Enhancing Security, Privacy, and Efficiency of Vehicular Networks

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Sarah Al-Shareeda, M.S.

Graduate Program in Department of Electrical and Computer Engineering

The Ohio State University

2017

Dissertation Committee:

Prof. Füsun Özgüner, Advisor
Prof. Can Emre Koksal
Prof. Xiaorui Wang
Abstract

Vehicular Adhoc Networks (VANETs) promises to empower the future autonomous vehicles with a cooperative awareness facility that will help in avoiding accidents and alleviating traffic congestion. The foreseen collective awareness requires the vehicles to communicate with their neighbors and with the infrastructure; such communication will need the fulfillment of many requirements such as security, privacy, and efficiency. The Dedicated Short-Range Communication (DSRC) standard has been formulated to afford these requisites. On one hand, when focusing on the application layer, DSRC adopts the successful Internet-based Public Key Infrastructure (PKI) framework to safeguard the vehicles. However, PKI alone cannot comprehensively meet all of the security and privacy requirements. On the other hand, the DSRC’s Medium Access Control (MAC) layer adopts the IEEE 802.11p access mode, which also needs augmentation to fulfill the efficiency of communication when collisions arise for safety beacons. Since many issues have not been well addressed in DSRC, academic, industrial, and governmental research has flourished over the last two decades to complement the standard. As being part of such large research community, we also have been incentivized to contribute with our own solutions. Our contributions have been ranging between two limits: either finding solutions to acclimate with the available DSRC shortcomings or disregarding the bias that DSRC has towards using only specific standards by bringing other alternative frameworks into scene. With the
first end in mind, our efforts are a mixture of high-level re-arrangement protocols such as grouping and overhead omissions to minimize the PKI and Carrier Sense Multiple Access - Collision Avoidance (CSMA/CA) privacy and efficiency shortcomings. For the other limit, we especially address the application layer level. Since some frameworks have small communication overhead while others have high anonymous traits, we have attempted low-level alternatives to PKI and Elliptic Curve Integrated Encryption Scheme (ECIES) and to overcome their confidentiality, privacy, and efficiency limitations. First, to augment the security of sensitive non-safety applications in PKI, our first research track concerns itself with finding alternatives for the used low-level encryption primitive such as ECIES and Advanced Encryption Standard (AES). The reason behind such effort is the authentication-dependability of ECIES/AES and key management of AES; therefore, we investigate the suitability of using a state-of-the-art low-level partial homomorphic encryption scheme to generate encrypted identities and keys to secure the sensitive non-safety data transfer. Our second research track concerns itself with preserving location privacy of vehicles since PKI does not afford privacy. To avoid the available privacy preservation solutions’ covering-encryption overhead and silent-periods’ lack of communication, we propose the idea of making vehicles create dynamic mix zones using an alternative super anonymous authentication scheme to hide their pseudonym change. Our third contribution falls within the augmentation of efficiency of communication when safety beacons collisions arise due to limited medium, CSMA/CA access mode, and PKI beached overhead. In this regard, we use the concept of grouping and overhead reduction to lower the vehicles’ competition for the channel. Rather than having many individual vehicles
communicate their information to the infrastructure, group leaders become main figures of communication. Our fourth work focuses on building an efficient identity based alternative authentication for VANETs other than PKI with the goal of having less communication overhead. Our built framework has fast computations, no elliptic curves pairings, smaller communication overhead, and more anonymous usage of pseudo identities to achieve the needed privacy. Focusing on the efficiency aspect of vehicular communication, in the fifth exerted effort, rather than using only PKI to authenticate users, we introduce a context aware authentication interchange protocol to match the situational neighborhood conditions of vehicles. If it is a dense network, our scheme switches to use a lower overhead authentication scheme; if it is a sparse network, the vehicle automatically switches to a more anonymous authentication.

In a nutshell, the domain of VANETs offers a unique set of challenges; yet they present immense opportunities for research. We address three major challenges and suggested five research directions that may help in overcoming these limitations. We hope through these tracks of research to cast a light on the suitability of new concepts in affording the security, privacy, and availability of VANETs communications while achieving a comparable performance to the already adopted schemes.

**Keywords:** Vehicular Networks (VANETs); Intelligent Transportation Systems (ITS); Cyber-Physical Systems (CPS); Public Key Infrastructure (PKI); Security; Privacy; Cryptography; Somewhat Homomorphic Encryption (SHE); Elliptic Curves Cryptography (ECC); Pairings; Group Signatures; ID-base Signature (IBS); Trust; Authentication; Conditional Privacy Preserving Authentication (CPPA) Schemes; Platooning; Convoying; Grouping; Collective Awareness; Dedicated Short-Range Communication (DSRC); Beaconing Rate
To my dear parents .. ❤️
Acknowledgments

First and foremost I must thank The Holy, The Guide, The Light, The Most High Allah, for giving me the opportunity and strength to persevere and complete this degree. I express my gratitude to my advisor Prof. Füsun Ö zgüner for her support and guidance. She has been a flexible welcoming caring spirit in various ways. I would like to express a deep appreciation to all professors and colleagues whom I worked with, met with, and gained knowledge from. A sincere gratitude is extended to the Bureau of Educational and Cultural Affairs of the United States / Department of State, for the Fulbright grant that is administered by the Institute of International Education. This scholarship gave me the opportunity to come to the US and pursue my studies. I also thank the US Department of Transportation, University Transportation Center (UTC) at The Ohio State University and the National Science Foundation (NSF) for funding and supporting my research under grant numbers DTRT13-G-UTC47 and CNS-1446735, respectively. I cannot end without thanking my parents, my family, and my true friends on whose constant support and love I have relied. I extend a deep appreciation for the beautiful memories I have had with my second family and my community here in Columbus at Masjid Omar Ibn El-Khattab. May Allah join us in Paradise.
Vita

February 13, 1982 . . . . . . . . . . . . . . . . . . . . . . . . . . Born - Basra, Iraq

2004 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . B.S. Computer Engineering, Basra University

2008 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . M.S. Computer Engineering, Basra University

2016 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . M.S. Electrical and Computer Engineering, The Ohio State University

2012-2017 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Ph.D. Student, Electrical and Computer Engineering, The Ohio State University.

Publications

Research Publications


Fields of Study

Major Field: Electrical and Computer Engineering

Studies in:

- Topic 1  Vehicular Networks and Cyber Physical Systems
- Topic 2  Security, Privacy, and Efficiency of Communication
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>v</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>vi</td>
</tr>
<tr>
<td>Vita</td>
<td>vii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xiii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xv</td>
</tr>
<tr>
<td>Glossary</td>
<td>xviii</td>
</tr>
<tr>
<td>Nomenclature</td>
<td>xxiii</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Vehicular Adhoc Networks and Dedicated Short Range Communications</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Problems in Focus and Motivations</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Contributions and Goals</td>
<td>4</td>
</tr>
<tr>
<td>1.3.1 A Low-Level Alternative to ECIES Encryption for Non-safety Sensitive Applications using Partial Homomorphic Encryption</td>
<td>5</td>
</tr>
<tr>
<td>1.3.2 A Location Privacy Preservation Scheme using A Low-Level Alternative Group Signature (GS) Authentication and A High-Level Concept of Dynamic Mixing Zones</td>
<td>6</td>
</tr>
<tr>
<td>1.3.3 A High-Level Beaconing Efficiency Enhancement using The Concept of Grouping and PKI Overhead Omission</td>
<td>7</td>
</tr>
</tbody>
</table>
1.3.4 A High-Level Conditional Privacy Preservation and Overhead Reduction using Alternative Identity Based Cryptography (IBC) Authentication .............................................. 7
1.3.5 A Beaconing Efficiency and Privacy Preservation using A Low-Level Context-Aware Authentication Interchange Scheme and A High-Level Concept of Clustering ........................................... 8
1.4 Dissertation Structure ................................................................. 9

2. Background and Literature Review .................................................. 10

2.1 Security Requirements of Safety Application Layer ............................ 12
2.1.1 Vehicular Public Key Infrastructure ........................................ 14
2.1.2 PKI’s Lack of Privacy and Available Solutions .......................... 16
2.2 Confidentiality Requirement of Non-Safety Application Layer ............. 19
2.3 Vehicular Communication Efficiency Requirements .......................... 23
2.3.1 Countermeasures To Overcome Inefficient Beaconing ................. 25
2.3.2 Identity Based Authentication (IBC) Schemes: Review ............... 27

3. Augmenting The Confidentiality of Sensitive Non-safety Applications Through SHE Encryption and Secure Pairwise Key Establishment .................. 32

3.1 Model Assumptions and Preliminary BGN-Blom Construction ............ 33
3.1.1 Limitation of Original Model .............................................. 40
3.2 Less Complex and More Efficient Alternatives ................................ 41
3.3 Functional Correctness Proofs .................................................... 43
3.3.1 Adversary Model and Security Fulfillment ............................... 44
3.4 Comparisons and Analytical Results Discussion ............................ 45
3.4.1 Key Storage Overhead and Key Size (bit) ............................... 46
3.4.2 Computation Complexity and Estimated Time (sec) .................... 47
3.4.3 Communication Overhead (byte) ......................................... 51

4. Preserving Location Privacy Using An Anonymous Authentication Dynamic Mixing Crowd .................................................. 54

4.1 Network Model and Assumptions ............................................... 55
4.2 Proposed Mixing Crowd Mechanism ......................................... 56
4.2.1 Modifying The Created Crowd ............................................ 58
4.3 Considered Attacker, Model Validity, and Correctness Proofs ............ 61
4.4 Numerical Analysis and Discussion of Effectiveness ........................ 63
4.4.1 Computation Time (msec) ............................................... 64
4.4.2 Storage Overhead (byte) .................................................. 66
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Motivations and Contributions Outline</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Safety Beacon Size and Cryptographic Overhead (bytes)</td>
<td>24</td>
</tr>
<tr>
<td>3.1 Storage Overhead $S_O$ per Key and Key Size: Comparison</td>
<td>49</td>
</tr>
<tr>
<td>3.2 Used Cryptographic Operations and Their Complexity over $E$:</td>
<td>49</td>
</tr>
<tr>
<td>$(\hat{a} \text{ and } \hat{b} \in \mathbb{Z}_q^*)$ and $(\hat{P} \text{ and } \hat{Q} \in G_1)$</td>
<td></td>
</tr>
<tr>
<td>3.3 Maximum Element Size in $\Psi$ and $C_C$ per Key: Comparison</td>
<td>50</td>
</tr>
<tr>
<td>4.1 Used Cryptographic Operations and Their Complexity over $E$:</td>
<td>64</td>
</tr>
<tr>
<td>$(\hat{a} \text{ and } \hat{b} \in \mathbb{Z}_q^*)$ and $(\hat{P} \in G)$</td>
<td></td>
</tr>
<tr>
<td>4.2 ECDSA vs. GS [1]: Performance and Key Lengths</td>
<td>64</td>
</tr>
<tr>
<td>4.3 Computations Comparison Inside And Outside The Zone With And</td>
<td>65</td>
</tr>
<tr>
<td>Without Optimization</td>
<td></td>
</tr>
<tr>
<td>4.4 Schemes’ Storage Cost $S_O$ Comparison</td>
<td>66</td>
</tr>
<tr>
<td>5.1 Exchanged Beacons And Communication in Our Scheme With Several</td>
<td>82</td>
</tr>
<tr>
<td>Authentication’s Omission Types</td>
<td></td>
</tr>
<tr>
<td>5.2 G-BEE Security Fulfillment Proofs</td>
<td>84</td>
</tr>
<tr>
<td>5.3 G-BEE Security Fulfillment Proofs (continued 1)</td>
<td>85</td>
</tr>
<tr>
<td>5.4 G-BEE Security Fulfillment Proofs (continued 2)</td>
<td>86</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Typical Vehicular Network Model</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>DSRC Channels and Standards</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Dividing Synchronization Interval into CCH and SCH Periods [7]</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>ECIES for Confidentiality in Vehicular PKI</td>
<td>21</td>
</tr>
<tr>
<td>2.3</td>
<td>IBC Authentication Schemes Classification</td>
<td>28</td>
</tr>
<tr>
<td>3.1</td>
<td>Preliminary BGN-Blom Session Key Generation</td>
<td>38</td>
</tr>
<tr>
<td>3.2</td>
<td>BGN-Blom Key for PKI Authentication</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>Storage Overhead $S_O$ per Generated Key: Comparison</td>
<td>48</td>
</tr>
<tr>
<td>3.4</td>
<td>Key Computation Comparison</td>
<td>52</td>
</tr>
<tr>
<td>4.1</td>
<td>The Proposed Scheme Timeline for Three Vehicles: $v$, $w$, and $u$</td>
<td>59</td>
</tr>
<tr>
<td>4.2</td>
<td>The Timeline After Eliminating The Group Signature Phase</td>
<td>60</td>
</tr>
<tr>
<td>4.3</td>
<td>Schemes Comparison Before Grouping (No Freezing Delay)</td>
<td>68</td>
</tr>
<tr>
<td>4.4</td>
<td>Schemes Comparison After Grouping (No Freezing Delay)</td>
<td>69</td>
</tr>
<tr>
<td>5.1</td>
<td>System Model</td>
<td>71</td>
</tr>
<tr>
<td>5.2</td>
<td>Grouping Scenario</td>
<td>75</td>
</tr>
</tbody>
</table>
5.3 After Group Formation and Several Broadcasts: CMV of members is updated because of the received leaders’ beacons 77
5.4 Addressed G-BEE Scenario 80
5.5 One Dimensional Bianchi Markov Model for CSMA/CA 88
5.6 D/M/1 Model for Each Vehicle’s Beacons Buffer 89
5.7 D/M/1/1 Model for Each Vehicle’s Beacons Buffer 91
5.8 Delay Rate $\bar{d}$, Collision Rate $\bar{\xi}$, and Drop Rate $\bar{\gamma}$ Metrics Comparison for Individual No Grouping Scenario, Simple G-BEE, and Enhanced G-BEE for $N_L = 6$ Evenly Formed Groups 96
5.9 Throughput Metric Comparison for Individual Case, Simple G-BEE, and Enhanced G-BEE w.r.t. Number of Vehicles $n$ and $N_L = 6$ clusters/Leaders 98
5.10 Throughput Metric Comparison for Individual Case, Simple G-BEE, and Enhanced G-BEE w.r.t. Number of Leaders/clusters $N_L$ and a Network of $n = 64$ Vehicles 99
5.11 Drop Rate $\bar{\gamma}$ Metric Comparison for Individual Case, Simple G-BEE, and Enhanced G-BEE w.r.t. Data Transmission Rate $r$ for $n = 64$ and $N_L = 6$ Leaders 100
5.12 Drop Rate $\bar{\gamma}$ Metric Comparison for Individual Case, Simple G-BEE, and Enhanced G-BEE w.r.t. Beacon Generation Period $\mathcal{T}$ for $n = 64$ and $N_L = 6$ Leaders 100
6.1 PME-CPPAS Network Model 109
6.2 EF-CMA’s Real Attack of $\mathcal{AT}$ 115
6.3 RS Allows $\mathcal{AT}$ To Replay RS’s Oracles To Solve ECDLP Problem and Extract $\hat{x}$ 120
6.4 A Comparison of Signature Generation (Extraction and Signing) Phase 128
6.5 Single Verification Phase Comparison 129
Glossary

\( \mathcal{AT} \) Attacker.

