=1}^n\) to a receiver with identity \(ID_{RSU}\). Then, we have aggregated the signature parts of ciphertexts and an aggregate signcryption generator computes the signature aggregation \(\text{sig}_{agg} = \sum_{j=1}^{n} C_{SIGN}\). Finally, it outputs the aggregate ciphertexts \(SRER_{agg}\)

\[
= ((Q_j)_{j=1}^n, U_1...U_n, C_{ENC_1}...C_{ENC_n}, C_{ID_1}...C_{ID_n}, T_{ID_1}...T_{ID_n}, \text{sig}_{agg}).
\]

ii. SRER Batch Verification

This step performs signature batch verification for all the ciphertexts simultaneously. Given the signature aggregation \(\text{sig}_{agg}\), the report sets \((SRER_{ij})_{j=1}^n\), corresponding public keys \((Sen_{PK})_{j=1}^n\) for all vehicles and a receiver’s identity \(ID_{RSU}\), and its corresponding public key \((PK_{RSU})\). In order to verify the signature, this algorithm computes \(Z = H_1(C_{ENC}, ID_{RSU}, Q_j, T_i, PK_{RSU}, Sen_{PK}, U)^{sID}\), for \(j = 1, ..., n\).

The signature aggregation \(Sig_{agg}\) accept if \(\hat{e}(C_{SIGN_1}, g)\)

\[
\hat{e}(P_{pub}, g_1) \prod_{j=1}^{n} \hat{e}(H_1(sen_j), C_{SIGN2}) \prod_{j=1}^{n} \hat{e}(H_2(T_i), C_{SIGN3}) \prod_{j=1}^{n} \hat{e}(Z, Sen_{PK})
\]

5. Data Receiving

Once RSUs receive \(SRER_{agg}\) from vehicles within the time period \(T_i\) and signature verification outputs true, RSU performs the following operations to complete the decryption phase.

- \(K' = \hat{e}(U, DID_{RSU}) \cdot \hat{e}(CID, DID_{RSU}^2)^{-1} \cdot \hat{e}(TID, DID_{RSU}^3)^{-1}\).
- \(V' = (U)^{sRSU}\).
- \(data_j = H_3(K', V') \oplus C_{ENC} \).
- Then, returns the report \(\{data_j\}_{j=1}^n\)
The correctness of our decryption scheme is as follows

\[ K' = \hat{e}(U, DID_{RSU}^1) \cdot \hat{e}(CID, DID_{RSU}^2)^{-1} \cdot \hat{e}(TID, DID_{RSU}^3)^{-1} \]
\[ = \hat{e}(g^r, g_1^s) \cdot \hat{e}(g^r, H_1(ID_{RSU})^s_{RSU} + z_{RSU} \cdot \hat{e}(g^r, H_2(T_r)_{RSU}^r \cdot \hat{e}(H_1(ID_{RSU})^s_{RSU} + z_{RSU}, g^r)^{-1} \]
\[ \cdot \hat{e}(H_2(T)^r_{RSU} + r_{RSU}, g^r)^{-1} \]
\[ = \hat{e}(g^s, g_1^s) \]
\[ = \hat{e}(P_{pub}, g_1^r) \]

\[ K' = K \]

\[ V' = (U)^{s_{RSU}} \]
\[ = (g^r)^{s_{RSU}} \]
\[ = (g^{s_{RSU}})^r \]
\[ = (PK_{RSU})^r \]

\[ V' = V \]

\[ m' = H_3(K', V') \oplus C_{ENC} \]
\[ = H_3(K', V') \oplus H_3(K, V) \oplus m \]

\[ m' = m \]

The correctness of our signature scheme is as follows.

\[ \hat{e}(C_{SIGN1}, g) = \prod_{j=1}^{n} \hat{e}(DID_{Q_i}, CID \cdot TID \cdot Z, g) \]
\[ = \prod_{j=1}^{n} \hat{e}(g_1^s, H(s_{sen})^q_{Q_j} + z_{Q_j}^r \cdot H_2(T_r)^q_{Q_j} + r_{Q_j}^r \cdot Z^s_{ID}, g) \]
\[ = \prod_{j=1}^{n} \hat{e}(g_1^s, g) \cdot \prod_{j=1}^{n} \hat{e}(H_1(s_{sen})^q_{Q_j} + z_{Q_j}^r, g) \cdot \prod_{j=1}^{n} \hat{e}(H_2(T_r)^q_{Q_j} + r_{Q_j}^r, g) \cdot \prod_{j=1}^{n} \hat{e}(Z^s_{ID}, g) \]
\[
\prod_{j=1}^{n} \hat{e}(g, g^s) \cdot \prod_{j=1}^{n} \hat{e}(H_1(senj), g^{z_{Q_j} + z_{Q_j} + r}) \cdot \prod_{j=1}^{n} \hat{e}(H_2(T_i), g^{r_{Q_j} + r_{Q_j} + r}) \cdot \prod_{j=1}^{n} \hat{e}(Z, g^{sID})
\]

\[
= \prod_{j=1}^{n} \hat{e}(P_{pub}, g_1) \cdot \prod_{j=1}^{n} \hat{e}(H_1(senj), C_{SIGN2}) \cdot \prod_{j=1}^{n} \hat{e}(H_2(T_i), C_{SIGN3}) \cdot \prod_{j=1}^{n} \hat{e}(Z, Sen_{PK})
\]

6. Revocation

If the private key of the user either a vehicle or RSU has been compromised or the attacker has been detected, the leaf node \( \eta_{Q_j} \) or \( \eta_{RSU} \), which is linked with identity \( Q_j \) or \( ID_{RSU} \) along with the revocation time period \( T_i \), is returned to the KGC who then updates the revocation list by \( RL \rightarrow RL \cup \text{neither} \ (Q_j, T_i) \) or \( (ID_{RSU}, T_i) \).

4.4 Security Analysis

In this section, we analyze how the proposed protocol achieves the design goals described in Section 4.2.

- **Revocation functionality.** The proposed protocol achieves revocation mechanism and efficiently expels misbehaving users from the system. Specifically, if the compromised vehicle \( Q_j \) or RSU \( ID_{RSU} \) has been revoked in the time period \( T_i \), the revocation function can be easily carried out by updating the revocation list \( RL \rightarrow RL \cup \text{neither} \ (Q_j, T_i) \) or \( (ID_{RSU}, T_i) \). The KGC then publishes the time update key \( DID \) for the set \( \text{KUNode}(BT, RL, T) \). Therefore, only the unrevoked users can get the updated key according to the complete \( BT \) algorithm [16]. Thus, the legitimate vehicles can signcrypt the reports while the legitimate RSUs have the ability to unsigncrypt ciphertexts. Satisfying compromised node revocation property helps achieving the following significant requirements.

- **Data confidentiality and integrity.** The proposed protocol guarantees the confidentiality and integrity of the crowdsensing reports. The reports are signed and encrypted under CDH problem. The encryption and signature achieve confidentiality and unforgeability. As a result, revoking the compromised RSU guarantees that it cannot
decrypt any report and disclose vehicles privacy. Furthermore, the compromised vehicle cannot modify the report in case it is successfully revoked from the system. In particular, the vehicle signcrypts $Data_j$ as $C = (U, C_{ENC}, C_{SIGN}, CID, TID)$, where $U$, $C_{ENC}$, $CID$, $TID$ fulfill the encryption part and $C_{SIGN}$ achieves digital signature in one logical step. Only the unrevoked RSU is able to unsigncrypt $Data_j$ because of the knowledge of its initial partial private key, user secret key and the updated key issued by the KGC.

- **Key escrow resilience.** The key escrow problem can be solved in the proposed protocol such that the compromised KGC cannot neither disclose the users sensitive information or impersonate them.