\( \mathcal{RS} \) Reduction Simulator.

\( \mathcal{SC} \) Signer/Challenger.

**AES** Advanced Encryption Standard.

**BGN** Boneh-Goh-Nissim.

**BLS** Boneh-Lynn-Shacham signature.

**BSM** Basic Safety Message.

**CA** Central/Certification Authority.

**CAAIS** Context-Aware Authentication Interchange Scheme.

**CCH** Control Channel.

**CCM** Counter with CBC-MAC.

**CMA** Adaptive Chosen Message Attacker.

**CPPA** Conditional Privacy Preserving Authentication.
CPS  Cyber Physical Systems.

CRL  Certificate Revocation List.

CSMA/CA  Carrier Sense Medium Access with Collision Avoidance.

D/M/1  (Deterministic Arrival / Exponential Service Time / Single Server) Queue.

D/M/1/1  (Deterministic Arrival / Exponential Service Time / Single Server / No Queue) Model.

DH  Diffie-Hellman.

DIFS  Distributed Coordination Function (DCF) Interframe Space.

DLP  Discrete Logarithm Problem.

DNF  Disjunctive Normal Form.

DoT  US Department of Transportation.

DSRC  Dedicated Short Range Communication.

ECC  Elliptic Curve Cryptography.

ECDLP  Elliptic Curves DLP.

ECDSA  Elliptic Curve Digital Signature Algorithm.

ECIES  Elliptic Curve Integrated Encryption Scheme.

EF  Existential Forgeability.

EFHE  Efficient FHE.
FCC  Federal Communications Commission.

FHE  Fully HE.

G-BEE  Grouping for Beaconing Efficiency Enhancement.

GM  Group Manager.

GPS  Global Positioning System.

GS  Group Signatures.

GSIS  GS and Identity-based Signature.

HE  Homomorphic Encryption.

I2V  Infrastructure-to-Vehicle.

IBC  Identity Based Cryptography.

IBS  Identity-Based Signature.

IEEE  Institute of Electrical and Electronics Engineers.

IETF  Internet Engineering Task Force.

ITS  Intelligent Transportation Systems.

KDF  Key Derivation Function.

M/G/1  (Markovian Arrival / General Service Time / Single Server) Queue.

MAC  Medium Access Control.
MNT  Miyaji-Nakabayashi-Takano.

NTRU  Number Theory Research Unit.

OBU  On Board Unit.

PKG  Public Key Generator.

PKI  Public Key Infrastructure.

PME-CPPAS  Pairingless Modified Efficient CPPA Scheme.

REP  Random Encryption Periods.

RL  Revocation List.

RSU  Road Side Unit.

SAE  Society of Automotive Engineers.

SCH  Service Channel.

SDA  Subgroup Decision Assumption.

SHA  Secure Hash Algorithm.

SHE  Somewhat HE.

SLOW  Silence at LOw Speeds.

TA  Trusted Authority.

TPD  Tamper Proof Device.
**TPM** Trusted Platform Module.

**V2I** Vehicle-to-Infrastructure.

**V2V** Vehicle-to-Vehicle.

**VANETs** Vehicle Adhoc Networks.

**WAVE** Wireless Access in Vehicular Environments.

**WSA** WAVE Service Advertisement.

**WSMP** Wave Short Message Protocol.
Nomenclature

$(m^{index}, h_{1-5}^{index}, \text{sign}^{index})$ List of $\mathcal{AT}$ queries to $\mathcal{RS}$

$\hat{c} \in Z_n^*$ small integer for the small exponent test

$\hat{P}_{BGN}$ Point generator of $G_2$ in $BGN$ of order $\bar{n}$

$\alpha$ Vector of $n$ small scalars $alpha^i \in Z_n^*$ for small exponent test

$\bar{b}_i$ Stationary probability of state $i$

$\bar{I}$ Idle state when vehicle has no beacon to send

$\bar{n}$ Order of $G_1$ and $G_2$ on $E_{BGN}$; Number of elements of the two groups; $\bar{n} = q_1.q_2;\bar{n} = 2^{1024}$

$\bar{r}$ Integer $\in Z_{\bar{n}}^*$

$\bar{t}$ Trace of Frobenius

$\bar{v}$ Number of beacons in vehicle buffer

$\bar{r}_{i,j}$ Random Entries $\in F_p$ of $\Upsilon$ in Blom

$\bar{\gamma}_v, \bar{\gamma}$ Beacon drop rate

$\bar{\xi}_v, \bar{\xi}$ Beacon collision rate
\( \bar{d}_v, \bar{d} \) Average beacon delay

\( \beta \) Timeslot (sec)

\( \Delta v_{\text{info}} \) Information relative to vehicle \( v \) current status: location difference, speed difference, acceleration difference, etc

\( \delta \) Propagation delay (sec)

\( \dot{\Psi} \) Remodeled \( \Psi \) matrix

\( \ell \) Positive Integer \( \in \mathbb{Z}^+ \)

\( \gamma_c \) Number of vehicles that cannot capture the medium

\( \lambda \) Maximum Tolerated Capturing Threshold in Blom

\( \bar{O} \) Point of infinity on \( E \)

\( \xi_{A_V} \) Threshold of \( A_V \)

\( \xi_{C_O} \) Threshold of \( C_O \)

\( \xi_{N_C} \) Threshold of \( N_C \)

\( \xi_{N_L} \) Threshold of \( N_L \)

\( \xi_{N_M} \) Threshold of \( N_M \)

\( A_V \) Acquired degree of anonymity

\( C_C \) Computation Complexity (bit)

\( C_O \) Communication Overhead (byte)
\( \mathcal{E}_c \)  
Expected value of backoff counter

\( N_c \)  
Number of received beacons

\( \mathcal{N} \)  
Number of Objects

\( \mathcal{O} \)  
Big O complexity notation

\( S_O \)  
Storage Overhead (byte)

\( T_{c_v} \)  
Context Timer at \( v \)

\( T \)  
Beacon generation rate (sec)

\( N_c \)  
Number of CL vehicles

\( N_M \)  
Number of CM vehicles

\( \mu \)  
Exponential beacon service rate

\( \omega \)  
Laplace - Stieltjes transform of \( \Pi_k \)

\( \Omega_\ast \)  
Random value from \( \mathcal{A}\mathcal{T} \)

\( \oplus \)  
XOR operation

\( \phi \)  
Empty

\( \Pi_i \)  
Stationary probability of each state \( \bar{v} \)

\( \Psi \)  
Public matrix in Blom of \( (\lambda + 1) \times n \) elements

\( \psi \)  
Generator \( \in F_{P\text{Blom}} \) of \( \Psi \) in Blom

\( \sigma_v \)  
Long term scalar signature of \( v \)
Scalar signatures $\sigma^i_v$, $e^i_v$ of $v$ when $i^{th}$ instance of keys are used

Limited time period of $AT$

Symmetric secret matrix of random $\bar{r}_{i,j}$ entries in Blom

"distortion map/rotation function on elliptic curves"

Embedding on elliptic curves

Binary string of any bit length

Binary string of an $L$ bit length

$a, b \in F_p$ large numbers that satisfy $16(4a^3 + 27b^2) \neq 0, 3 \mod p$

Element $k \in \{1, ..., (\lambda + 1)\}$ of row $v$ of $(\Upsilon, \Psi)^T$ Blom matrix at vehicle $v$

Element $k \in \{1, ..., (\lambda + 1)\}$ of column $v$ of $\Psi$ Blom matrix at vehicle $w$

Ciphertext

Ciphered identity of $I$, $v$, and $w$

Ciphered of $A_v,k$

$k^{th}$ element of resulted ciphertext after pairing $C_{A_v,k}$ and $C_{B_k,w}$

Ciphered of $B_{k,w}$

BGN-Blom created encrypted key

Ciphered of $m^i_v$

Certificate from CA
$cer_{CA}(pk^i_v)_\text{serial}$ Serial number of $cer_{CA}(pk^i_v)$

**CMV** Cluster Members Vector

**create** - **ack** Neighboring vehicles respond with acknowledgement

**create** - **cover** $v$ requests the authentication cover of $duration$ lifetime

**create** - **group** Initiating vehicle broadcasts a request to create a group

**create** - **session** Session creation request by $v$

**CRL_{generation\ date}** CRL generation date

**CS** Cluster Size

**Dlog()** Discrete logarithm function

$E$ Elliptic curve with large parameters where $E : y^2 \equiv (x^3 + ax + b) \mod p \cup \{\hat{0}\}$

$e$ pairing function on elliptic curves

$E_{BGN}$ Super singular elliptic curve with $a = 1$, $b = 0$, $x$, $y$, all $\in F_{pBGN}$

$\text{enc}(\ ), \text{dec}(\ )$ Encryption and decryption functions

**end** - **cover** $v$ requests to demolish the mixing crowd

**end** - **session** End session message from $v$ or $I$

**extend** - **cover** $v$ asks to extend the mixing crowd lifetime by $duration$ period

$f$ Function such as $+$ and $\times$

$F_{pBim}$ Galois Field over $p_{Bim}$ in Blom’s

xxvii
$G, \tilde{G}_1$ Group of points on $E$ with order $q$ and generator $P$

$G_1$ Group of points on $E_{BGN}$ with order $\bar{n}$ and generators $P_{BGN}$ and $\hat{P}_{BGN}$

$G_2$ Group of numbers in extension field of elliptic curve in BGN with order $\bar{n}$ and generator $\hat{P}_{BGN}$

$G_T, \tilde{G}_2$ Group of numbers of a finite extension field $F_{p^\rho}$

$g_{pk_{GM}}$ GM public key $\in G$

$g_{sk_v}$ Vehicle $v$’s secret key within the group $\in \mathbb{Z}_q^*$

$g_{sk_{GM}}$ GM secret key $\in \mathbb{Z}_q^*$

$GPG$ Channel gain (Group Power Gain) between any vehicle and the leader of its group

$h_i^i - h_5^i$ $i^{th}$ query to $h_1$-$h_5$ oracles

$h_{1,2,3,4,5}()$ Collision-resistant one-way secure hash function that maps to $\mathbb{Z}_q^*$

$HP$ Hash to Point

$HS$ Hash to Scalar

$I$ Infrastructure

$ID_v$ Vehicle’s real identity $\in G$

$ID_v^{track}, ID_v^{i\text{track}}$ Vehicle tracking identity $\in G$

$ID_{CA}$ Certification authority identity $\in G$
\( \text{join} – \text{ack} \) Vehicle \( v \) accepts the \( \text{join} – \text{cover} \) request made by \( z \)

\( \text{join} – \text{cover} \) Vehicle \( z \) wants to join the created mixing crowd

\( \text{join} – \text{group} \) Request to join an already created group

\( k_v, k_m \) Keys in ECIES

\( L \) Security Level/Strength (bit)

\( L_{h_1} – L_{h_5} \) Lists represent \( h_1-h_5 \) oracles

\( m_v \) Message of \( v \)

\( m_v^i \) Message of \( v \) when \( i^{th} \) instance of keys are used

\( MA \) Modular Addition

\( ME \) Modular Exponentiation

\( MI \) Modular Inversion

\( MM \) Modular Multiplication

\( mod \ q \) Modulus operation

\( n \) Network size; Number of vehicles; Largest identity of a vehicle in Blom’s

\( NOT \) Negation

\( P \) Point generator of \( G = E \) of order \( q \)

\( p, q \) Large prime numbers of bit length equals \( L \) and \( 2L \), respectively

\( p_{BGN} \) Large prime number of BGN (\( p_{BGN} = \ell.n – 1 \equiv 2 \mod 3 \)
$P_{BGN}, \dot{P}_{BGN}$ Point generators of $G_1$ in $BGN$ of order $\bar{n}$

$p_{Blom}$ Prime number $\geq 3$ in Blom’s

$PA$ Point Addition

$PD$ Pollard Lambda Decryption

$pg^w_v$ Signal strength (Power Gain) between $v$ and $w$

$pk^i_v \in G$ $i^{th}$ random public key of $v$

$pk_{BGN}$ Public key of $BGN$ equals $(\bar{n}, G_1, G_2, e, P_{BGN}, \dot{P}_{BGN})$

$pk_{CA/TA/PKG}, \dot{P}$ Master public key of certificate/trusted/PKG authority ($\in G$)

$pk^{i}_{IBC_v} \in G$ $i^{th}$ random public key of $v$ for IBC authentication

$Pr()$, $Pr_i$, $Pr_{i,j}$ Probability Function and Value

$PWD_v$ String password $\{0,1\}^*$ of $v$

$q_1, q_2$ Large prime numbers of bit length equals $\frac{L}{2}$

$q_{h_1} - q_{h_5}$, $q_s$ Queries to $h_1$-$h_5$ and sign oracles

$R$ Beacon Size (byte)

$r$ Data transfer rate (Mbps)

$R_{Certificate}$ Size of certificate

$R_{Hash}$ Size of hash

$R_H$ Size of all headers in beacon

xxx
$R_{\text{signature}}$ Size of signature

$\text{revocation}_\text{date}$ Revocation date of certificate

$S_{v,w} = S_{w,v}$ Secret session key $\in G$ of PME-CPPAS

$s_{v,w} = s_{w,v}$ The x-coordinate of $S_{v,w} = S_{w,v}$ in PME-CPPAS

$\text{session}_\text{index}$ Current pairwise session index

$\text{sid}^1_v, \text{sid}^2_v, \text{sid}_v, H^i_v$ Alias keys of $v$

$\text{SID}^{i,1}_v, \text{SID}^{i,2}_v, \text{SID}^i_v, h\text{SID}^i_v$ Alias identities of $v$