## 4.5 Performance Evaluation

In this section, we evaluate the performance of the proposed protocol in terms of computational cost, communication overhead and KGC overhead.

### 4.5.1 Computational Cost

Currently, there are no scalable and revocable privacy-preserving protocols proposed in VCS based on CLASC. Therefore, we have investigated some of the existing privacy-preserving protocols that are suitable for large-scale application in VANET. We compare our proposed protocol with counterparts [98], [68], [92] in terms of computational cost and communication overhead. As the operations exponentiation in $G$ and $G_T$, and pairing dominate the computational cost of the algorithms, we consider these three operations in the time consumption. We denote $t_p$ the time consumption of pairing, $t_1$ the time consumption of exponentiation in $G$, $t_2$ the time consumption of exponentiation in $G_T$.

The proposed protocol utilizes the signcryption technique, which achieves efficiency in terms of computational cost and communication overhead. The proposed protocol is more efficient than the existing protocols in terms of data generation, receiving and batch verification. The comparison of the computational cost among protocols are demonstrated in Table 4.1.
Table 4.1: Data sending, receiving and batch verification comparison of different privacy-preserving protocols

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Data Generation/Receiving</th>
<th>Batch Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang [98]</td>
<td>$22t_1 + 3t_2 + 5t_p$</td>
<td>$13nt_1 + 2nt_p$</td>
</tr>
<tr>
<td>Qin [68]</td>
<td>$30t_1 + 12t_2 + 22t_p$</td>
<td>$16nt_1 + 7nt_2 + 13nt_p$</td>
</tr>
<tr>
<td>Wang [92]</td>
<td>$17t_1 + 7t_2 + 8t_p$</td>
<td>$13nt_1 + 2nt_p$</td>
</tr>
<tr>
<td>Our protocol</td>
<td>$5t_1 + t_2 + t_p$</td>
<td>$nt_1 + 5nt_p$</td>
</tr>
</tbody>
</table>

To evaluate the computational efficiency of the proposed protocol and quantify the running time of the operations, we use the results in [81] as the benchmark for comparisons. To quantify the running time of the cryptographic operations in the proposed scheme, the pairing-based cryptography (PBC) library *ver.0.4.18* [52] is used and the implementation of the algorithms was executed on a 3.2 Ghz Pentium IV machine [81]. The running time are $t_1 = 6.4$ ms, $t_2 = 0.6$ ms and $t_p = 5.9$ ms. Figure 4.3 shows the comparison of time consumption between existing protocols and our proposed protocol. It demonstrates that the proposed protocol requires less time compared to other protocols.

![Figure 4.3: Time consumption comparison](image)
It is fair to point out that the protocols [98], [68], [92] provide a significant security feature in VANET, i.e., a group signature. However, in our application scenario every vehicle only needs to generate its road condition report. Hence, due to application requirement, we consider every vehicle only signcrypts its report separately once road condition takes place.

### 4.5.2 Communication Overhead

We denote $|G|$, $|G_T|$ and $|m|$ the size of the elements in $G$, $G_T$ and the message respectively. We consider the communication overhead at the size of the ciphertext. The ciphertext $C$ is composed of $U, C_{ENC}, C_{SIGN}, CID, TID$. Thus, the size of the ciphertext is $4|G| + |m|$. It is not possible to reduce the communication overhead of the proposed protocol to a constant value because four parts of each ciphertext are needed for decryption. Therefore, $C$ has four elements in $G$ for achieving the security level. From Table 4.2, we can observe that the communication overhead of our protocol is slightly more than the other protocols [98], [92]. This degradation is forgivable considering the fact that the proposed protocol has the lowest computational cost. It also achieved signature and encryption in one logical step along with key escrow resilience.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Computational Cost</th>
<th>Communication Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang [98]</td>
<td>$22t_1 + 3t_2 + 5t_p$</td>
<td>$</td>
</tr>
<tr>
<td>Qin [68]</td>
<td>$30t_1 + 12t_2 + 22t_p$</td>
<td>$4</td>
</tr>
<tr>
<td>Wang [92]</td>
<td>$17t_1 + 7t_2 + 8t_p$</td>
<td>$</td>
</tr>
<tr>
<td>Our protocol</td>
<td>$7t_1 + t_2 + 9t_p$</td>
<td>$4</td>
</tr>
</tbody>
</table>

### 4.5.3 KGC Overhead

We take the KGC overhead into account in computing and issuing key updates. In particular, KGC broadcasts key update $DID_t$ in each time period $T$; therefore, by using a binary tree data structure, the size of $DID_t$ can be much smaller and logarithmically increased in the number
of users $n$. To be more precise, KGC just needs to send key updates $DID_t$ to certain ancestor nodes who link to the unrevoked leaf nodes. For example, KGC broadcasts key updates to a set of nodes (i.e., $x_2, x_7, x_{12}$) at time period rather than of sending each leaf node separately (i.e., $x_8, x_9, x_{10}, x_{11}, x_{12}, x_{14}, x_{15}$) as illustrated in Figure 4.2. Consequently, the proposed protocol improves KGC’s efficiency at the stage of distributing key updates and thus achieves efficient revocation.

### 4.6 Summary

In this chapter, we propose an efficient revocable privacy-preserving protocol for enhancing security in FVCS. Furthermore, user revocation and batch verification are provided in the proposed protocol. Also, the proposed protocol fulfills the security requirements such as confidentiality, integrity and key escrow resilience. Through extensive performance evaluation, we have demonstrated that the proposed protocol can fulfill much better efficiency than counterparts.

### 4.7 Limitation

Most security and privacy schemes have limitations and ours was no exception. The proposed system is not intended to be used for malicious node detection. Since malicious node detection is a well-studied problem in the literature [65], [37], we do not discuss the design of the detection system itself. Therefore, the major concentration is how to provide a promising and efficient mechanism to help achieve a significant security objective, which is compromised node revocation with respect to CL-PKC. In this work, we assume that the malicious user has already been detected by the system (e.g., KGC). Then, the proposed protocol has tackled this security and privacy issue by revoking the existed malicious user from the system.
Chapter 5

Efficient Deduplicated Reporting in Fog-Assisted Vehicular Crowdsensing

5.1 Introduction

The previous chapters (e.g. Chapters 3 and 4) introduced a security architecture, including secure and efficient privacy-preserving protocols in FVCS. The ultimate goal of an FVCS framework is to provide a secure and promising environment to apply safety/non-safety related applications to improve road safety and facilitate traffic management for the community. Nevertheless, generating crowdsensing reports at the same location may result in some duplication in these reports leading to vast communication bandwidth and storage resources for RSUs. Furthermore, handling this concern with a semi-trusted node brings another security and privacy challenge. This chapter addresses these challenges from a different perspective by allowing semi-trusted RSUs to perform secure computations by finding duplicated data on crowdsensing reports before RSUs provide them to an organization.

Since modern vehicles are equipped with mobile sensors and OBUs, vehicles are able to provide driving information and report on road conditions such as potholes, bumps, and slipperiness. Thus, the deployment of distributed fog nodes improves the accuracy and efficiency of road quality and safety since the fog nodes are located at the edge of the network and close
to the vehicles. The goal of these vehicles is to share their resources based on task specification requirements such as sensing areas (e.g., downtown or school areas). As an example, when organizations (such as municipalities or insurance companies) need to know information about possible traffic congestion in the downtown area, they ask RSUs as fog nodes [91] to recruit the relevant vehicles that are on duty to assess.

Specifically, vehicles need to collect data from a required location and then outsource the crowdsensing reports to an organization. However, some duplication in these crowdsensing reports is bound to occur [63]. This challenge results in massive communication bandwidth and storage resources for RSUs. As a result, RSUs provide a promising approach by discarding redundant copies and only need one copy from every set of identical reports. For instance, suppose that there are a specific number of vehicles generating the same crowdsensing reports from the downtown area, as illustrated in Figure 5.1. In this scenario, the RSU should detect the replicated reports generated from the same location in order to reduce the burden on its computations and storage. Furthermore, this approach helps to improve efficiency by only having one report from a collection of identical reports in the same position. This approach also saves bandwidth cost. However, selecting one copy from the set of reduplicate reports and deleting the rest does not guarantee fairness among participating vehicles. Therefore, the contributions of the participating vehicles should be taken into account.

Nevertheless, RSUs are considered as semi-trusted nodes which may be curious about the contents of crowdsensing reports. Detecting repeated reports by RSUs may incur potential privacy disclosure. Thus, participating vehicles have definite concerns about their private information. In order to ensure that they are free from security risks and privacy threats, we utilize a promising cryptographic signcryption mechanism to prevent private information of the participating vehicles from being disclosed, and fulfill data confidentiality, integrity and authentication in one logical step. In our previous work [11], we designed a CLASC scheme to achieve security requirements such as confidentiality, integrity, mutual authentication, anonymity and key escrow resilience. However, this solution brings a critical challenge to the RSUs in detecting reduplicate ciphertext. As a result, vehicles attempt to outsource encrypted crowdsensing reports while supporting deduplication techniques such as convergent encryption (CE) [26], the concept of which we adopt in this work.

In this chapter, we propose a novel efficient deduplicated reporting scheme for FVCS in
order to successfully address the aforementioned challenges. A thorough search of the relevant literature suggests that the proposed scheme is the first privacy-preserving scheme that supports secure data deduplication mechanism based on the CLASC scheme, while no existing work considers secure deduplication based on CL-PKC. The proposed scheme is resilient to malicious semi-trusted RSUs, finds repeated reports and preserves the privacy at the same time. The main contributions are as follows:

- We propose a privacy-preserving data deduplication scheme in order to improve security and privacy for FVCS [9].

- We design a secure deduplication mechanism that is capable to perform safe computation and find replicated reports via a semi trusted RSU.

- We integrate a homomorphic concept with a CLASC scheme [11] in order to preserve the crowdsensing reports from being disclosed. Specifically, RSUs are able to select one copy from a set of detected signencrypted reduplicated reports and delete the rest without accessing any information about the reports.
The proposed scheme also achieves fairness between vehicles whose reports are reduplicated and deleted.

The proposed scheme demonstrates secure data deduplication property and efficiency in terms of computational cost, communication overhead and bandwidth overhead.

The remainder of this chapter is organized as follows. In Section 5.2, the system models and design goals are presented followed by the preliminaries in Section 5.3. In Section 5.4, the proposed deduplicated reporting scheme is presented in detail. Security analysis is given in Section 5.5. Performance evaluation is analyzed in Section 5.6. A summary of the chapter is presented in Section 5.7 while the limitation of this chapter is discussed in Section 5.8.

5.2 System Models and Design Goals

In this section, we present the system model, threat model and design goals.

A. System Model

As shown in Figure 5.2, the system model consists of key generation center (KGC), organizations, crowdsourcing server (CS), RSUs as fog nodes and vehicles.

- Organizations release their spatial tasks to CS in order to collect crowdsensing reports. Based on the task information, CS then assigns these tasks to the fog nodes (RSUs), who are deployed at the edge of the network and close to the users.

- RSUs are powered with computational capabilities and storage spaces for providing computation and storage services. They are in charge to recruit a set of vehicles to perform spatial tasks. They also perform data deduplication on the crowdsensing reports submitted by vehicles and forward deduplicated reports to the CS and organizations. In this system model, the RSU is considered as a semi-trusted component.

- Vehicles utilize their own on-board sensors and OBUs that are capable of data sensing, processing and communications in order to earn benefits.
B. Threat Model

In this threat model, we assume vehicles are fully-trusted to submit crowdsensing reports for benefits. We focus our attention on the threat to fog nodes (e.g., RSUs) that are considered as semi-trusted. RSUs are regarded as honest in faithfully following the proposed scheme to identify deduplication, but they may be curious regarding the vehicles private information, therefore launch passive attacks. Specifically, RSUs may strive to know sensitive crowdsensing information for a specific vehicle, e.g., location. We also take into account the scenario where malicious RSUs can transmit forged reports to an organization by making them evaluate in a certain way. Furthermore, external attackers may eavesdrop on wireless communication channels to capture the information exchanged between entities vehicles,
RSUs, CSs and organizations. In addition, the KGC cannot be fully trusted because it may be managed by a commercial organization. As a result, it may access users private information and disclose their crowdsensing reports as well.

C. Design Goals

We aim to achieve the following security objectives based on the system model and potential threats. Our proposed CLASC [11] has achieved security requirements such as data confidentiality, integrity, mutual authentication, anonymity, and key escrow resilience. Motivated by the problem statement described, our design goal is to develop an efficient deduplicated reporting scheme in FVCS, which can achieve the secure data deduplication requirement. This technique provides efficient and secure ways to reduce the storage, bandwidth, and computation overhead of the RSU, CS and organizations. In particular, the RSU is able to detect reduplicated crowdsensing reports and delete them without learning any information about the reports. For example, the RSU only needs one copy of the repeated reports, which the remainder of them can be deleted. The contributions of the vehicles who submit the repeated reports should be recorded in order to fulfill vehicles’ fairness. Consequently, the RSU aggregates the signatures on the identical crowdsensing reports generated by different vehicles in order to simultaneously achieve fairness and minimize storage costs and communication overhead.

5.3 Preliminaries

This section starts with basic concepts and presents the necessary complexity assumptions of the proposed scheme.

A. Bilinear Maps.

Let $G$ be a cyclic multiplicative group with the generator $g$ of large prime order $q$, and $G_T$ be another cyclic multiplicative group of the same large prime order $q$. An admissible bilinear pairing $\hat{e} : G \times G \rightarrow G_T$ is a map with the following properties.

- Bilinearity. For all $u, v \in G$ and $a, b \in \mathbb{Z}_q^*$, we have $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$. 

75
• Non-degeneracy. \( \hat{e}(g, g_1) \neq 1_{G_T} \) where \( 1_{G_T} \) denotes the identity element of group \( G \).

• Computability. There exists an efficient algorithm to compute \( \hat{e}(g, g_1) \) for \( g, g_1 \in G \). \( \hat{e} : G \times G \to G_T \) is an admissible bilinear pairing.

B. Complexity Assumptions.

We recall the following intractability assumptions related to the security of our scheme.

**Definition 1** (Computational Diffie-Hellman (CDH) Problem): Given a tuple \( (g, g^a, g^b) \in G \) for some \( a, b \in \mathbb{Z}_q^* \), the CDH problem in \( G \) is to compute the element \( g^{ab} \).

**Definition 2** (Decisional Bilinear Diffie-Hellman (DBDH)): Given a tuple \( (g, g^a, g^b, g^c) \in G \) for some \( a, b, c \in \mathbb{Z}_q^* \) and \( x \in G_T \), the DBDH problem is to decide whether \( x \) holds \( \hat{e}(g, g)^{abc} \) or not.

5.4 Proposed Privacy-Preserving Data Deduplication Scheme

In this section, we describe the proposed scheme, which consists of four steps: system setup, reports collection and sending, reports deduplication, and reports receiving.

1. System Setup

All participants including vehicle, RSU, CS and organization register with the KGC to generate their public and full private keys. Given the security parameter \( k \), this phase is performed by the KGC as follows:

• Selects a cyclic multiplicative group \( G \) with the generator \( g \) of large prime order \( q \), and \( G_T \) be another cyclic multiplicative group of the same order \( q \) and a bilinear map \( \hat{e} : G \times G \to G_T \).