$\text{sig()}$ Signature

$sk_v \in Z^*_q, pk_v \in G$ Long-term scalar secret and point public keys given to vehicle $v$

$sk^i_v \in Z^*_q$ $i^{th}$ random secret key of $v$

$sk_{BGN}$ Secret key of BGN equals $q_1$

$sk_{\text{CA/TA/PKG}, x}$ Master private key of certificate/trusted/PKG authority ($\in Z^*_q$)

$sk_{\text{IBC}, v} \in Z^*_q, pk_{\text{IBC}, v} \in G$ Long-term scalar secret and point public keys given to vehicle $v$ for IBC authentication

$sk^{i}_{\text{IBC}, v} \in Z^*_q$ $i^{th}$ random secret key of $v$ for IBC authentication

$\text{SM}$ Scalar Multiplication

$\text{SMA}$ Small Modular Addition

$\text{SMM}$ Small Modular Multiplication
SSM  Small Scalar Multiplication

t  small integer; security parameter of small exponent test

t_b  Average backoff time

t_f  Average freezing time

t_r  Average transmission time

T_v  Timestamp of v

T_v^i, t t_v^i  Timestamp of v when i^{th} instance of keys are used

Threshold_{joining/leaving/permission}  Minimum thresholds on signal strength for group modification

Tran  Matrix Transpose in Blom

v, w, u, z K, U, J, Z, O, V  Unique vehicle chassis identity (numerical \( \in \mathbb{Z} \) or string \( \in \{0,1\}^* \))

v_{info}  Vehicle v current status: location, speed, acceleration, direction, etc

W_0  Contention Window

W_u  Up-to-date throughput

W_v  Throughput; V2I throughput

W_{min}  Lowest achievable throughput

WP  Weil Pairing
$x, y \in F_p$ coordinates of a point on $E$

$\hat{k}^i, \hat{d}^i, \hat{b}^i$ Random private scalar keys $\in Z_q^*$
Chapter 1: Introduction

1.1 Vehicular Adhoc Networks and Dedicated Short Range Communications

The dramatic changes in vehicles’ underlying control systems intermingled with the added computing, sensing, and communication abilities have been contributing in the emergence of a new era of Intelligent Transportation Systems (ITS). In such new Vehicular Adhoc Networks (VANETs) era, autonomous vehicles coordinate among themselves (Vehicle-to-Vehicle (V2V) communication) as well as with the infrastructure (Vehicle-to-Infrastructure (V2I) communication) to provide a cooperative awareness that helps avoiding accidents, alleviating traffic congestion, as well as infotaining drivers. A typical VANETs system model is shown in Fig. 1.1; each vehicle is given an On-Board Unit (OBU) to store any given parameters at time of registration with a Trusted Authority TA. Road Side Units (RSU)s represent the infrastructure with which the vehicles will communicate after deployment. For the VANETs communication aspect, the United States Federal Communications Commission (FCC) dedicated a 75 MHz spectrum bandwidth named Dedicated Short Range Communications (DSRC) at the 5.85 – 5.925 GHz band; DSRC is structured as six 10 MHz Service Channels (SCH), one 10 MHz Control Channel (CCH), and 5 MHz is reserved
as guard band, see Fig. 1.2a [7]. These different channels are envisioned to support diverse vehicular applications that are mainly categorized as safety and non-safety applications\(^1\); basically, out of the seven mentioned DSRC channels, only the CCH is dedicated for the broadcasting of the high-priority safety messages and the remaining six SCH channels are used for non-safety purposes. Whether a vehicle is communicating over CCH for safety applications or over SCH for non-safety purposes, it needs to correctly and efficiently converse with its neighbors and with the infrastructure with no alteration or drop of the communicated data. To achieve this correctness and efficiency, DSRC glues together several standards adopted from other networks as depicted in Fig. 1.2b; however, not all of the embraced standards suit the nature of vehicular communications, i.e., they exhibit some shortcomings when being applied in VANETs; especially, the standards that are embraced to afford authentication such as PKI, to secure the sensitive data such as Elliptic Curve Integrated Encryption Scheme (ECIES), and to provide wireless access mode in the Medium Access Control (MAC) layer such as Carrier Sense Multiple Access - Collision Avoidance (CSMA/CA).

\(^1\)For a complete list of sought vehicular applications, see [8] that is defined by US Department of Transportation (DoT).
1.2 Problems in Focus and Motivations

The shortcomings, which we are concerned with lie within the confidentiality, privacy, and efficiency requirements of VANETs. In terms of confidentiality, the adopted ECIES encryption does not provide authentication on its own, i.e., it has to have the PKI authentication established beforehand to be able to openly exchange the encrypted data afterwards (Section 2.2 of Chapter 2 expands on this topic). For the authentication aspect, the adopted PKI authentication has two major deficiencies. First, PKI does not provide user privacy/anonymity on its own such that any eavesdropper could easily link the used certified public key with the vehicle’s identity and can track vehicles. Second, PKI has a large security overhead attached with every broadcasted beacon. This puts a heavy load on the wireless access, especially due to the limited CCH channel width. Finally, in terms of efficiency, since CSMA/CA is used in a broadcast setting, to avoid flooding the feedback channels, acknowledgements and handshaking cannot be utilized. This means that there will be no collision
detection in broadcast-CSMA/CA making the beacon collision rate and access delay increase with the number of contending vehicles. Also, the addition of PKI payload to beacons will definitely worsen the beacon drop and collision rates in dense VANETs scenarios; as a result, the total number of vehicles that can concurrently capture the wireless medium will be decreased. These three particular shortcomings have helped research to flourish in both academia and industry as is exhibited in Chapter 2. Likewise, these shortcomings have motivated us to contribute with our own solutions. Our contributions focus on finding solutions to acclimate with the available DSRC shortcomings and disregarding the bias that DSRC has towards using only specific standards by bringing other alternative frameworks into the scene. With the first direction in mind, our efforts are a mixture of high-level re-arrangement protocols such as grouping and overhead omissions to acclimate with the PKI and CSMA/CA privacy and efficiency limitations. For the other direction, since some frameworks have less communication overhead while others have more anonymous traits, we have attempted low-level alternatives to ECIES and PKI to overcome their confidentiality, privacy, and efficiency limitations. Table 1.1 clearly identifies and categorizes our motivations and contributions.

1.3 Contributions and Goals

The coarse-grained description of our five contributions, which we have just touched upon, is introduced here. We will recurrently use the terms low-level and high-level when we describe the introduced schemes. The low-level term refers to the underlying cryptographic primitives while the high-level term refers to the arranging of protocol/framework/scenario:
1.3.1 A Low-Level Alternative to ECIES Encryption for Non-safety Sensitive Applications using Partial Homomorphic Encryption

The DSRC’s 1609.2 standard suggests the ECIES hybrid encryption for securing sensitive non-safety applications such as payment services. Although ECIES has short keys and fast computations, it does not offer any privacy for the involved parties and it needs a preliminary step of PKI authentication. Furthermore, the ECIES use of private keys incurs the need for a hardware security via the Tamper Proof Device (TPD) in each vehicle. Therefore, in case no external authentication is available, to add the overlooked handshaking’s privacy to the confidentiality of transmission and to eradicate the need for expensive hardware security of OBUs, we presented in [9] an alternative encoding scheme making use of a new state-of-the-art low-level Somewhat Homomorphic Encryption (SHE) concept [10]. Our contribution exchanges the involved vehicles’ identities in an obfuscated manner before dynamically generating
the final veiled keys to be used for confidentializing the high-level data transfer; in this way the obvious benefit of our model would be in letting no outsider obtain any information since all of the transfers and storage are encrypted and the dynamic establishment of one to one mutual authentication. Full description of the scheme, proofs, and obtained results are included in Chapter 3.

1.3.2 A Location Privacy Preservation Scheme using A Low-Level Alternative Group Signature (GS) Authentication and A High-Level Concept of Dynamic Mixing Zones

Preserving location-privacy in vehicular networks is a challenge. On one hand, under PKI realm, changing pseudonyms has been the state-of-the-art as we will see in Chapter 2; however, to avoid being tracked, vehicles have to change their pseudonym when they are within fixed or dynamic mix zones, created using encryption or silent periods. On the other hand, under the Group Signatures (GS) authentication realm, super anonymity is afforded for vehicles but at the expense of slow signature verification. To benefit from the super anonymity of an optimized hybrid group signature authentication, we suggest the idea of altering authentication to form the dynamic mix crowd for an originating vehicle that needs to change its pseudonym [11]. Our scheme has neither silent periods nor encryption; it simply utilizes the new authentication as a dynamic hiding cover for the pseudonyms changing; once the pseudonym is altered, the vehicle watches for at least one other change by any cooperative neighbor in the formed zone before it automatically demolishes the group by reverting back to the baseline PKI authentication. Full description of our contribution is presented in Chapter 4 where we show the easiness of forming and demolishing the dynamic
groups. We prove the validity of the model by satisfying the needed vehicular security requirements; we show that our scheme has acceptable performance in comparison with the baseline authentication.

1.3.3 A High-Level Beaconing Efficiency Enhancement using The Concept of Grouping and PKI Overhead Omission

The payload of each beacon is sizable due to the large overhead of PKI authentication, the bandwidth is limited in VANETs, and CSMA/CA is typically highly inefficient in dense scenarios where a large number of users compete for the medium. We use this insight to develop a new cooperative beaconing strategy, Grouping for Beaconing Efficiency Enhancement (G-BEE) [12]. In G-BEE, vehicles dynamically form secure authenticated groups with their leadership roles being decided in a decentralized fashion. In this way, the main load of V2I beaconing is assigned to group leaders. This transforms the problem of medium access from a dense network of short sessions into one with a sparse network of longer sessions, the ideal setting for CSMA/CA. In order to not compromise the security of the created groups, we introduce a simple and an enhanced version of G-BEE to tackle two levels of authentication-overhead omission/reduction. To investigate the gain of our contribution, we build three stochastic analytical models. In Chapter 5, we describe G-BEE and exhibit the conducted numerical evaluations that support the effectiveness claim of our approach.

1.3.4 A High-Level Conditional Privacy Preservation and Overhead Reduction using Alternative Identity Based Cryptography (IBC) Authentication

To overcome PKI overhead while affording needed privacy, Identity Based Cryptography (IBC) authentication schemes have been introduced over the last years as a
powerful alternative competitor. Most of these IBC approaches rely on pairing operation on elliptic curves in their underlined calculations. Since pairings is a complex and time-consuming operation, the gears are shifted towards building pairingless IBC authentications; both incorporate the use of the hardware security of TPD to secure the cryptographic parameters at any end-user. Motivated by this trend, we devise in [6] a new identity based authentication scheme called Pairingless Modified Efficient Conditional Privacy Preserving Authentication Scheme (PME-CPPAS). Our contribution eliminates the use of the complex pairings operation as well as the need for expensive TPDs. Chapter 6 presents PME-CPPAS and analyzes its performance. We show the gain behind PME-CPPAS in comparison with the available pairing-based and pairingless approaches in terms of computation complexity and communication overhead.

1.3.5 A Beaconing Efficiency and Privacy Preservation using A Low-Level Context-Aware Authentication Interchange Scheme and A High-Level Concept of Clustering

Once more, to have a private efficient VANETs, we introduce an authentication interchange scheme by which we intermingle between three frameworks: PKI, GS, and a new IBC authentication scheme that we specifically develop to serve our needs in this contribution. Our Context-Aware Authentication Interchange Scheme (CAAIS) scheme is context-aware such that every specific time interval, the PKI vehicle checks its context; if the vehicle encounters a crowded neighborhood, it switches to a lower-overhead IBC authentication. If the car’s context is sparse, it alters to a more anonymous GS framework. Chapter 7 outlines the contribution [13] and proves its security
using a valid argument we developed via the strategies of attacker games and forking lemmas. In terms of the defined performance metrics and the two suggested communication scenarios, the interchanged CAAIS is shown to achieve better performance than the case of having one secure and totally pure authentication throughout the network lifetime.