• Chooses a random master secret key \( s \in \mathbb{Z}_q^* \) and a random arbitrary generator \( g \in G \). It also computes its master public key \( P_{pub} = g^s \).

• Chooses three secure hash functions: \( H_1 : \{0, 1\}^* \to G \), \( H_2 : G_T \to G \) and \( H_3 : \mathbb{Z}_q^* \to G \).
Publishes the system parameter $\text{params} = (G, G_T, \hat{e}, g, q, P_{pub}, H_1, H_2, H_3)$ and the master secret key $s$ will be kept secure.

In order to accomplish the key generation phase, a vehicle with identity $U_i$ can randomly choose $x_i \in \mathbb{Z}_q^*$ as its secret value and compute its public key $PK_i = g^{x_i}$. Then, $U_i$ sends its identity and public key $(U_i, PK_i)$ to the KGC for registration. To achieve the identity privacy, KGC computes $Q_i = H_1(U_i)$ as its pseudo identity. Furthermore, the KGC then computes the partial private key $D_i = Q_i^s$ that is sent to the corresponding user $Q_i$ via a secure channel. Finally, $Q_i$ receives the partial private key $D_i$. Thus, its full private key is $(D_i, x_i)$.

Likewise, all participants are provided the key pair as follows:

- RSU with the identity $Q_{RSU} = H_1(ID_{RSU})$ has its public key $PK_{RSU}$ and full private key $(D_{RSU}, x_{RSU})$.
- CS with the identity $Q_{CS} = H_1(ID_{CS})$ has its public key $PK_{CS}$ and full private key $(D_{CS}, x_{CS})$.
- Organization with the identity $Q_{org} = H_1(ID_{org})$ has its public key $PK_{org}$ and full private key $(D_{org}, x_{org})$.

2. Reports Collection and Sending

The organizations release their spatial tasks on a CS that then provides them for vehicles through RSU as a fog node. When vehicle $Q_i$ receives the task $T$ with a unique identifier $N \in \mathbb{Z}_q^*$, $Q_i$ starts to sense the data $m \in \mathbb{Z}_q^*$ from the environment according to the requirements of the $T$. As a result, $Q_i$ then generates a CrowdSensing Report $CSR_i$. In order to protect the $CSR_i$ from being disclosed and achieve confidentiality, integrity and authenticity in one logical step, $Q_i$ signcrypts $m$ and performs the following steps:

- Randomly selects $r_i \in \mathbb{Z}_q^*$ and computes the following,
  
  $\bullet$ $C_i = g^{r_i}$.
  
  $\bullet$ $K_i = \hat{e}(P_{pub}, Q_{org}^{r_i})$.
  
  $\bullet$ $X_i = H_2(K_i)$. 

77
- \( L_i = H_1(m)^{r_i} \).
- \( P_i = PK_{org}^{r_i} \).
- \( ENC_i = m X_i \cdot P_i \).
- \( h_a = (Q_{org} \parallel PK_{org} \parallel C_i \parallel ENC_i) \).
- \( \alpha_i = D_i \cdot h_a^{r_i} \cdot ENC_i^{\alpha_i} \).
- \( U_i \) sets \( CSR_i = (N, C_i, L_i, ENC_i, \alpha_i) \) and forwards it to the fog node \( Q_{RSU} \).

It is worth pointing out that we have used the concept of CE by considering that \( L_i \) is derived from the report data \( m \) to support the reduplicate data detection and deletion in the ciphertext \( CSR_i \). In the phase, \( C_i \) and \( L_i \) are used to detect the duplication of \( (CSR)_i^{n-1} \). Therefore, the fog node \( RSU = Q_{RSU} \) can identify the reduplicate data based on \( C_i \) and \( L_i \) in the ciphertext \( CSR_i \).

3. Reports Deduplication

In this phase, \( RSU \) is able to detect the reduplicate data while \( CSR_i \) is signcrypted. In this application scenario, \( RSU \) is a semi-trusted node that may bring security and privacy issues to participating vehicles. As a result, \( RSU \) performs homomorphic calculations on the ciphertext to address these concerns. After \( RSU \) receives \( CSR_i = (N, C_i, L_i, ENC_i, \alpha_i) \) from \( Q_i \), it performs the following:

- Checks if the reduplicate data exists or not in the crowdsensing reports \( (CSR_i)_i^{n-1} \).
- Checks the condition 
  \[ \hat{e}(C_i, L_j) = \hat{e}(C_j, L_i) \]
- If it is fulfilled, therefore user \( i \)’s report \( CSR_i \) is identical to user \( j \)’s report \( CSR_j \).
- \( RSU \) sorts the identical reports \( \{CSR_i\}_i^{k} \), where \( k \subseteq n \) and is the set of indices of reduplicate reports.
- In order to achieve fairness among vehicles and record all the contributors, \( RSU \) aggregates the corresponding signatures for all identical reports as 
  \[ \text{sig}_{agg} = \sum_{i=1}^{k} \alpha_i \].
• As illustrated in Figure 5.3, RSU randomly chooses one copy $CSR'_i = (N, C_i, ENC_i, sig_{agg})$, where $i \in k$.

• RSU also sets the other reports that are not reduplicated with others as $CSR_w = (N, C_w, ENC_w, \alpha_w)$, where $w \in n$ and $w \not\in k$.

• RSU then forwards the deduplicated reports $(CSR'_i, (CSR_w)^n_{w=1 \not\in k})$ to the CS who will forward it to the organization $Q_{org}$.

4. Reports Receiving

The organization receives the deduplicated reports from the CS. In order to verify the signature aggregation $sig_{agg}$ and check the contributors for $CSR'_i$, $Q_{org}$ computes $h_a = (Q_{org} \mid PK_{org} \mid C_i \mid ENC_i)$. Then, the signature aggregation $sig_{agg}$ accepts if
\[\hat{e}(\mathit{sig}_\text{agg}, g) = \prod_{i=1}^{k} \hat{e}(P_{\mathit{pub}}, Q_i) \prod_{i=1}^{k} \hat{e}(C_i, h_a) \prod_{i=1}^{k} \hat{e}(PK_i, \mathit{ENC}_i)\]

When the signature verification outputs true, \(Q_{\text{org}}\) decrypts the deduplicated report \(\mathit{CSR}'\) as follows:

- Computes \(K_{\text{org}} = \hat{e}(D_{\text{org}}, C_i)\).
- Computes \(P_{\text{org}} = (C_i)^{x_{\text{org}}}\).
- Computes \(Y_{\text{org}} = H_2(K_{\text{org}})\).
- \(Q_{\text{org}}\) then accesses the report by computing \(m' = ENC_i / Y_{\text{org}} \cdot P_{\text{org}}\).
- Likewise, \(Q_{\text{org}}\) can also unsigncrypt other deduplaicated reports \((\mathit{CSR}_w)_{w=1}^{n} \notin k\). Finally, \(Q_{\text{org}}\) obtains the reports and distributes the rewards to vehicles based on their contributions.

The correctness of the decryption is as follows

\[
m' = \frac{ENC_i}{Y_{\text{org}} \cdot P_{\text{org}}} = \frac{mX_i \cdot P_i}{H_2(K_{\text{org}}) \cdot (C_i)^{x_{\text{org}}}} = \frac{mH_2(K_i) \cdot PK_{\text{org}}^{r_i}}{H_2(\hat{e}(D_{\text{org}}, C_i)) \cdot (g^{r_i})^{x_{\text{org}}}} = \frac{mH_2(\hat{e}(P_{\mathit{pub}}, Q_{\text{org}}^{r_i})) \cdot PK_{\text{org}}^{r_i}}{H_2(\hat{e}(Q_{\text{org}}^s, g^{r_i})) \cdot (g^{x_{\text{org}}})^{r_i}} = \frac{mH_2(\hat{e}(P_{\mathit{pub}}, Q_{\text{org}}^{r_i}))) \cdot PK_{\text{org}}^{r_i}}{H_2(\hat{e}(Q_{\text{org}}^s, g^{s})) \cdot PK_{\text{org}}^{r_i}} = m
\]

The correctness of the signature is as follows

\[\hat{e}(\mathit{sig}_\text{agg}, g) = \prod_{i=1}^{k} \hat{e}(\alpha_i, g)\]
\[
\prod_{i=1}^{k} \hat{e}(D_i \cdot h_{a_i}^{r_i} \cdot ENC_{i}^{x_i}, g) \\
= \prod_{i=1}^{k} \hat{e}(D_i, g) \prod_{i=1}^{k} \hat{e}(h_{a_i}^{r_i}, g) \prod_{i=1}^{k} \hat{e}(ENC_{i}^{x_i}, g) \\
= \prod_{i=1}^{k} \hat{e}(Q_i^s, g) \prod_{i=1}^{k} \hat{e}(h_{a_i}, g^{r_i}) \prod_{i=1}^{k} \hat{e}(ENC_{i}, g^{x_i}) \\
= \prod_{i=1}^{k} \hat{e}(Q_i, g^s) \prod_{i=1}^{k} \hat{e}(h_{a_i}, C_i) \prod_{i=1}^{k} \hat{e}(ENC_{i}, PK_i) \\
= \prod_{i=1}^{k} \hat{e}(Q_i, P_{pub}) \prod_{i=1}^{k} \hat{e}(h_{a_i}, C_i) \prod_{i=1}^{k} \hat{e}(ENC_{i}, PK_i)
\]

The correctness of the deduplication is as follows

\[
\hat{e}(C_i, L_j) = \hat{e}(g^{r_i}, H_3(m)^{r_j}) \\
= \hat{e}(g^{r_i}, H_3(m)^{r_i}) \\
= \hat{e}(C_j, L_i)
\]

### 5.5 Security Analysis

In this section, the proposed scheme is analyzed and shows the achievement of the design goals described in Section 5.2. In the considered system model, RSU as a fog node is capable to perform data deduplication without disclosing any information about the crowdsensing reports. In order to support data deduplication property in the crowdsensing reports, the concept of CE is used in the CSR as described in the reports collection and sending phase. To achieve data confidentiality, integrity, mutual authentication, anonymity, and key escrow resilience in the CSR, we implement our proposed CLASC scheme. Once RSU receives ciphertexts \((CSR_i)_{i=1}^{n}\) from vehicles, it can identify whether two reports are identical or not by comparing their ciphertexts. Specifically, the RSU checks duplicates by running reports’ deduplication algorithm...
$\hat{e}(C_i, L_j) = \hat{e}(C_j, L_i)$ between two ciphertexts. RSU can determine the reduplicate data based on $C_i$ and $L_i$ tags in the ciphertext $CSR_i$. After the RSU performs homomorphic calculations and runs the reports deduplication algorithm to detect reduplicate reports, it keeps one copy of them to achieve deduplication property and address vehicles’ privacy concerns simultaneously. In order to save the storage and bandwidth in the organization side, it only needs to decrypt the deduplicated reports by its private key $(D_{\text{org}}, x_{\text{org}})$ to follow the CLASC scheme correctly to obtain these reports. Specifically, the convergent item $L_i$ is not needed to use for decryption by the organization such as insurance company due to the RSU being the only component who needs $L_i$ to perform server side secure deduplication and insure performing secure deduplication without learning the information. The organization only concerns about the signature verification and report decryption. We utilize signature aggregation technique to help record all the contributions of identical reports by vehicles $(Q_i)_{i=1}^k$. $\alpha_i$ generated on $CSR_i$, such that $Q_{\text{org}}$ is able to verify $\alpha_i$ and check out whether $Q_i$ is the contributor of $CSR_i$ or not.

In particular, when vehicles $(Q_i)_{i=1}^k$ sense and generate the same report $CSR_i$, $Q_{\text{RSU}}$ aggregates the corresponding signatures $\alpha_i \in k$. Therefore, deduplication technique improves the communication overhead by reducing the bandwidth from $Q_{\text{RSU}}$ to $Q_{\text{org}}$. Obviously, $Q_{\text{org}}$ can perform verification of the signatures $\alpha_i \in k$ with corresponding public keys of vehicles $(Q_i)_{i=1}^k$. On the basis of CLASC scheme [11], the proposed scheme achieves confidentiality and unforgeability under CDH and DBDH assumptions. Therefore, the adversary cannot forge the signatures or claim the contributions.

### 5.6 Performance Evaluation

In order to evaluate the efficiency of the proposed scheme, we count the number of cryptographic operations, including the exponentiation in $G$ and pairing. These operations dominate the computational cost of the algorithms and are considered in the time consumption. The other operations such as the hash and multiplication in $G$ are negligible and not time consuming. We denote $t_{\text{expG}}$ the consumption of the exponentiation in $G$ and $t_p$ the consumption of pairing. To quantify the running time of the cryptographic operations in the proposed scheme, the pairing-based cryptography (PBC) library ver.0.4.18 [52] is used and the implementation of the algorithms was
executed on a 3.2 Ghz Pentium IV machine [81]. The running time are \( t_{\text{expG}} = 6.4 \) ms and \( t_p = 5.9 \) ms. The cryptographic operations and the run time of each phase in the proposed scheme are illustrated in Table 5.1. The pairing is utilized to realize the bilinear pairing operation and the elliptic curve is defined as \( y^2 = x^3 + x \) over a base field size of 512 bits. To ensure the security of the proposed scheme, the parameter \( q \) is approximately 160 bits.

<table>
<thead>
<tr>
<th>Phases</th>
<th>Operations</th>
<th>Run time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>System setup</td>
<td>( 3 t_{\text{expG}} )</td>
<td>19.2</td>
</tr>
<tr>
<td>Report collection and sending</td>
<td>( 6 t_{\text{expG}} + t_p )</td>
<td>44.3</td>
</tr>
<tr>
<td>Report deduplication</td>
<td>( 2 t_p )</td>
<td>11.8</td>
</tr>
<tr>
<td>Report receiving</td>
<td>( t_{\text{expG}} + 5 t_p )</td>
<td>35.9</td>
</tr>
</tbody>
</table>