1.4 Dissertation Structure

The rest of this dissertation is organized as follows. In Chapter 2, we introduce a wealthy background for the problems we mentioned and what has been done in literature to countermeasure these shortcomings. Chapter 3 presents our first contribution of tackling confidentiality of sensitive non-safety applications in VANETs. Chapter 4 describes the details of the second contribution of providing location privacy for vehicles using dynamic mix zones and alternative authentication schemes. Chapter 5 discusses the G-BEE contribution for efficiency and overhead reduction in VANETs under the concept of grouping to reduce beacon collision and drop under the CSMA/CA access mode. Chapter 6 delineates our fourth contribution of developing a pairingless IBC authentication scheme that preserves conditional privacy of vehicles and gives less overhead than PKI. Our final contribution is demonstrated in Chapter 7; the CAAIS is comprehensively described, proved to be secure, and shown to be effective in comparison with pure authentication frameworks. Finally, in Chapter 8, we sum up the outcomes of this dissertation. Also, we point out some future directions that can be followed based on our research contributions. Terminologies and cryptographic primitives are given in Appendix A.
Chapter 2: Background and Literature Review

DSRC has been going through several phases of standardization. On one hand, the IEEE 802.11 community introduced the IEEE 802.11p or Wireless Access in Vehicular Environments (WAVE) as an amendment to the original IEEE 802.11 standard to suit the high data transfer rates (3-27 Mbps) and the minimal latency needed by DSRC (operates over a short to medium range of 1 km); however, since 802.11p only describes the lower physical and MAC layers of the suggested protocol stack in Fig. 1.2b and how the communications take place over the different channels, the operational functionality and complexity of DSRC need to be handled by some additional upper layers. Therefore, the IEEE 1609 family of standards has been adopted to define how the applications will function in VANETs; for example, the 1609.3 is for the one-hop networking services including the Wave Short Message Protocol (WSMP) while the 1609.2 will be dedicated for security services. On the other hand, DSRC is decided to support the use of some well-known Internet protocols defined by Internet Engineering Task Force (IETF) for the network and transport layers in case the multi-hop broadcast is utilized by some applications. Additionally, the Society of Automotive Engineers (SAE) has contributed with its J2735 standard to define a dictionary of over fifteen possible message formats to use at the application layer;
specifically for the Basic Safety Message (BSM) type, the J2945.1 extension is introduced to tackle the related transmission rate, transmission power, and channel congestion control [7,14]. To define how vehicular applications function over the planned US DSRC 75 MHz spectrum, the 1609.4 standard has been adopted. As we mentioned in Chapter 1, basically in the US WAVE specifications, out of the seven mentioned DSRC channels, only the CCH would be dedicated for the broadcasting of the high-priority safety messages; besides, CCH is also used to broadcast some control WAVE Service Advertisement (WSA) message that announces about a service on the remaining six SCH channels. Therefore, all vehicles, which periodically beacon their status, have to rendezvous at the same CCH channel at the same time to be able to communicate with each other; while they are on CCH and hear a WSA message, they switch to the relevant SCH to use the announced non-safety service. To synchronize the switching, DSRC adopts a time division concept by which vehicles switch between the two channels. By synchronizing with the vehicle’s Global Positioning System (GPS) of a default 10 Hz location-update-rate, the 1609.4 MAC extension divides synchronization interval into two periods: CCH and SCH; each set to be a 50 msec including a 4 msec guard interval [7,14], as shown in Fig. 2.1 [7]. WAVE allows vehicles to remain on CCH channel as long as there is no WSA or the announced service is not relevant. Once the device needs to switch to an SCH, it either departs at the end of current CCH or directly when receiving the relevant WSA. When finishing with the service that might take single or multiple synchronization periods, the device has to wait until the beginning of the next CCH to switch back to safety beaconing [7,14].
The goal behind having these standards is to afford VANETs with a healthy communication via achieving some essential requirements; in what follows, we navigate through such requirements. Alongside, we show how DSRC promises to fulfill these needs. Since DSRC assertions are not precise, this chapter serves as a comprehensive literature review that constructs a map of the taken counteracts to endure the problems and shortcomings addressed in this dissertation.

### 2.1 Security Requirements of Safety Application Layer

The safety-oriented applications will provide vehicles with the cooperative-awareness needed to improve the quality of transportation by avoiding collisions and optimizing traffic conditions through alarm and warning messages. Therefore, safety-related applications mainly constitute V2V time-critical BSM messages that (i) include vehicle’s real-time information such as location, speed, lane, direction, and sending time and (ii) are periodically beaconed to demonstrate a real-time vehicle status to the neighborhood. Besides, occasionally some event-driven messages would be included within the safety messages to warn the receiving neighborhood about any encountered event.

To secure the safety applications in VANETs and to protect the communicated data
from being compromised or altered, research has offered many solutions. Whether being designed for the physical layer security or the application layer cyber-security, any solution has to fulfill the following requirements [15–18]:

- **Trustability and Authentication**: when vehicles communicate with each other or with the infrastructure, there must be a means to guarantee that the communicating parties are legitimate and belong to the same network [15–18].

- **Non-repudiation**: senders must not be able to deny sending messages [15–18].

- **Freshness and Integrity of the transferred data**: to ensure that the data has not been altered or replayed since it has been sent from the original sender [15–18].

- **Identity and Location Privacy**: with more secure networks, vehicles’ identities become less private; furthermore, even when keeping vehicles’ identities private, observers can still track vehicles’ locations disregarding any undertaken identity privacy measure [15–18].

- **Conditional Privacy and Traceability**: VANETs’ vehicles have to be untraceable by the public. Still, the trusted authorities, with which vehicles are registered, have to identify their members for accountability considerations [15–18].

- **Unlinkability**: the receivers must not be able to link the broadcasted beacons to each other and to their senders [15–18].

- **Credentials Revocation**: revocation of expired credentials of vehicles or of credentials of misbehaving vehicles has to be guaranteed [15–18].

To afford these needs at the application layer, out of three available proved-to-be comprehensively secure cyber-security systems: **PKI**, Group Signatures (GS),
and Identity-based Cryptography (IBC), DSRC 1609.2 standard has chosen the PKI framework [15–19] as explained in what follows.

### 2.1.1 Vehicular Public Key Infrastructure

In its initial form, PKI uses the concepts of public key cryptography, digital signatures, timestamps, hash functions, Elliptic Curves Cryptography (ECC), certificates, and Certificate Revocation Lists (CRL)s to fulfill the mentioned security requirements of VANETs. We refer the reader to Appendix A where most of these concepts [20,21] are outlined. At the outset, in ECC-based PKI, vehicles register with a Certification Authority (CA) that setups the $L$ security level and chooses the to-be-used elliptic curve parameters: a $2L$ bit prime number $p$ (for non-singular curves), Galois field $F_p$, and elliptic curve equation $E : y^2 \equiv (x^3 + ax + b) \mod p$, where $a, b, x, y \in F_p$. On such $E$, a group $G$ of $(x,y)$ points is chosen with generating point $P$. Since the Elliptic Curves Discrete Logarithm Problem (ECDLP), see Definition A.2 in Section A.2.1, is as hard as the largest (prime order)- subgroup of $E$, we can choose a curve of co-factor = 1 to let group $G$ equals $E$ itself, i.e. of order $(p + 1 \pm 2\sqrt{p})$. Thus, $P$ will be a generator of $G = E$ and will have an order $q$ of a bit length $|q| = |p|^2$. CA will have a (point) identity $ID_{CA} \in G$, a secret key $sk_{CA} \in Z_q^*$, and a related public key $pk_{CA} \in G$. By giving its electronic license plate number $v \in \{0,1\}^*$ for example, $v$ obtains the CA’s public key $pk_{CA}$, a long-term identity $ID_v \in G$ that has no public relationship with the auto’s electronic license plate number $v$, and a certified long-term $(sk_v, pk_v)$ key pair. In this way, CA keeps record of every registered vehicle and its assigned credentials, so whenever a vehicle misbehaves or is being compromised, CA adds it to the CRL list that is periodically broadcasted by the infrastructure to the

---

2In supersingular curves, the bit length of $q$ and $p$ are calculated in a different way.
members of the network through the infrastructure RSUs. When deployed, autos will periodically broadcast safety messages $m_v$ to inform neighbors about their current status. The beaconing rate mainly pertains to the GPS frequency and synchronization interval of DSRC as we stated in Section 2. Whenever OBU$_v$ wants to broadcast $m_v$, $v$ digitally signs the concatenation of a hashed $m_v$ with a fresh timestamp $T_v$ (to avoid replay attacks) using $sk_v$ before sending $m_v$ with the related $pk_v$’s certificate; the format of the broadcasted $m_v$ is:

$$m_v, \text{sig}_{sk_v}(h_1(m_v)|T_v), \text{cer}_{CA}(pk_v),$$

(2.1)

where:

$$\text{cer}_{CA}(pk_v) = pk_v|\text{sig}_{sk_{CA}}(pk_v|ID_{v\text{or}CA}).$$

(2.2)

Receiving (2.1), OBU$_w$ or any RSU will only be able to verify (2.1) if it belongs to the network, i.e., it has the $pk_{CA}$. $w$ first finds whether the received $pk_v$ is revoked by checking it against the CRL. If no, $w$ validates the signature and accepts $m_v$ only if its hash equals the received $h_1(m_v)$. The main building blocks of PKI is the Elliptic Curve Digital Signature Algorithm (ECDSA) of 256 bit keys [20,21]. For the integrity of the data, DSRC recommends the Secure Hash Algorithm (SHA)-2 family of hash functions, specifically SHA-256 that produces 256 bit message digests [7,14]. With PKI, the security requirements of authentication, non-repudiation, message freshness, traceability, credential revocation, and message integrity are fulfilled. However, PKI lacks preserving vehicles privacy and attackers can link the vehicle used $pk_v$ to its identity; therefore, in what follows we review the available solutions that address this problem.
2.1.2 PKI’s Lack of Privacy and Available Solutions

To overcome the PKI privacy-deprivation problem, an intuitive solution was presented in [22] where CA assigns each registered vehicle a certificated set of short-term public keys (pseudonyms) rather than only one long term key. These \( \{pk_i^v\} \) with their related \( \{sk_i^v\} \) will be updated yearly upon vehicle checkups\(^3\). Thus, whenever OBU\(_v\) wants to broadcast \( m_i^v \), it signs using one of the secret \( sk_i^v \) keys before sending with the associated \( pk_i^v \)'s certificate:

\[
m_i^v, \text{sig}_{sk_i^v}(h_1(m_i^v)|T_i^v), \text{cer}_{CA}(pk_i^v).
\] (2.3)

To improve upon such adjustment, some research concerns itself with when to change pseudonyms [23] and what is the best rate of changing [24], ignoring the fact that passive attackers could still track a single vehicle if it is the only one changing pseudonyms within an observed area (successful correlation of old and new pseudonyms). Thus, changing pseudonyms is ineffective if it takes place arbitrarily [25]; vehicles can still be tracked [26]. Therefore, research has been shifted to a new intuition in which changing pseudonyms is conducted while the vehicle is within a crowd to confuse the passive attackers. Researchers start building their protocols making use of the infrastructure RSUs [27–29]; [27] suggests the fixed mix zones idea where vehicles only change their pseudonyms when they arrive at road intersections. In such mix zone, all communications are symmetrically encrypted using RSU’s secret key to hide the process; once a vehicle detects that it is in the mix zone it is forced to change its pseudonym even if it does not need to. The cascading of mix zones throughout VANETs leads to broader more robust mix networks. Other works [28,29] have been

\(^3\)In [16], \( v \) should change pseudonym approximately every 60 sec; therefore, any \( pk_i^v \) is used for around 600 broadcasts if the considered beaconing rate \( T \) equals 100 msec.
elaborating on the best placement of a mix zone as well as on how to improve the preliminary work of [27]. Another route of research [30–35] focuses on creating dynamic mix zones/crowds/groups, within which vehicles can safely change pseudonyms without the help of any RSU:

a) One direction of research investigates the idea of inserting silent periods in-between successive broadcasts during which vehicles can change their pseudonyms; once the silent period is concluded, the vehicle resumes its usual broadcasting using the new pseudonym [30]. CARAVAN [33] efficiently expands this idea such that autos enter random silent periods in probe applications only when they can form a group of same-velocity travelling neighbors to baffle the attacker. Furthermore, AMOeba [34] focuses on increasing the randomness of occurrence of silent periods while Silence at LOw Speeds (SLOW) [35] suggests relating the occurrence of silent periods with speed such that the grouped-vehicles enter silent periods only when the speed is less than 30 km/h.

b) The idea of allowing the vehicles to change their pseudonyms only if their surrounding context (in terms of number of neighbors, their velocity, and their direction) is adequate is adopted in [31]. Each vehicle has a stability timer which when it reaches its limit, the vehicle automatically assesses its context; only if the context is adequate the OBU changes the pseudonym. Such scheme assumes that all of the neighbors who adhere to the required context are cooperative.

c) Random Encryption Period (REP) [32] handles the dynamic mix zones’ issue differently. When any vehicle needs to change pseudonym, it asks the neighbors to surround it by an encrypted zone making use of a shared secret group key given
to the OBU's by a trusted authority. During the encrypted period, the initiating vehicle can safely change its pseudonym while keep checking the neighborhood for additional pseudonyms' changes; if no other change occurs, the originating vehicle extends the hiding period. In their scheme, before deployment each vehicle obtains the same initial group key as well as a set of symmetric keys chosen randomly from a large pool of keys. Therefore, in case of any revocation, the revoked OBU(s) credentials need to be removed from all other vehicles and the group key need to be updated accordingly, i.e., the new group key must be derived from an updated symmetric-key set in each OBU.

In these schemes, digital signatures constitute the main ingredient in achieving the authenticity of identities (through certification) and data. However, because such methods suffer from mixing/encryption overhead plus certificates' large storage, refilling and linkability problems, another research track has been followed where certificates are not needed. Making use of the advanced cryptography of Group Signatures GS [36], at the time of registration, all OBU's are included in the same group; the Group Manager (GM), which has a group public point key $g_{pkGM} \in G$ and a master secret scalar key $g_{skGM} \in Z_q^*$, allots each vehicle with his group public key $g_{pkGM}$ as well as a vehicle-designated secret scalar key $g_{skv}$. After activation, whenever $v$ wants to broadcast beacons, OBU$_v$ signs using the given secret key $g_{skv}$ as

$$m^i_v, \text{sig}_{g_{skv}}(h_1(m^i_v)|T^i_v).$$

(2.4)

Any receiving entity that has the group public key can verify (2.4) using the given group public key $g_{pkGM}$. By observing how GS signs outgoing messages, it is easy to pinpoint the offered super-anonymity feature; the receivers can verify that the
signer is authentic without knowing who is the signer. Only TA/GM can reveal the identity of the signer since it has the master secret key $g_{sk_{GM}}$. In other words, identity exposure is avoided in GS framework. The main disadvantage in group signatures is its lengthy signatures that lead to slow verification at receivers and to a bulky communicated security overhead. For this reason, researchers have been suggesting shorter versions of GS [1, 37]. Other research focuses on batch verification of group signatures [38, 39] to expedite the testing at receivers. Another followed route is by merging group signatures with PKI to improve PKI and to optimize GS [39]. In such approach, the certificates are removed and are substituted with group signatures over the $pk^i_v$ public keys of vehicles:

\[
m^i_v, \text{sig}_{sk^i_v}(h_1(m^i_v)|T^i_v), \text{sig}_{g_{sk_v}}(pk^i_v). \tag{2.5}
\]

This scheme (2.5) does not reduce the number of verification, i.e., the receiver still needs two verifications to authenticate any message. However, [39] presents triple optimization to reduce the cost of computation at senders, communication over air, and verification at receivers. As far as we know, up to now, [39] has been the only effort of hybridizing PKI with other security framework. Our second research track in Chapter 4 uses the new authentications to build the cover that hides the auto’s pseudonym changing and preserves the privacy.