Table 5.1: Computational cost of the proposed scheme

In the proposed scheme, the communication overhead is analyzed among vehicles, RSUs, CSs and organizations in terms of the length of ciphertext. We first consider the communication between vehicles to RSU, where vehicles generate their crowdsensing reports and deliver them to the local RSU. The report is in the form \( CSR_i = (N, C_i, L_i, ENC_i, \alpha_i) \) for vehicles \((Q_i)_{i=1}^n\), who need to forward 1664 bits. Notably, the parameter \( q \) is set to be 160 bits. RSU with identity \( Q_{RSU} \) receives \((CSR_i)_{i=1}^n\) from \((Q_i)_{i=1}^n\) and performs the data deduplication. If the reduplicate data exists, it sorts the identical reports \( \{CSR_i\}_{i=1}^k \), where \( k \) is the number of reduplicate reports in \((CSR_i)_{i=1}^n\), and randomly chooses one copy from \( k \). Then, we consider the communication of both RSU, CS and organization. Eventually, RSU forwards only one copy 1504 bits for the set of identical reports \( k \). In case there is no reduplicate crowdsensing reports, the communication overhead between RSU, CS and organization is 1504 bits for each report in \((CSR_w)_{w=1}^n\). Consequently, these binary lengths represent the ciphertext \( (CSR', (CSR_w)_{w=1\neq k}) \). Then, the CS receives this ciphertext and sends it to the organization \( Q_{org} \). Also, we compare the communication overhead with an existing deduplication scheme in cloud server side [99], which consumes \( 2|G_T| + 2|Z_q^*| \) to find data deduplication, where \( |G_T| \) and \( |Z_q^*| \) denote the size of the elements in \( G_T \) and \( Z_q^* \). The proposed scheme is much efficient and only needs \( 2|G_T| \) to perform data deduplication on ciphertexts.
We also take into account the bandwidth overhead in terms of the total size of crowdsensing reports between RSU, CS and organization. In fact, when the number of vehicles increases, the chance of existing identical data in the crowdsensing reports is highly possible and may increase as well. In our experiment, we compare with a scheme that does not provide deduplication mechanism as shown in Figure 5.4. We consider that there are 20\%, 40\%, 60\% and 80\% of identical
data detected in the reports with the number of vehicles 50, 100, 150 and 200 respectively. As illustrated in Figure 5.4 (a), RSU utilizes the ordinary way to send all the generated reports including similar reports. The RSU thus incurs a high cost for report size to process and forward. For example, in Figure 5.4 (a), RSU sends 37.5 kilobyte (KB) of reports size with 200 vehicles. On the other hand, we implement the deduplication technique in this experiment and show that the RSU only sends one copy of reduplicated reports and the other dissimilar reports. Specifically, RSU sends 7.6 KB of reports size with 200 vehicles as shown in Figure 5.4 (b). We can deduce that the bandwidth overhead of the proposed scheme remains efficient even if the number of vehicles increases, e.g., in Figure 5.4 (a) we observe RSU processes 9.4 KB generated from 50 vehicles while in the proposed scheme RSU processes 7.6 KB generated from 200 vehicles. In particular, Figure 5.5 illustrates how the proposed scheme can tackle the scalability concern and evidently has lowest computational than the traditional way that does not provide deduplication technique. From the above analysis, the proposed scheme is indeed efficient in terms of computational cost, communication and bandwidth overhead. Thus, it is suitable for the real-time data applications in FVCS.

![Figure 5.5: Communication overhead comparison](image-url)
5.7 Summary

In this chapter, we propose a deduplicated reporting scheme for reducing the communication overhead between fog nodes, CS and organizations. Moreover, the proposed scheme provides a homomorphic technique to insure the performance of a secure computation by a semi-trusted RSU. Specifically, the RSU is able to detect reduplicated data in crowdsensing reports and delete any repeated copies without accessing any information about the reports’ contents. Furthermore, fairness is achieved between the vehicles whose reports are reduplicated and deleted. Thus, the proposed scheme fulfills security requirements including secure data deduplication, confidentiality, integrity, anonymity, mutual authentication and key escrow resilience. This chapter elaborates on the properties of our proposed scheme and demonstrates its efficiency in terms of computational cost, communication and bandwidth overhead.

5.8 Limitation

While we strove to provide a secure environment to insure users can join and participate our system, the proposed privacy-preserving scheme limits to meet a certain privacy scenario. For example, distributing rewards to the participating vehicles when sharing their resources may be unfair. Specifically, a lazy vehicle can submit a crowdsensing report that is received from the neighbor vehicle. In this work, we don’t discuss the incentive techniques that may use to address such kinds of these privacy issues. To be more specific, the major concentration is to discuss and address a promising security and privacy problem representing in a semi-trusted node who is in charge to perform computations on sensitive information. Furthermore, the work’s scope is also to solve another efficiency challenge where vehicles generate repeated crowdsensing reports from the same location.
Chapter 6

Conclusions and Future Work

In this chapter, the contributions of this dissertation are concluded and the future work is introduced as well.

6.1 Contributions

The major contributions of this thesis are mainly in threefold:

- First, a novel fog-assisted vehicular crowdsensing (FVCS) framework is proposed for improving the efficiency and accuracy of the road condition monitoring system that results in road quality and safety. Specifically, all the reports generated by vehicles are processed near the end user rather than being processed in the centralized cloud. Therefore, the proposed framework is suitable for the real-time VCS applications. Furthermore, a novel privacy-preserving protocol is designed for enhancing security in the FVCS [11] in order to protect vehicles privacy from being disclosed during a report formulation and generation. Initially, we propose a highly efficient certificateless aggregate signcryption (CLASC) scheme. On the basis of the CLASC scheme, a privacy-preserving protocol for monitoring road surface conditions is designed. The proposed protocol combines CL-PKC and signcryption technique in order to protect the vehicles privacy from being disclosed.
during generating reports. The proposed scheme is significantly more efficient than the existing schemes [31], [51] in terms of computational costs and communication overhead.

• Second, an efficient revocable privacy-preserving protocol in FVCS is designed in order to let the system revoke the compromised users [10]. The revocation technique is implemented to address the issues when a private key of a participated user is being compromised after it is detected as malicious. The proposed protocol makes use of a combination of a binary tree structure with a certificateless signcryption technique to achieve compromised user revocation that results in preserving road condition reports generated by vehicles from being disclosed. It also protects these reports that are processed by compromised RSUs from accessing and revealing them. Extensive simulations demonstrate efficiency with regard to computational cost and ciphertext size of the proposed protocol. In terms of scalability, user revocation, signature verification process and key escrow problem evasion, the proposed protocol outperforms existing competing schemes [98], [68], [92].

• Finally, due to the fact that there are inevitably some duplicates in the crowdsensing reports generated by vehicles at the same location, and that gateways as RSUs may be corrupted, an efficient deduplicated reporting scheme in FVCS is proposed [9] in order to address these challenges. A homomorphic concept with a signcryption technique is integrated to let semi-trusted gateways process and analyze the encrypted crowdsensing reports. The proposed scheme also supports deduplication process on the reports without revealing the sensitive information of the participating vehicles. Furthermore, the proposed scheme is much more efficient and guarantees fairness between vehicles whose reports are reduplicated and deleted. The simulation results show the efficiency of the proposed scheme in terms of both communication and computational overheads. The proposed scheme also demonstrates the achievement of secure data deduplication property.

6.2 Future Work

Future work will endeavor to advance the developed framework into a real-life situation by building FVCS applications in collaboration with the automobile industry. Furthermore, the following research topics require further investigation as continuation of the current Ph.D. study.
6.2.1 Differential Privacy in Fog-Assisted Vehicular Crowdsensing

The ability to collect data from modern connected vehicles presents opportunities for increased analysis, which enables vehicle manufacturers to both improve existing work and develop new services. For example, investigating driving behaviour would make it possible to learn more about the drivers’ needs and preferences, allowing manufacturers to better cater to customers’ needs.

However, gathering data from vehicles is not only an opportunity for further analysis, but also a possible privacy risk to the individual drivers. A recent survey shows that drivers privacy concerns include disclosure of private information, vehicle tracking and commercial use of their personal data. Therefore, privacy is a concern for drivers when it comes to connected vehicles. Thus, the problem needs to be addressed by the manufacturers in order to maintain the drivers trust.

While there is no general solution or scheme for a wide variety of many privacy issues in VCS and how we should properly address them using different techniques, we plan to study differential privacy as a promising technique. Unlike other types of privacy models, differential privacy might be applied to statistical databases for VCS in order to preserve drivers’ data privacy. Also, it is able to provide high accuracy and privacy in many cases.

6.2.2 A Fog-Assisted Vehicular Crowdsensing Framework: A Technology Management Perspective

Although a fog-assisted vehicular crowdsensing (FVCS) framework potentially provides major advances in the field of vehicular crowdsensing, there is an urgent need for understanding the business-related aspects surrounding FVCS technology. One of the design goals is to propose a competitive product that outperforms its competitors. Thus, in order to facilitate achieving this objective, a business perspective is also taken into account in terms of: technology management and its importance; strategic analysis; technology recommendations; technology forecasting; cost-efficient FVCS deployment; and stakeholders.
6.2.2.1 Technology Management

Basically, technology management refers to the set of policies and practices that leverage technologies to build, maintain, and enhance competitive advantage [20]. Managing technology within an organization is extremely important in order to maximize profits and gain an edge over the competition. In specific terms, planning, designing, optimizing, and operating and controlling technology are the fundamentals phases for technology management, which aims at maximizing the cost effectiveness of investments in technology development that contributes to the value of an organization. As an example, if an organization fails to plan for its technology, it may encounter issues such as data loss or misuse of that technology. In contrast, the output for this organization will increase if it creates a framework and plans for its technology. Indeed, an important initial phase of technology management is how strategic analysis of the proposed technology is conducted.

6.2.2.2 Strategic analysis

This subsection focuses on the potential of FVCS by conducting a SWOT analysis as well as Porter’s five forces analysis. Both of these tools are commonly used by companies to conduct analyses and to make strategic decisions. Each of the models seeks to define the company’s position in the market. The major distinction is that Porter’s five forces model is used to analyze the competitive environment within an industry and focus on external forces, while a SWOT analysis tends to look more deeply within an organization to analyze its internal potential.

i. SWOT Analysis. The strengths, weaknesses, opportunities and threats of FVCS are described as follows:

- *Strengths*. As previously mentioned in Chapter 1, Section 1.1, fog computing fills the gap between the cloud network and Internet of Things (IoT) devices. As such, fog computing enables end-user IoT devices to have computing, storage, networking and communications functions. This promising paradigm solves the bandwidth, latency and communications challenges associated with the next generation networks that will utilize IoT, 5G and artificial intelligence devices. Significantly, fog computing, which
is emerging as a powerful architecture for extending cloud network services to the network edge, provides a business with the strengths to build the necessary architecture. Fog computing adds immediate value to a business by enabling the acceleration of rollout cycles, reducing costs, and broadening revenue bases. By solving bandwidth, latency and communications challenges, fog computing makes the production of revenue generating products and services more efficient and thus more cost-effective. In addition, the shred-and-spread nature of fog architecture means that the business also runs more efficiently and cost-effectively. Because the architecture shreds across devices and spreads across clouds, fog computing provides highly functioning internal business services while expanding the overall scalability of the business. Furthermore, IoT and fog computing paradigms bring the advantage of new revenue streams. Thus, fog computing will promote the development of long-awaited revenue-generating applications and services because fog creates value for IoT. An additional strength lies in the leverage of current information technology (IT) investments. Because fog provides cloud-to-IoT attributes, it augments business investment in cloud networking and enables future efficient, cost-effective, and constructive use of IoT server technology. Fog architecture is designed to grow wherever the IoT market grows. Routers, switches, application servers and storage servers will converge into fog nodes where each fog node will be capable of providing a common hardware and software platform that supports computing, networking, and storage.

• **Weaknesses.** There are some concerns that need to be addressed before fog computing can be accepted as a viable choice in business computing. Organizations will be justifiably wary of the loss of physical control of the data that is put on fog nodes. This is due to the fact that the fog computing concept enables developers to access the most important IoT data from different locations. However, it still keeps a significant volume of less important information in local storage. Furthermore, some organizations are reluctant to have data leave their premises. In particular, fog computing provides the property whereby a considerable amount of data is stored on the devices themselves (which are often located outside of organizations’ buildings). Thus, this may be perceived as a risk for some developer communities.

• **Opportunities.** Fog computing will enable new and potentially highly disruptive busi-
ness models for computing and networking. It is already influencing how edge networks are being built. Specifically, routers, switches, application servers, and storage servers will converge into fog nodes. Such a transformation can significantly reshape the networking, server, and software industry landscape, which as a result often features a unified networking platform that supports heterogeneous networking technologies, and a common computing platform that supports applications from multiple suppliers. This ongoing convergence of computing, networking and storage at the edge will significantly reduce system complexity and cost, increase system and application manageability, and make it easier for applications to interact with each other. In addition, Fog-as-a-Service (FaaS) will enable new business models to deliver services to customers. Unlike clouds that are mostly operated by large companies that can afford to build and operate vast data centers, FaaS will enable companies, large and small, to deliver private or public computing, storage, and control services on different scales to meet the needs of a wide variety of customers.

- **Threats.** While we believe that many forward-looking organizations will see fog computing as an opportunity to migrate to better computing practices that open up exciting opportunities for IT, there will probably be many threats to their corporate IT culture in terms of data security and privacy. Any business will focus its attention on these threats in order to benefit from the services provided by edge paradigms. One of the most important fog aspects is data security. As with any IoT project, the presence of multiple weakly-secured devices connected to a network creates greater vulnerability to cyber-attacks. An edge strategy can significantly strengthen a network by keeping major threats away from its core. With this activity taking place between local end-points, threats such as privacy disclosure or data breaches can be identified at an earlier stage and contained at device level. Individual privacy can be increased by processing person-specific data in the fog, rather than it being collected and stored in a centralized database available to company staff.

ii. **Porter’s Five Forces Analysis**

   One of the greatest challenges to any business is competition. In particular, a business
should identify its competitors and how their actions in the marketplace are going to affect the current bottom line and future planning. A useful analysis tool is the model devised by Porter [66], which considers five distinct categories that help determine whether a business can be profitable, based on other businesses in the industry. Thus, understanding competitive forces, and their underlying causes, reveals the roots of an industry’s current profitability while providing a framework for anticipating and influencing competition over time. Porter regarded understanding both the competitive forces and the overall industry structure as crucial for effective strategic decision-making. In Porter’s model, the five forces that shape industry competition are described as follows:

- **Competitive rivalry.** This force examines the current level of competition intensity in the marketplace, which is determined by the number of existing competitors and what each is capable of doing. For example, if the existing competitors in mobile crowdsensing real-time systems, such as Google maps and Waze, are about to enter into this framework, the proposed novel technology can tackle intense rivalry among them. Regardless, the proposed product can serve better and more efficiently in terms of fog-assisted vehicular crowdsensing real-time systems. Intense rivalry among existing competitors can be managed by building a sustainable differentiation scale, creating increased competitiveness and greater collaboration with competitors. Rather than just competing within a small market, this will increase market size as well as challenge intense rivalry among existing competitors.

- **Bargaining power of suppliers.** This force analyzes how much power a business’s supplier has and how much control it has over the potential to raise its prices which, in turn, would lower the business’s profitability. Building an efficient supply chain with multiple suppliers, and developing dedicated suppliers whose business depends upon the firm, will challenge the bargaining power of suppliers.

- **Bargaining power of customers.** This force looks at the power of the consumer to affect pricing and quality. Consumers want to buy the best offerings available while paying the minimum possible price. FVCS technology can tackle their bargaining power by building a large customer base. This will help reduce the bargaining power of customers and provide an opportunity for the firm to streamline its sales and pro-
duction process. In addition, rapidly innovating new services help limit the bargaining power of customers who often seek discounts and offerings on established services.

- Threat of new entrants. In the mobile and vehicular crowdsensing fields, new entrants, who bring innovation at all levels, put pressure on the proposed FVCS technology through applying a lower pricing strategy, reducing costs, and providing new value propositions to customers. A business must manage all these challenges and build effective barriers to safeguard its competitive edge. In order to address the threats of new entrants, innovating new services can bring new customers and also give long-standing customers a reason to buy new services.

- Threat of substitute products or services. This force studies how easy it is for consumers to switch from a business’s product or service to that of a competitor. It considers how many competitors there are, how their prices and quality compare to the business being examined and how much profit those competitors are earning, which would determine if they can lower their costs even more. Thus, firm profitability suffers when a new product or service meets similar customer needs in different ways. By being service-oriented, understanding the core need of the customer rather than what the customer is buying and increasing the switching cost for them, FVCS can tackle the threat of substitute products or services.