### 2.2 Confidentiality Requirement of Non-Safety Application Layer

The anticipated V2I communications define non-safety non-time-critical type of applications such as providing drivers with weather/travel/road infotainment services,
offering Internet access, facilitating video/audio streaming, and even allowing sensitive data transactions like electronic payments and pricing [8, 15, 16]. In most cases, the sensitive non-safety information would either be unicastly exchanged on demand with the infrastructure such as the electronic payments in toll collections or with other vehicle(s) in same convoys or platoons where the vehicles are interrelated but they do not want the content of their communication to be visible to the public. In either one of these V2I or V2V setups, we need a secrecy/hiding feature besides the mentioned security requirements [15–18, 40]; therefore, confidentiality research has been concerned with:

a) *Finding A Suitable Low-level Encryption Primitive:* since encryption has been recommended to prevent uninvolved parties from gaining any private data [15, 16, 41], the IEEE 1609.2 has adopted the Advanced Encryption Standard (AES) symmetric block encryption primitive (in a CCM mode of authentication\(^4\)). Although AES is fast, its key length is critical for security such that 256 bit keys are required; also, to afford identity privacy, AES needs mechanisms for keys’ management and distribution and a hardware security for the TPD that will hold the private keys. Later on, with the advancement of ECC field, the DSRC IEEE 1609.2 standard suggests using the hybrid Elliptic Curve Integrated Encryption Scheme (ECIES). Although ECIES introduces fast computations and short keys, it mainly needs a preliminary Diffie-Hellman (DH) key agreement protocol (Fig. 2.2). After authenticating the OBU\(\text{s}\) with the aid of certificates and digital signatures of PKI, when vehicles need confidentiality they would use their own certified asymmetric key pair (e.g. \(sk_v\) and \(pk_v\) for vehicle \(v\) and \(sk_w\) and \(pk_w\) for vehicle \(w\)) in deriving two

\(^4\text{CCM=Cipher Block Chaining Message Authentication Code (CBC-MAC)}\)
new symmetric keys using the Key Derivation Function (KDF). KDF creates $k_e$ for symmetric encryption/decryption functions $enc/dec$ using AES-256. Also, KDF creates $k_m$ to use with the message authentication code (MAC) algorithm to create an authentication tag for verifying integrity of $m_v$ at receivers [20,21]. Once again, since ECIES lacks the identity privacy-preservation property of AES, ECIES needs a preliminary step of authentication before it can securely encrypt any transferred data [7].

![Figure 2.2. ECIES for Confidentiality in Vehicular PKI](image)

b) *Arranging A High-level Sensitive Payment Services Protocols:* aside from deciding which encryption primitives to embrace, the WAVE suite has been fashioning a theme for managing any anticipated vehicular electronic payment service under the 1609.11 standard [7,14,41–43]. In addition to such 1609.11’s proposal, there have been other individual efforts that also focus on suggesting arrangements for payment services in VANETs [44,45]. On one hand, Isaac et al. [44] use the idea of
symmetric encryption to secure a transaction that a registered client vehicle might make when buying some merchandise from a merchant who is not registered with the infrastructure; in such situation, the client will be the proxy through which the transaction takes place. Their protocol has two phases: (i) a registration phase in which the client gives the unregistered merchant, through an encrypted message, a secret master key with which and with its derivatives the next phase is conducted and (ii) a payment phase in which the client and merchant will converse through several symmetrically encrypted request/response messages to agree on the funds to be transferred before the client/proxy makes the actual transaction with his bank through an infrastructure’s payment gateway. Once the funds are transferred from his bank to the merchant’s bank, the client confirms that to the merchant. Their protocol suffers from an extensive amount of messages, many sets of symmetric encryption keys discarded and regenerated after every single transaction the client might make with any merchant. To eliminate such extensive amount of exchanged messages and encryption overhead, Chaurasia and Verma [45] present a payment protocol for toll tax collection that does not need the traditional stop-and-pay scenario. They devise a secure fast lightweight procedure using the concept of blinded e-coin to send the money to the toll collector in a private, authentic, and swift way. Blinded coin is a lightweight financial cryptography mechanism that uses anonymous blind signatures to generate an anonymous random-looking electronic object/text (coin) with least delay. Their procedure has three phases: offline withdrawal, online payment, and deposit. Considering the payment phase, five steps are needed to establish the payment. However, their protocol lacks the trustability of PKI framework where no certificates and signatures are used for authentication;
instead a one to one mutual authentication is established at the time of collection between the vehicle and infrastructure. In this way, any vehicle that does not make the mutual authentication can slip out without paying and cannot be traced.

Our contribution in Chapter 3 merges these two aspects by utilizing a low-level homomorphic encryption concept in achieving the high-level confidentiality of any sensitive non-safety application.

### 2.3 Vehicular Communication Efficiency Requirements

The DSRC IEEE 802.11p utilizes the rules of CSMA/CA paradigm to access the single limited-interval CCH channel. Before a vehicle $v$ can send a beacon, it has to first *sense* the medium for a Distributed Coordination Function (DCF) Interframe Space (DIFS) time interval. If the channel is sensed idle, $v$ backs off or counts down for a number of $\beta$ time slots before it begins the transmission. If $v$ finds the medium busy, $v$ freezes the count down and resumes when the channel becomes idle again. Once vehicle starts sending, the CCH channel would be reserved for quite a bit allowing no other vehicle to send; therefore, communication beacon collisions arise for safety applications when vehicles compete for the single available CCH channel. In other words, CSMA/CA is a slotted-contention-based access mode that generates collisions between senders if they choose the same time slot to access the medium. When CSMA/CA is used in unicast communications, collision detection is achieved using the acknowledgements and handshaking mechanisms. Once a collision is detected, doubling the CSMA/CA contention window $W_0$ (binary exponential backoff counter) gives the vehicles a better opportunity to choose a different time slot when sending.
and in return lower the collision probability. However, in broadcast settings, acknowledgments and handshaking cannot be utilized since an acknowledgment generated by all receivers will flood the feedback channels. Since there is no collision detection, the collision rate increases with the number of contending vehicles. Furthermore, the access delay in CSMA/CA is unacceptably high when the number of contending users sharing the same medium goes above a certain limit [7, 14, 46, 47]. Besides, the availability of only one limited-width CCH channel for beaconing in DSRC worsens the contention issue of CSMA/CA. Another factor, by which the aforementioned CSMA/CA problem is affected, is the high security overhead of PKI to be attached with each broadcasted beacon [7, 14, 46] as shown in Table 2.1. When many vehicles compete for the single CCH using CSMA/CA, such large security overhead causes a bandwidth penalty and wastes the channel resources.

| Table 2.1 |
| Safety Beacon Size and Cryptographic Overhead (bytes) |

<table>
<thead>
<tr>
<th>All Headers</th>
<th>$m_v$</th>
<th>Security Level of 128 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\text{sig}_{sk_v}(h_1(m_v)T_v)$</td>
</tr>
<tr>
<td>50</td>
<td>50-150</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\text{sig}<em>{sk</em>{CA}}(pk_v</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These three problems show that congestion is a burden that affects the efficiency of beaconing communication. Therefore, researchers have been investigating congestion through building stochastic analysis models [46, 48–50]. Ma et al. [46] propose an M/G/1 queue to model the occurrences of safety beacons in VANETs. Their assumption, of having a Poisson beacon generation rate, is unrealistic since messages

$^5$Most literature refers to a total size of 500 byte rather than the 360 byte calculated here. This comes from [16] where the authors suggest two underlying cryptographic primitives for PKI: ECC and the Number Theory Research Unit (NTRU)'s lattices. Based on ECC, the signature leads to the 360 byte total size while in NTRU a 679 byte size is resulted. Therefore, an average of $\approx 500$ byte is suggested to be the dominant beacon size.
are periodically broadcasted. Therefore, Vinel et al. [48,49] prefer to model by using a D/M/1 queue of deterministic beacon generation rate. They analyze the performance of the periodic broadcast in terms of beacon collision probability and average beacon delay in both saturated and unsaturated traffic conditions. Both models [46, 48, 49] ignore the freezing mechanism of CSMA/CA backoff-counter when sensing a non-idle channel. Therefore, authors in [50] address freezing time when modeling using a D/M/1 queue. Their numerical results show that a relatively large backoff contention window $W_0$ and a short beacon size $R$ help improve the reliability of broadcast. Still, building a more precise analytic model is a necessity. Other research simply uses simulations to investigate congestion. To effectively evaluate the beaconing performance, Naghavi [51] simulates different channel conditions, different control parameters, and different congestion control mechanisms. Another group of researchers choose to simply conduct empirical experiments [52]. They choose to collect empirical measurements from four test sites across Europe to explore the achieved performance of beaconing. Their results show that manipulating the power of transmission and/or beaconing rate can affect the neighborhood awareness of vehicles.

### 2.3.1 Countermeasures To Overcome Inefficient Beaconing

In an IEEE Spectrum article [53], authors highlight the inevitable performance degradation that beaconing undergoes when the network grows in size, i.e., when congestion increases. Thus, decentralized congestion control research has been vital in DSRC communications [7, 14, 46, 47]. Ranging from controlling the power of transmission, rate of beaconing, MAC/physical layer parameters, and/or other application layer criteria, the available mechanisms can be mainly categorized as follows:
a) Beaconing rate adaptation schemes: this research path attempts to answer an essential question of how frequently a vehicle should beacon. Beacons that are broadcasted too often intensify the CCH load with slight profit. On the contrary, if beacons are transmitted scarcely, they may fail to provide the anticipated real-time cooperative awareness. In any of the suggested adaptive beaconing rate approaches [54–56], each vehicle individually reduces its own rate based on channel quality, beacon delivery error rate, local vehicle status, surrounding-traffic’s situational conditions, or simply pure priority settings and time slot reservations per vehicle. Combining these settings into a hybrid adaptive beaconing rate approach to achieve the best communication quality is addressed in [57]. For a full survey of the existing approaches, readers could obtain more information from [58]. However, none of these approaches utilizes the concept of grouping or convoys. To the best of our knowledge, only one study [59] mentions the notion of adaptive group rate. In addition, in a most recent work by Llatser et al. [60], the adaptive beaconing in multi-lane convoys is studied. The authors attempt to find the sweetest spot of beaconing rate that any vehicle could use to offer best information to the neighborhood without being excessively nor rarely beaconing.

b) Authentication Overhead Omission. In addition to the adaptive beaconing rate schemes, other literature [61–64] choose a different way towards achieving the communication’s efficiency. Most specifically they focus on vehicular PKI and the upper layers’ parameters. They basically honor the ideology of omitting/reducing the PKI security overhead that accompanies each beacon, see (2.4) and Table 2.1. Kargl et al. [61] with Schoch in [62] suggest cutting the PKI’s signature and certificate portions without compromising the required level of security. Removing the signature,
the certificate, signature and certificate, signature verification, certificate verification, or both verification, are all variations of omissions they propose. They suggest the elimination to occur conditionally only in certain vehicle’s contexts when the PKI-primitives are not heavily needed. Yet, although “short-term” eliminations reduce the communicated overhead, it might open a door for attackers to breach into the system. Therefore, Feiri et al. [63] revisit their idea of omission but this time by adaptively omitting the authentication certificate from beacons. “adaptively” means making the omission rate dependable on the estimated channel congestion. By analyzing [63], Feiri et al. [64] conclude that out of the five omission mechanisms, the congestion-based certificate omission is the most acceptable scheme.

Our contribution in Chapter 5 shows how we use overhead reduction techniques with grouping concept to boost beaconing efficiency without relaxing the needed security level. To estimate the performance of our introduced G-BEE beaconing efficiency scheme, we build our own D/M/1 and D/M/1/1 analytic models.

c) A third line of research lies within using alternative authentication schemes that have less communication overhead; in the following subsection, we review the available efforts in such area of providing beaconing efficiency.

2.3.2 Identity Based Authentication (IBC) Schemes: Review

In 1984, Shamir [65] lays the ground for identity-based cryptography schemes. The improvement in IBC authentication over PKI depends on discarding the certificates and signatures for the sake of a better cost-effective authentication; instead, alias/pseudo/anonymous identities are used as the public verification keys; the signing keys are derived from these identities. Over the last years, various identity-based
authentication approaches have been proposed including Chameleon signatures, blind signatures, ring signatures, group signatures, Boneh-Lynn-Shacham (BLS) signature, and Identity-Based Signature (IBS). Within the vehicular community, several studies \cite{2-5,66-76} (Fig. 2.3) have been investigating the feasibility of embracing these authentication schemes to serve the needs of vehicular security, privacy preservation, and efficiency.

![IBC Authentication Schemes Classification](image)

**Figure 2.3. IBC Authentication Schemes Classification**

Depending on the type of realization of the underlying elliptic curves operations, these vehicular-adapted IBC authentication strategies can be mainly divided into two categories. Until recently, most of the IBC adaptations have heavily relied on the pairing-based cryptography (see A.2.2 in Appendix A) on elliptic curves. Gamage et al. \cite{66} and Zhang et al. \cite{68} build a Conditional Privacy Preserving Authentication (CPPA) authentication schemes in VANETs using ring and blind signatures. However, ring and blind signatures afford no traceability, i.e., faulty vehicles cannot
be spotted. To overcome such unconditional privacy, Lin et al. [67] introduce GS and Identity-based Signature (GSIS); a scheme that uses GS for V2I communication and IBS for V2V interaction. Trying to examine the feasibility of the other IBC pairing-based signatures, Shen et al. [69] use Chameleon hash function to achieve a CPPA scheme. However, their scheme does not provide unlinkability; although every sent message is signed, it can be easily linked to other messages. To have all-in-one solution, in 2008, C. Zhang et al. [70, 71] examine the use of the IBS signature for authentication and batch verification for V2I communication in VANETs to solve the weaknesses in the aforementioned IBC authentication schemes. However, Chim et al. [72] and Lee et al. [73] analyze Zhang’s proposals [70, 71] and find that they are vulnerable to impersonation and replay attacks since the same key is used in all TPDs; also, C. Zhang uses a specific inefficient pairing function. To maneuver around such weakness, [72] uses Boneh’s BLS signature to modify Zhang et al.’s proposal; Chim et al. eliminate the need for TPDs. Their proposal suits V2V and V2I batch verification. After analyzing Chim et al.’s scheme, Tzeng et al. [74] find that [72] is still susceptible to the impersonation attack that it is initially built to overcome. Aside from these nested investigations, Shim [4] carves a different path to improve Zhang et al.’s proposals for V2I scenarios. First, [4] does not use the same complex inefficient mapping/hashing (MapToPoint) function; she gives better TPDs’ assumptions where there is no pre-loading of the long-term TA’s master secret. Despite her CPPA scheme efficiency, Liu et al. [75] spot that Shim has a flaw in her proof and her scheme does not counteract modification attacks. Both Bayat et al. [5] and Jianhong et al. [76] build on Liu et al.’s investigations. Unfortunately, they point out that [75] scheme does not afford non-repudiation feature, i.e., it still has Shim’s modification
attack; although both [5, 76] propose two improved CPPA protocols, their proposals do not withstand modification attacks.

Since the performance and complexity of bilinear pairings do not fit the real-time periodic vehicular communications, the gears are shifted in the recent years to investigate a second category of IBC strategies that do not use bilinear pairings. The new pairingless-based realization category skirts around the cost and complexity of pairings by using modified versions of Schnorr signature on elliptic curves [77]. Since Schnorr signature uses secure hashing functions that map to scalar elements, multiplication of points by scalars on elliptic curves is easier and faster than multiplying/pairing of elliptic curves points. Two research studies: He et al. [3] and Lo and Tsai [2] propose two similar IBC authentications with no bilinear pairings. [3] is for any communication setup while [2] is for the bidirectional communication between vehicles and infrastructure. Their schemes incur much less computation and communication cost in comparison with the other pairing-based approaches. Both [3] and [2] CPPA approaches preserve the anonymity of the vehicles since they use pseudo-identities rather than real identities, unlinkability of the broadcasted messages as the pseudo-identities change overtime, and traceability by the TA or by the Public Key Generator (PKG) of any liable vehicle in case of dispute. An instantly noticed weakness of these two schemes is their dependability on the costly installed TPDs for achieving hardware security and for generating the needed authentication materials; hence, the use of the TPDs will considerably escalate the cost of the vehicular networks. Although TPDs can erase themselves whenever their sensors detect physical manipulation by launching side-channel attacks, such as power analysis and laser scanning, adversaries can effectively leak out the stored sensitive data. In this sense,
schemes that do not utilize TPDs are more preferable; Trusted Platform Module (TPM)s that cost only tens of dollars [78] and software-manipulated techniques [72] are used as alternatives. In this sense, the emphasis on having pairing-less and TPD-free authentication scheme has motivated us to propose our PME-CPPAS approach in Chapter 6. Furthermore, our preliminary PME-CPPAS has encouraged us to introduce a cleaner and a finely tuned blending version of these three IBC frameworks to use in our final CAAIS contribution in Chapter 7.
Chapter 3: Augmenting The Confidentiality of Sensitive Non-safety Applications Through SHE Encryption and Secure Pairwise Key Establishment

As mentioned in 1.2 and 2.2, the embraced AES and ECIES encryption schemes, by DSRC’s 1609.2 security services standard, come with slight nuisances. The AES’s keys length are critical for security; if one key is captured the whole system is compromised. The ECIES is fully contingent on the beforehand availability of a secure authentication framework. For these two hinted at points, this track of research attempts to test the feasibility of using an alternative encoding scheme. Making use of a new state-of-the-art partial homomorphic encryption concept and an elegant key pre-distribution scheme, we imagine a scenario where vehicles can dynamically generate encrypted shared session keys to use for confidentializing a high-level vehicular sensitive-data transfer. The obvious advantage of our model over the other encrypting schemes is the hiding of all stored and transferred data even when there is no TPDs installed. Also, our proposal emphasizes a dynamic establishment of one to one mutual authentication without an already established authentication framework; it can still work with an already established authentication. Before we delve into the description of our constructions, we refer the reader to A.4.1 in Appendix A to grasp
the building blocks of Boneh-Goh-Nissim (BGN) system and to A.3.1 to understand Blom’s key deployment scheme.

3.1 Model Assumptions and Preliminary BGN-Blom Construction

Our system has \( n \) vehicles registered with a central authority CA that has three sets of parameters at \( L \) bit security level:

a) PKI authentication parameters of elliptic curve \( E: (a, b, p, q, G, P) \). The group \( G \) is of order \( q \), where \(|q| = 2L\) bit. From these parameters, \((ID_{CA} \in G, sk_{CA} \in Z_q^*, pk_{CA} \in G)\) are derived as in Section 2.1.1.

b) BGN system parameters of a super singular elliptic curve \( E_{BGN}: (a = 1, b = 0, \bar{n} = q_1q_2, p_{BGN}, G_1, G_2, P_{BGN}, \hat{P}_{BGN}) \) with group \( G_1 \) of order \( \bar{n} \) and \( G_2 \) group at extension field over embedding \( \varphi = 2 \) where pairing \( e \) results lie. \(|\bar{n}| \geq (|q_1|,|q_2|) \geq 2\frac{L}{2}\) bit and \((p_{BGN} = (\ell \in Z^+), \bar{n} - 1 \equiv 2 \mod 3)\). From such parameters, public key of BGN \( pk_{BGN} = (\bar{n}, P_{BGN}, \hat{P}_{BGN}, G_1, e, G_2) \) is set and secret key \( sk_{BGN} = q_1 \). Both \( q_1 \) and \( q_2 \) are two hidden random primes of \( \frac{L}{2} \) bit length as in Section A.4.1.

c) For the Blom parameters, CA contains the underlined prime \( p_{Blom} \geq 3 \) of the used Galois Field \( F_{p_{Blom}} \). CA has the generator \( \psi \in F_{p_{Blom}} \) and the maximum tolerated/allowed capturing level \( \lambda \), from which the \( \Psi \) matrix (A.1) is generated. Also, the \( \Upsilon \) matrix (A.1) of \( \bar{\bar{r}}_{i,j} \in F_{p_{Blom}} \) is set at CA.

When vehicle \( v \in \{1, ..., n\}^6 \) registers with our CA, \( v \) obtains three sets of parameters. First, \( v \) is given the \((ID_v \in G, \{sk^j_v \in Z_q^*, pk^j_v \in G, cer_{CA}(pk^j_v)\})\) elements

\(^6\)We assume \( v \) to be a numerical identity between 1 and \( n \).
of PKI. Second, $v$ obtains the $sk_{BGN}, pk_{BGN}$ keys of BGN. Third, $v$ is given the $(\lambda, \psi, n, p_{Blom})$ of Blom scheme besides the related column $v$ of $(\Upsilon, \Psi)^{Tran}$ matrix as well as a designated row $v$ of the $\Psi$ matrix.

The motivation behind our preliminary construction is that BGN can evaluate quadratic multivariate polynomials on ciphers provided that the resulting value falls within a small set. In this sense, the preliminary BGN-Blom construction boils down to the idea that row $v$ of matrix $\Psi$, column $v$ of $(\Upsilon, \Psi)^{Tran}$ become inputs $A_{v,k}$ and $B_{k,v}$ as well as identity $v$ to an offline BGN encrypting system that encrypts them using $pk_{BGN}$ key and places their ciphertexts

\[ C_{A_{v,k}} = enc_{pk_{BGN}}(A_{v,k}) = P_{BGN}^{A_{v,k}} \cdot \hat{r}^{\in \mathbb{Z}_n^{\ast}} \in G_1, \tag{3.1} \]

\[ C_{B_{k,v}} = enc_{pk_{BGN}}(B_{k,v}) = P_{BGN}^{B_{k,v}} \cdot \hat{r}^{\in \mathbb{Z}_n^{\ast}} \in G_1, \text{ and} \tag{3.2} \]

\[ C_v = enc_{pk_{BGN}}(v) = P_{BGN}^v \cdot \hat{r}^{\in \mathbb{Z}_n^{\ast}} \in G_1 \tag{3.3} \]

at OBU$_v$, where $k \in \{1, \ldots, \lambda + 1\}$. After deployment, whenever $v$ wants to establish a sensitive data transmission with a specific vehicle/infrastructure $w$, $v$ can either use the already established PKI authentication to assist the sought handshake with $w$ whose certificate was received at some point. $v$ broadcasts a request message $m_v^i$ accompanied by the associated PKI authenticating signature and certificate:

\[ m_v^i = <create - session, h_1(C_v), h_1(\text{cer}_{CA}(pk_v^{i})) >, \text{sig}_{sk_v}(h_1(m_v^i)[T_v]), \text{cer}_{CA}(pk_v^{i}). \tag{3.4} \]
Upon receiving and verifying the certificate and signature of (3.4), \( w \) knows that it is the intended receiver and only \( w \) can reply with \( m^i_w \) to acknowledge \( m^i_v \):

\[
m^i_w = < creat - ack, h_1(C_w), h_1(\text{cer}_{CA}(pk^i_v)), sig_{sk^i_w}(h_1(m^i_w)|T^i_w), \text{cer}_{CA}(pk^i_w) >.
\]

(3.5)

Or \( v \) can use a dynamic one to one mutual authentication to establish the desired handshake without the PKI assistance; \( v \) can broadcast

\[
m^i_v = < C_w, h_1(C_v) >, h_1(m^i_v|T^i_v)
\]

(3.6)

through which only the intended receiver \( w \) can understand and accept the message.

This type of authentication is contingent upon having the \( C_w \) of the intended receiver stored at the sender, i.e, sender \( v \) knows all of the legitimate identities of network members; this can only be done if lookup tables are used as we will see below. once accepts (3.6), \( w \) can acknowledge the reception by broadcasting

\[
m^i_w = < C_v, h_1(C_w) >, h_1(m^i_w|T^i_w).
\]

(3.7)

By the end of this handshaking phase, each of the involved OBU’s needs to have the plain real identity of the partnered vehicle to obtain the related column vector of \( \Psi \). Here, the OBU has two options to choose from:

Option 1. Assuming each OBU has been fed offline by \( n \) encrypted column vectors each of \( \lambda + 1 \) length, so each vehicle needs to store a large lookup table of \( n \) entries in the form of \( (C_{1 \leq k \leq n}, [k^{th \text{ Encrypted Vector Elements}}]_{\lambda + 1}) \). This option is fast since no BGN decryption is needed. OBU can easily recognize false received identities.
Option 2. To eliminate large lookup tables, a slower option can be chosen by OBUs. Vehicles $v$ and $w$ have to decipher each other encrypted identity using BGN system:\footnote{Decrypting identities using BGN is efficient as long as $v$ identity originates from small space since decryption complexity is $\mathcal{O}(\sqrt{|v|})$.}

\[
v = \text{dec}_{sk_{BGN}}(C_v) = \text{Dlog}_{p_{BGN}^{q_1}}^{C_v^{q_1}} \quad \text{and} \]

\[
w = \text{dec}_{sk_{BGN}}(C_w) = \text{Dlog}_{p_{BGN}^{q_1}}^{C_w^{q_1}}.
\]

Having the numerical identity of OBU $w$, $v$ acquires the related $\Psi_w$ column by dynamically generating $\Psi_w$ from $\psi^w$ using the square-and-multiply algorithm \cite{20} then $v$ ciphers with BGN.

By following either option, the modified Weil pairing (Definition A.4 in Appendix A) is used to homomorphically calculate the product of the owned encrypted row vector $C_{A,v,k}$ and the other party’s column vector $C_{B,k,w}$ to obtain the following $C_{AB}^1 \leq k \leq \lambda + 1$ ciphertext:

\[
C_{AB_k} = e(C_{A,v,k}, C_{B,k,w}) = e(\text{enc}_{pk_{BGN}}(A_{v,k}), \text{enc}_{pk_{BGN}}(B_{k,w})).
\]

\[
e(P_{BGN}, P_{BGN}^{\rho_{q_2}})^{\bar{r} \in \mathbb{Z}_n^{\lambda}} = e(P_{BGN}, P_{BGN})^{A_{v,k}B_{k,w}} \in G_2.
\]

By homomorphically multiplying $C_{AB_k}$, we acquire their addition to be hashed using SHA-256 to give the encoded 256 bit key

\[
C_{K_{v,w}} = h_1(\prod_{k=1}^{\lambda+1}(C_{AB_k}). e(P_{BGN}, P_{BGN}^{\rho_{q_2}})^{\bar{r} \in \mathbb{Z}_n^{\lambda}}) \in G_2
\]

that will be used with AES encryption scheme to symmetrically confidentialize any further sensitive data transfer from $v$ to $w$ or from $w$ to $v$:

\[
< C_{m_v} = \text{enc}_{C_{K_{v,w}}}(m_v^i) >, \ h_1(C_v|C_w|T_v^i) \text{ and } \]

\[
(3.12)
\]
In both (3.12) and (3.13), we attach a hashed value of the involved parties’ identities with a current timestamp to avoid any chance of impersonation and replay attacks. Fig. 3.1 exemplifies how BGN-Blom constructs the secretive key $C_{K_{v,w}}$ between $v$ and $w$. In addition to such pairwise secret data sharing, we show that our model also works in a broadcast setting; the created key is used for encrypting broadcasted data as in (3.14) and in Fig. 3.2:

$$< C_{m^i_v} = enc_{C_{K_{v,w}}} (m^i_w), h_1(C_w|C_{v}|T^i_{w}) >.$$  \[ (3.13) \]

To the best of our knowledge, our study represents the first investigation on the feasibility of utilizing the partial homomorphic BGN cryptosystem [10] in encapsulating a dot product of Blom’s vectors [79] for a dynamic generation of an identical obfuscated key pair for encoding an established delicate V2V or V2I transaction. Having defined the construction, we articulate two probable application scenarios:

a) automated-driven automobiles target the same destination, travel at the same speed, maintain small distances in-between, and remain in the power range of each other for a long time. Such vehicles would use our BGN-Blom model in establishing a private V2V intra-convoy communication to converse, letting no outsider disclose the content of the intra-transferred data.

b) an OBU is engaged with infrastructure $I$ through a V2I link to make an electronic payment. In such setup, to expedite processing and reduce congestion, the vehicles do not need to stop and make a manual payment; rather, they use the model
Figure 3.1. Preliminary BGN-Blom Session Key Generation

Figure 3.2. BGN-Blom Key for PKI Authentication
in establishing a confidential communication with the facility. Here, the involved
RSU $I$, which registers with only one regional CA, is responsible for initiating the
communication with any approaching vehicle $v$ by periodically beacons its double
encrypted identity $C_I$:

\[
< m^i_I, C_I >, \text{sig}_{sk_I}(h_1(m^i_I)|T^i_I), \text{cer}_{CA}(pk_I). \quad (3.15)
\]

Any OBU$_v$, which has $pk_{CA}$, would be able to verify the certificate of (3.15), unless
it belongs to another CA; in such case, $v$ must be augmented with any cross-
certification techniques [41]. Another subtle tweak, of which we need to be aware,
is the identifier of the facility $I$; since it belongs to a single CA that has $n$ registrees,
the infrastructure identity $I$ can be $\geq n + 1$ without affecting the $\lambda n$ upper bounds
that are limited to specific values as we will see in 3.1.1.