This present work proposes to extensively investigate these forces in existing competitive mobile crowdsensing real-time systems, such as Google maps and Waze, compared to the developed FVCS framework. After considering the aforementioned strategic analysis regarding the proposed framework, how it can be recommended to other firms is discussed.

### 6.2.2.3 Recommendations

A set of recommendations to help manage this technology and attract other businesses into the developed technology is also considered and investigated. The focus of this present study is on the proposed FVCS technology that will be needed to migrate to fog computing rather than the existing cloud computing paradigm. In fact, businesses such as VCS applications handle significantly large amounts of data on a regular basis [57]. As a result, cloud-based and fog-based solutions were developed to streamline the data organization process and help businesses
manage their data in real-time. Choosing which model is best for a particular business depends on the amount of data the business organizes and manages. Fog computing and cloud computing are both excellent models for data storage, management and analytics. However, many emerging IoT applications that require services rather closer to the edge cannot wait for data to travel all the way to the cloud and back [57]. Thus, cloud computing alone has become increasingly inadequate for IoT applications. Furthermore, for a business that accumulates higher levels of data, such as real-time road surface condition monitoring system (RSCMS) applications, as well as the ability to maintain a strict budget and control scalability, fog computing will optimize the data flow. Fog computing is ideal for businesses that regularly process large volumes of data as well as those that handle sensitive data, such as those in the transportation agencies for detecting road abnormalities. Indeed, in order to keep attracting other organizations, while increasing gain and reducing loss, technology forecasting should be considered.

6.2.2.4 Technological Forecasting

Technology changes have been recently acknowledged as a critical factor in determining the competitiveness of any organization [58]. In such an environment, accurate anticipation and forecast of this factor has been of considerable importance for incorporating technological changes into strategic planning processing [30]. Therefore, in order to monitor technology changes, an efficient forecast of technology can help maximize gain and minimize future loss for the organization.

Technological forecasting (TF) is the process of predicting the future characteristics and timing of technology [33]. TF is aimed at predicting future technological capabilities, attributes, and parameters. Forecasting a technology includes not only specific mechanical/physical hardware, but also includes associated software such as procedures and methods for organizing human activity, as well as the means for manipulating or engineering human behavior. Technology forecasting methods can be classified as exploratory and normative [15]. Exploratory technological forecasting starts from today’s certain basis of knowledge and is oriented towards the future. By contrast, normative technological forecasting focuses on the creation of alternative technological paths to a desired, predefined end state and assesses issues such as future goals, needs, and desires. In fact, many changes are currently taking place in different technologies such as soft-
ware, computing and hardware. Thus, there is a need to find the trends and predict the future of these technologies. Technology forecasting methods are useful in identifying these trends and predicting the future of the proposed FVCS framework. This present work focuses on normative technological forecasting since the proposed framework consists of two promising paradigms, namely fog computing and vehicular crowdsensing, which will make a bright emergence in the near future in the form of smart cities.

Normative methods are also known as goal-setting methods. Normative technological forecasting first estimates future goals, needs, and desires, and works backwards to the present. Such methods are used to determine the level of functional capability that must be achieved to solve a problem or overcome a difficulty. There are several types of normative technological forecasting methods.

Firstly, the relevance trees method is an organized normative approach that starts with a particular objective and is used for forecasting as well as planning. The basic structure resembles an organizational chart and presents information in a hierarchical structure. The principle behind using the relevance tree is to systematically evaluate all the related technologies that would lead to the success of the intended objective [58]. The relevance tree is a powerful and general technique with a wide range of applicability. It can be used for identifying new system alternatives and can be a method for obtaining different solutions to a given problem. However, the relevance tree for a large complex technology could become too complicated to handle.

Secondly, the Morphological Analysis (MA) method is a normative technique that provides a framework for exploring all possible solutions to a particular problem. This type of analysis involves the systematic study of the current and future scenarios of a particular problem. MA can be used to identify the requirements for individual technologies of a specific system, but cannot be used to obtain quantitative estimates of the relative importance of various technological goals. For example, MA is normally carried out only for specific short-term planning purposes and has not yet been applied to long-term forecasting. MA is a useful technique for encouraging the thinking process and allows examination of all combinations of alternatives to achieve the objective. Nevertheless, it is a static model and is not suited to manage systems that change with time or to describe a logical sequence of events.

Although normative methods provide advantages such as the generation of a range of al-
ternatives, cost effectiveness and simplicity of application, they suffer from certain drawbacks. These drawbacks include: the decision variables and hierarchical structure of technology must be known; the weights must be objective in nature; the time period is usually not clearly forecast; and accuracy cannot be clearly defined. As a consequence, the aim of this present work is to design a suitable and efficient method that will enhance the performance of conventional normative methods and determine the ideal configuration for the proposed FVCS framework.

6.2.2.5 Cost-Efficient FVCS Deployment

The previous subsections discuss technology management and forecasting for FVCS. Since businesses are more concerned about how to cost effectively deploy such technologies, this will be further developed in future work. In particular, RSU is a key component of the VCS infrastructure connecting mobile vehicles and the rest of the infrastructure. Road surface condition monitoring systems (RSCMS) is a large-scale application that RSUs should densely deploy in order to maximize their availability and avoid the blind spots that may exist causing vehicles to lose connection with the infrastructure. Nevertheless, the massive deployment of RSUs needed to seamlessly cover the total area of interest can be very expensive. Motivated by this observation, development of the present work will include the study of a new strategy to best deploy RSUs so that their spatial and temporal coverage is maximized under a limited budget. Thus, the proper distribution of RSUs is of great importance in improving the service quality of VCS applications. The system components that participate in the deployed technology should be determined as system stakeholders.

6.2.2.6 Stakeholders

Chapter 2, Section 2.4 describes the infrastructure of a FVCS framework for a road surface monitoring conditions system, and also shows how computing resources can be accessed from a variety of organizations. The proposed technology is evidently beneficial to organizations, such as insurance companies and municipalities, who seek access to transportation systems with real-time applications.

As shown in Figure 6.1, the main stakeholders of the proposed framework are as follows:
• **Consumers.** In the RSCMS, the consumers (e.g., insurance companies and municipalities) are effectively subscribers, who only purchase the use of the system from the providers on an operational expense basis. Corporate users of fog computing have an active role to play in ensuring that fog computing actually delivers on its promise to revolutionize corporate computing, by liaising with industry groups as well as national and international regulators.

• **Providers.** Fog computing service providers, such as Cisco IOX, own and operate fog computing systems in order to deliver services to third parties. They also provide RSUs as fog devices and data services. The providers will perform the maintenance and upgrades to the system of which the consumers were in charge when they owned the systems.
providers will also be responsible for maintaining the software used on the fog, along with the pricing of the fog services.

• **Customers.** In the RSCMS, the customers are the vehicles that sense and generate the crowdsensing reports in response to the announced tasks released by the organizations.

• **Regulators.** These are the public authorities or government agencies responsible for exercising autonomous authority over some area of human activity in a regulatory or supervisory capacity. In the RSCMS, the regulator comprises the directives that safeguard information technology and computer systems with the purpose of forcing companies and organizations to protect their systems and information from cyber attacks.

Investigating a business perspective related to FVCS technology is a very significant step in introducing a solid product into the real world. The specific focus of this present work is to evaluate the proposed technology from different business aspects in order to provide a competitive product that outperforms its competitors. These aspects include technology management, strategic analysis, technology recommendations, technology forecasting, cost-efficient FVCS deployment and stakeholders. Finally, future development of this present study will include further extensive investigation of the business perspective in relation to the proposed technology as well as to produce a successful product that will outperform its counterparts.
Bibliography


104