In both scenarios, the created key pair is used only throughout the intended
period of communication. Once the session's time ends, one of the involved parties,
the vehicle in case of V2V and the infrastructure in case of V2I, must initiate an
end-session message:

\[
m^i_v = < \text{end} - \text{session}, \text{sig}_{sk_v}(h_1(m^i_v)|T^i_v), \text{cer}_{CA}(pk^i_v), \text{and} \quad (3.16)
\]

\[
m_I = < \text{end} - \text{session}, \text{sig}_{sk_I}(h_1(m_I)|T_I), \text{cer}_{CA}(pk_I). \quad (3.17)
\]

In the case of the mutual one to one authentication, either $v$ or $w$ can end the session
by broadcasting

\[
m^i_v = < \text{end} - \text{session}, C_w, h_1(C_v) >, h_1(m^i_v|T^i_v) \quad (3.18)
\]
Moreover, the encrypted keys do not need to be destroyed and refreshed using a one-way hash function of the original key; instead, they can be stored for future sessions as a $< C_{K_v,w}, session_{index}, w >$ tuple at vehicle $v$ for its session with vehicle $w$. When the vehicle is evicted, for being compromised or misbehaved, the tuple would be automatically erased in a way similar to revoking other credentials.

### 3.1.1 Limitation of Original Model

The values of network size $n$, capturing threshold of Blom $\lambda$, and elliptic curve group order $\bar{n}$ of BGN are related in many ways. A first relationship comes from from (3.1), (3.2), (3.3), (3.10), and (3.11); each OBU has $(2\lambda + 3)$ encrypted $G_1$ elements besides $(\lambda + 2)$ elements from pairings and multiplication in $G_2$. Since the network has $(n + 1)$ entities, $n$ vehicles and 1 infrastructure, then

\[(3\lambda + 5)(n + 1) < \bar{n}\]  

inequality has to be satisfied. The value of $\lambda$ is preset by CA beforehand to six possibilities from $5\% n$ to $50\% n$. Also, order $\bar{n}$ has to be large as [10] suggests, e.g. $\bar{n} = 2^{1024}$. Second relationship is through the presence of $\psi^{\lambda n}$ terms, in all Blom matrices’ row and column elements and subsequently $\psi^{2\lambda n}$ terms after pairings. These terms have to be small, i.e., Boneh et al. suggest a $\{0, 1\}^{q_2}$ message space [10] to have robust encryption; therefore,

\[\psi^{2\lambda n} \leq (q_2 = 2^L).\]  

(3.20)

If the $p_{Blom}$ prime in Blom is chosen to be as small as 3, $\psi$ can be chosen to be 2; since Blom vectors are obfuscated with BGN of large $p_{BGN}$ prime, it is safe to choose small $\psi$ value. This setting makes $\lambda n$ value limited to $\frac{L}{4}$ in (3.20). Based on these
two relationships, we can clearly see the limitation of this preliminary model where
we can only deal with small networks of $\lambda n \leq 32$ if $L = 128$ bit security level is
considered. However, even such small $\lambda n$ term will amplify the elements of Blom’s
matrices and will affect the model efficiency; as [80] suggest, only $\{0, 1\}^1$ to $\{0, 1\}^6$
message spaces are practically efficient. In this sense, to overcome such limitation,
the following three modifications are introduced.

3.2 Less Complex and More Efficient Alternatives

To have larger OBUs’ identities while decreasing the expected storage and com-
putation complexity needed per key in the original construction, we must reduce the
size of the BGN inputs. Such goal is achieved either by changing the form of entries or
reducing the dimensions of matrix $\Psi$ [79] that has high impact on efficiency. Because
of the existence of $\psi^{(0 \leq v \leq \lambda) \times (1 \leq w \leq n)}$ terms, we limited $\psi$ to 2 and restricted matrix
$\Upsilon$ to have only random 0 and 1 $\overline{\text{bar}}_{i,j}$ values. The BGN encryption will still be
inefficient since such large terms are the exponents of the BGN’s large base generator
$P_{BGN}$. Therefore, we present the following three modifications to eliminate the need
for such large-size matrix elements:

1. **Modification 1: Reducing Matrices $\Psi$ and $\Upsilon$ Dimensions**. We assume
$\lambda = 1$ to shrink the dimensions of the input matrices such that $\Upsilon$ becomes a single
unit matrix

$$\Upsilon = \begin{bmatrix} \overline{\text{bar}}_{0,0} = 1 \end{bmatrix}_{1 \times 1} \quad (3.21)$$

and $\Psi$ becomes a row vector

$$\Psi = \begin{bmatrix} \psi^1 & \psi^2 & \ldots & \psi^v & \ldots & \psi^n \end{bmatrix}_{1 \times n}. \quad (3.22)$$

41
No matter how small are the matrices’ dimensions, this modification will not accommodate larger identities since \( n \) is still limited to \( \frac{L}{4} \); this ongoing restriction urges us to introduce modification 2.

2. **Modification 2: Remodeling Matrix** \( \Psi \). To overcome modification 1’s restriction, i.e., to accommodate larger \( n \) and \( \lambda \) values, without affecting the key similarity proof (Proof A.1 in Appendix A.3.1) or remodeling \( \Upsilon \), modification 2 slightly reshapes \( \Psi \) by first incrementing its row index before using the \( \lfloor \sqrt{\log \psi} (\Psi) \rfloor \) function to remove the \( \psi^{(1 \leq v \leq \lambda+1) \times (1 \leq w \leq n)} \) terms. Hence, we obtain a reduced

\[
\hat{\Psi} = \begin{bmatrix}
\lfloor \sqrt{1 \times 1} \rfloor & \cdots & \lfloor \sqrt{n \times n} \rfloor \\
\vdots & \ddots & \vdots \\
\lfloor \sqrt{w \times 1} \rfloor & \cdots & \lfloor \sqrt{w \times n} \rfloor \\
\vdots & \ddots & \vdots \\
\lfloor \sqrt{(\lambda+1) \times 1} \rfloor & \cdots & \lfloor \sqrt{(\lambda+1) \times n} \rfloor
\end{bmatrix}_{(\lambda+1) \times n}
\] (3.23)

matrix that automatically accommodates larger \( \lambda n \); rather than the original \( \{0, 1\}^{\lambda n \leq \frac{L}{4}} \) message space, the size of inputs becomes \( (\lfloor \sqrt{(\lambda+1) \times n} \rfloor)^2 \leq 2^{\frac{L}{2}} \). With \( \hat{\Psi} \) and \( \Upsilon \) in hand, the same offline and online steps used in BGN-Blom can be followed to create the final keys. Although this modification is better than modification 1, it still exhibits large \( \Upsilon \) and \( \hat{\Psi} \) dimensions; therefore, we introduce our third adjustment.

3. **Modification 3: Reduce Remodeled \( \hat{\Psi} \) Matrix Dimension**. To further enhance our construction, we merge the benefits of both modification 1 and modification 2, i.e., fewer dimensions as well as smaller \( \hat{\Psi} \) elements to create a final encrypted key with least cost:

\[
\hat{\Psi} = \begin{bmatrix}
\lfloor \sqrt{1} \rfloor & \lfloor \sqrt{2} \rfloor & \cdots & \lfloor \sqrt{v} \rfloor & \cdots & \lfloor \sqrt{n} \rfloor
\end{bmatrix}_{1 \times n}.
\] (3.24)
The usefulness of these three modifications, in comparison with the original model, is examined in Section 3.4. Yet, an intuitive eye-catching change occurs in modification 1 and modification 3. The single $A_{v,1}$ and $B_{1,w}$ entries of the BGN system necessitates no further multiplications in $G_2$; the final key will be represented only by the pairings of $C_{A_{v,1}}$ and $C_{B_{1,w}}$ ciphertext.

3.3 Functional Correctness Proofs

In cryptography, having successful decryption of the included encrypting functions can effectively and simply prove the cryptosystem’s functional correctness. In this sense, the correctness, of the original BGN-Blom construction and its modifications, can be expressed firsthand through the following proofs:

Proof 3.1 (Correctness of Identity Encryption). Given $v$ and $C_v$, the operations performed during identity’s BGN encryption are successfully inverted by their counterparts in decryption:

$$dec_{sk_{BGN}}(enc_{pk_{BGN}}(v)) = Dlog_{P_{BGN}}^{(P_{BGN})^v} = Dlog_{P_{BGN}}^{(P_{BGN}^{\ell})^v} = v$$  \hfill (3.25)

Proof 3.2 (Correctness of Sensitive-Data Transfers). Given message $m^i_v$ at $v$, veiled key $C_{K_{v,w}}$ and $C_{m^i_v}$, the following correctness proof holds where the symmetric encryption is effectively reversed by its symmetric decryption function:

$$dec_{C_{K_{w,v}}}(enc_{C_{K_{v,w}}}(m^i_v)) = m^i_v$$  \hfill (3.26)

Despite the straightforwardness of these two proofs, the correctness of the generated encrypted key $C_{K_{v,w}}$ cannot be seen directly; however, we can trace it back to the
first step when BGN encrypts the $A_{v,k}$ and $B_{k,w}$ inputs before it pairs their ciphertext in $G_1$ to obtain $C_{AB_{1 \leq k \leq \lambda + 1}} \in G_2$ on which a final multiplication is conducted to obtain the key.

**Proof 3.3** (Correctness of Key Encryption). Given $C_{A_{v,k}}$, $C_{B_{k,w}}$, $C_{AB_{1 \leq k \leq \lambda + 1}}$, and $C_{K_{v,w}}/C_{K_{w,v}}$, the correctness of the generated encrypted key can be successfully proved in connection with the first encryption step in our construction:

$$\text{dec}_{sk_{BGN}}(\text{enc}_{pk_{BGN}}(A_{v,k})) = D\log_{P_{BGN}}^{\mathbb{F}_{q_1}}(P_{BGN}^{A_{v,k}} \cdot \prod_{i \in \mathbb{Z}_n^{*}} P_{BGN}^{q_1}) = D\log_{P_{BGN}}^{\mathbb{F}_{q_1}}(P_{BGN}^{A_{v,k}})^{q_1} = A_{v,k} \quad (3.27)$$

### 3.3.1 Adversary Model and Security Fulfillment

To describe the capability of our $\mathcal{A}T$ attacker, we use the quadruple terminology stated by Raya and Hubaux [15] in the (*Membership. Motivation. Method. Scope*) form. Our scheme considers an attacker who is an (*Outsider. Malicious/Rational. Active/Passive. Local*). $\mathcal{A}T$ is assumed to be local in scope with a limited communication range. He can passively eavesdrop on the established communication, also he can actively generate beacons of his own. His future interests are not well identified; he is trying to penetrate into the network either for rational or malicious purposes. Yet, he is not malleable, i.e., he is not able to distinguish the nature of the transferred certificates and signatures. He cannot transform the transferred data into other forms that could be verified to some plaintext similar to the original ones. In face of such attacker, our proofs 3.1, 3.2, and 3.3 clearly show that every intermediate as well as output component is related to the security of BGN system; if BGN is secure, our BGN-Blom constructions are secure. In this sense, our scheme security relies on the hardness of two irreversible computational problems: the Subgroup Decision Assumption (SDA) (Definition A.5 in A.4.1) and ECDLP (Definition A.2 in A.2.1).
Therefore, to fulfill the SDA asset, we keep factors $q_1$ and $q_2$ hidden and large leading to large group order and ciphertexts; for example, when $L = 128$ bit, the group order is $\bar{n} = 2^{1024}$ and ciphertexts’ size is 1 Kbit and 2 Kbit in $G_1$ and $G_2$, respectively. Also, in terms of ECDLP, we choose large $G_1$ group lest pairing lessens the ECDLP hardness in $G_2$.

### 3.4 Comparisons and Analytical Results Discussion

As we stated earlier, the established key in our scenario is a result of evaluating dot products on BGN ciphertexts of Blom’s keying materials. In other words, the scheme is an arithmetization of the Disjunctive Normal Form (DNF), Or of Ands, of width 2 to calculate $\sum_{k=1}^{\lambda+1}(C_{A_{v,k}}C_{B_{k,w}})$ for any $v$ and $w$ vehicles. To test the effectiveness of our four proposed models and the extent of their practicality (for $L$ security level), we designate the following strict conditions:

- the random generator $P_{BGN}$ of $G_1$ must be a large prime since $\bar{n}$ is large and $= 2^{1024}$.
  
  $P_{BGN}$ is the $x$-coordinate of the elliptic curve point.

- we set the generator of the Galois Field of Blom $\psi$ to 2 in $\Psi$ and $\hat{\Psi}$ as well as we limit the $\bar{r}_{i,j}$ in $\Upsilon$ to be of only random 0 and 1 values.

- in the original BGN-Blom, we are required to either assume $1 - 6$ bit [10, 80] size for BGN inputs or suggest bit concatenation when $\lambda n = 32$ because of the large $\Psi$ and $(\Upsilon, \Psi)^{\text{Tran}}$ elements; therefore, to overcome such unrealistic limitation and to obtain enhanced results, our three adjustments must treat BGN inputs as a concatenation of 6 bit blocks of data; such a tweak greatly minimizes the 1 Kbit and 2 Kbit ciphertexts to be encrypted in $G_1$ and $G_2$ groups.
At the outset, our original model shows large overheads because of the included BGN system itself and the large unrefined raw Blom matrices; therefore, our three modifications are presented. A noticeable advantage of the modifications is perceived in the second and third adjustments where it becomes feasible to accommodate larger identities $\lambda n < 2^{\frac{L}{4}}$ using fewer number of bits in $\Psi$’s elements. By comparison, in the original model, even when $\lambda n$ is limited to $\frac{L}{4}$, the BGN inputs are lengthy. To value the effectiveness of the modifications, we compare their performance with that of the original construction with and without lookup matching and for different values of $n$. We make use of the Storage Overhead $S_O$ per key, Computation Complexity $C_C$ per key, and Communication Overhead $C_O$ metrics. For realizing our analysis, we use the MATLAB numerical computations environment [81] over a personal Macbook Pro machine of a 2.9 GHz Intel Core i7 processor and a 8 Gbyte DDR3 memory.

### 3.4.1 Key Storage Overhead and Key Size (bit)

Although VANETs vehicles are equipped with sufficient storage resources, any introduced algorithm or protocol should still incur acceptable storage cost. Our initial model, in all of its six $\lambda$ cases, necessitates slightly high storage cost per key; in Fig. 3.3a, 83.5 Kbit per key is needed when vehicle’s numerical identity $n = 29$, $\lambda = 5\% n$, and no lookup matching is used. With the three introduced modifications, we aim to minimize such cost; for example, only 20.14 Kbit, 6.7 Kbit, and 2.35 Kbit $S_O$ storage per key is required for the same $n$ and $\lambda$ values in modification 1, 2, and 3, respectively. While modification 2 shows poor performance when the numerical identity of the vehicle grows large, modification 3 has a steady overhead regardless of how large the value of the vehicle’s identifier is; for example, in modification 3 even when
$n = 5000$, which is not realistic, only 6.13 Kbit would be needed. To further assess the performance of the modifications, in Fig. 3.3b, we run the analysis with lookup tables, i.e., when the partner’s encrypted identity and its related vector/element are stored in the OBU. The obtained results show that the modifications scale up well with the growing identity; still, adjustment 3 performs best. Since we are considering the $S_O$ storage per key metric, a look at the ultimate obfuscated key size can also show how the modifications solidify the initial scheme. In comparison with the 13 Kbit key size obtained from the original BGN-Blom, when maximum $n$ is considered, enhancements 1, 2, and 3 decrease the key size to 9.6 Kbit, 1 Kbit, and 770 bit, respectively. Even when $n$ grows large in both adjustments 2 and 3, a reasonable-size key is achieved; for example, when $n = 5000$, only 3 Kbit and 2 Kbit are needed to represent the obfuscated key. We can consider hashing the created keys using SHA-2 hash function. Only 256 bit small keys are resulted to match the AES encrypting keys. Table 3.1 compares the calculations of the four models in terms of $S_O$ per key and obfuscated key size.

3.4.2 Computation Complexity and Estimated Time (sec)

Similar to the periodic safety-based vehicular communications, which necessitate real-time performance, the non-safety applications, such as the considered inter-convoy and toll pricing, require real-time response; however, they are more flexible with less stringent delay constraints. In this sense, in terms of Computation Complexity $C_C$ per key, using the underlined complexities of the used cryptographic operations in Table 3.2, it can be seen from Table 3.3 that our original model exhibits higher $C_C$ per key. The reason is the large BGN inputs that need many pairings from $G_1$.
Figure 3.3. Storage Overhead $S_O$ per Generated Key: Comparison

(a) $S_O$ without Lookup Tables

(b) $S_O$ with Lookup Tables
Table 3.1
STORAGE OVERHEAD S_O PER KEY AND KEY SIZE: COMPARISON

<table>
<thead>
<tr>
<th>Metric</th>
<th>Original Construction with</th>
<th>Modification 1 with</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[</td>
<td>A_{v,k}</td>
</tr>
<tr>
<td>Storage Overhead (Kbit)</td>
<td>[S_O = \frac{(4\lambda + 6)</td>
<td>A_{v,k}</td>
</tr>
<tr>
<td>Lookup</td>
<td>[S_O + \frac{(n - 1)((\lambda + 1)</td>
<td>A_{v,k}</td>
</tr>
<tr>
<td>Obfuscated Key Size (Kbit)</td>
<td>[\frac{2</td>
<td>A_{v,k}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>Modification 2 with</th>
<th>Modification 3 with</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[</td>
<td>A_{v,k}</td>
</tr>
<tr>
<td>Storage Overhead (Kbit)</td>
<td>[S_{O2} = S_O]</td>
<td>[S_{O3} = S_{O1}]</td>
</tr>
<tr>
<td>Lookup</td>
<td>[S_{O2} + \frac{(n - 1)((\lambda + 1)</td>
<td>A_{v,k}</td>
</tr>
<tr>
<td>Obfuscated Key Size (Kbit)</td>
<td>[\frac{2</td>
<td>A_{v,k}</td>
</tr>
</tbody>
</table>

to \(G_2\), many point additions in \(G_2\), decrypting the received partner’s identity, and generating its related vector when no lookup tables are utilized.

Table 3.2
USED CRYPTOGRAPHIC OPERATIONS AND THEIR COMPLEXITY OVER \(E\):
\((\hat{a} \text{ and } \hat{b} \in Z_q^*)\) AND \((\hat{P} \text{ AND } \hat{Q} \in G_1)\)

| Operation | \(C_C \text{ in terms of } \mathcal{O}(|q_2| = \frac{L}{2}) \text{ bit}\) |
|-----------|--------------------------------------------------|
| Modular Multiplication (MM): \((\hat{a}\hat{b})\) | \(\mathcal{O}((\log_2 q_2)^2) \rightarrow \mathcal{O}(L^2)\) |
| Modular Inversion (MI): \((\hat{a}^{-1})\) | \(\mathcal{O}((\log_2 q_2)^2) \rightarrow \mathcal{O}(L^2)\) |
| Weil Pairing (WP) on \(G_1\): \((e(\hat{P}, \hat{Q}))\) | \(2(4(MI) + 25(MM))\mathcal{O}(\log_2 q_2) \rightarrow \mathcal{O}(L^3)\) |
| Point Addition (PA) on \(E\): \((\hat{P} + \hat{Q})\) | \(\mathcal{O}((\log_2 q_2)^2) \rightarrow \mathcal{O}(L^2)\) |
| Pollard Lambda Decryption (PD): \(dec_{sk_BGN}(n)\) | \(\mathcal{O}(\sqrt{|n|})\) |

In terms of estimated computation time, Fig. 3.4a and Fig. 3.4a confirm the complexity calculations findings. For example, as an average of six \(\lambda\) possibilities for \(n = 29\), 1.89 sec is needed to form the final key in the original model when lookup tables are used. likewise, in the original model, 1.75 min is needed in case of no lookups. Such result is obtained even though we handle the BGN inputs as a parallel
### Table 3.3
Maximum Element Size in $\Psi$ and $C_C$ per Key: Comparison

| Metric                               | Original Construction with $|A_{v,k}| = \log_2^{\lambda n}$ | Modification 1 with $|A_{v,k}| = \log_2^{\psi n}$ |
|--------------------------------------|---------------------------------------------------------------|--------------------------------------------------|
| Maximum Elements’ Size in $\Psi$    | $\lambda n \leq 32$                                           | $n \leq 32$                                      |
| Computation Complexity               | $C_C = (\lambda + 1)WP + \lambda PA$                          | $C_C1 = WP$                                     |
|                                      | $C_C + PD + O(\lambda n\log(\psi)^{\epsilon}\log(P_{BGN})^{\epsilon})$ | $C_C1 + PD + O(n\log(\psi)^{\epsilon}\log(P_{BGN})^{\epsilon})$ |
| Metric                               | Modification 2 with $|A_{v,k}| = \log_2^{\sqrt{(\lambda+1)n}}$ | Modification 3 with $|A_{v,k}| = \log_2^{\sqrt{\psi n}}$ |
| Maximum Elements’ Size in $\Psi$    | $\lambda n \leq 64$                                           | $n \leq 64$                                      |
| Computation Complexity               | $C_C2 = (\lambda + 1)WP + \lambda PA$                          | $C_C3 = WP$                                     |
|                                      | $C_C2 + PD + O\left(\frac{\log((\lambda+1)n)}{\psi}\log(P_{BGN})^{\epsilon}\right)$ | $C_C3 + PD + O\left(\frac{\log(|n|)^{\epsilon}\log(P_{BGN})^{\epsilon}}{\psi}\right)$ |

$\psi = 2$, $P_{BGN} = 1024$ bit, $L = 128$ bit, $q = 2$, $|C| = 1$ Kbit $\in G_1$, $|C| = 2$ Kbit $\in G_2$.

$\epsilon$ = small integer, $|n| = \log_2^\psi$
concatenation of 6 bit-blocks to eliminate the need for large entries. Still, it would add more value to our scheme if we can conduct its online phase computations in least amount of time to have a prompt final key formation. In comparison with the original construction, modification 2 shows a proximate performance despite the fact that it uses a different entries’ formula. This is because of the presence of the same number of pairing and multiplication computations. For example, for \( n = 29 \), it takes 4.53 sec with and 15.5 sec without lookup tables to form the final key; the only difference between modification 2 and original model is the ability to accommodate larger identities.

Since modification 2’s results are not satisfying, we turn our attention to adjustments 1 and 3 expecting they could perform better. Remarkably, when lookups are utilized, their anticipated complexity would only have the \( G_1 \) to \( G_2 \) pairings, i.e., they only need 0.07 sec to generate the veiled key regardless of which identity is in use. However, without lookups, the requirement of identity decryption and its related vector/element’s generation complicates the results leading to a longer than 0.07 sec computation time. As an example, in Fig. 3.4b when \( n = 29 \), adjustment 3 takes 8.58 sec to generate the key because of the two added complexities; still, it does not exceed 21.6 sec for a large \( n = 5000 \) value.

### 3.4.3 Communication Overhead (byte)

In terms of the Communication Overhead \( C_O \) added to each transmitted beacon, in comparison to PKI where 160 byte is used for authentication: 64 byte signature and 96 byte certificate, our scheme does not add large overhead. At the time of handshaking, vehicles need to include hashed versions of their own encrypted identities.
Figure 3.4. Key Computation Comparison
and their partners pseudonyms, i.e., 64 byte is added to the communicated beacon as in (3.4) and (3.5) for a total $C_\sigma$ of 224 byte. If the one to one mutual authentication is used, (3.6) and (3.7), a total $C_\sigma$ (including the $m_i^i$ size) of 192 byte is needed. After establishing the shared session key, pairwise data transfer needs only 48 byte in total since no PKI authentication is used as in (3.12) and (3.13). If the encrypted data transfer is used in a broadcast setup (3.14), 12 byte is added; total $C_\sigma$ of 172 byte to transfer the AES-ciphered $C_{m_i}$. Aside from these V2V and V2I communications, when the infrastructure wants to handshake with an approaching vehicle in (3.15), it has to add its encrypted identity $C_I$ that will add 128 byte to the PKI authentication, i.e., a total $C_\sigma$ of 288 byte is communicated via the I2V links. At the end of the session, a normal PKI message is broadcasted and no overhead is added as in (3.16); in case of no PKI is utilized a total beacon size of 192 byte is used as in (3.18).

In brief, our simple analysis emphasizes that the original model, in terms of final key’s storage overhead and computation time, as [10, 80] assessed, could only be applied when $\lambda n$ is between 1 to 6 bits; at best, a key storage of 2 KB to 9 KB and a computation time of 2.58 sec to 4.13 min are needed. However, through our modifications, we could grow beyond that limit. in particular, modification 3 achieves best performance such that only 77.5 msec key computation time is needed. Also, the obtained key size is 256 bit that is similar to AES keys. However, it is still larger than the ECIES’s 160 bit keys; this can be overlooked since we can trade off key size for the sake of better security.
Chapter 4: Preserving Location Privacy Using An Anonymous Authentication Dynamic Mixing Crowd

The nuisances of the available privacy preservation approaches have motivated us to devise the idea of this research track; to offer the best privacy shield for vehicles when they change their pseudonyms without the need to exploit any encryption overhead nor silent periods, we design an approach that combines the benefits of the fast pseudonyms, dynamic mix zones, and super anonymity of group signatures authentication. Our scheme allows eavesdroppers to track and reveal no vehicle identity. Our protocol utilizes the plain group signatures GS [36] and/or the hybrid GS scheme of [39] in creating the on-demand super anonymous cover for hiding the vehicles’ pseudonym-change. By doing so, we eliminate the need for the symmetric-encryption zone of [32], encryption overhead, and the communication overhead required to update each PKI’s symmetric keys set. Also, we avoid using the silent periods of [30] that do not suit the real-time VANETs. Rather, in our scheme, when any vehicle needs to change its pseudonym, it dynamically asks the neighboring crowd to surround it by an alternate authentication zone such as GS and/or hybrid hiding. To be computationally comparable to the fast pseudonym authentication, our scheme utilizes the GS and/or the hybrid authentication on-demand and only throughout the formed group’s lifetime, where vehicles are considered “relatively static to each other”. Our different
authentication usage ends when the vehicles change pseudonyms and switch back to their normal authentication. Since our vehicles use the super anonymity feature of GS only inside the created group, the communication overhead of GS can be traded off for the sake of superior anonymity; also, when we use the first optimization of [39], we obtain equivalent performance to that of the baseline pseudonyms of PKI.

4.1 Network Model and Assumptions

Our network model is mainly a PKI system as explained in 2.1.1. In our model, to achieve the goal of robust anonymous privacy, vehicles obtain the group signature cryptographic ingredients. As in the basic pseudonym approach, vehicle $v$ periodically changes its pseudonym to deceive eavesdroppers. However, such change is not successful unless $v$ is surrounded by a cluster of autos (a minimum of two vehicles). The cluster members do not mind to cooperate with $v$ by changing their pseudonyms, speeds, and lanes. Like in convoys, their change of speeds and lanes is with respect to $v$’s change of speed and lane$^8$. The reason behind such emphasis on being within a suitable context is to prevent the considered local, passive, malicious, outsider AT attacker [9] from correlating the old and new pseudonyms. When AT has several observation points on the road, he overhears $v$’s messages; if there is no other vehicle changing pseudonym at the same time, AT can easily deduce the relationship between the used $v$’s pseudonyms at various points on the road. If more than one vehicle, within the communication range of $v$, simultaneously changes pseudonyms,

$^8$Even if we assume $v$ to be stationary, the neighbors will not accelerate far from $v$. For example, assuming vehicles are traveling at 112 km/h (the approximate maximum speed on highways in the US); if a generated beacon of a 500 byte transferred at 6 Mbps rate, it needs 0.66 msec to reach the receiver ignoring any medium access contention. Adding the signing and verifying computation costs of say a 3.61 msec, a 4.27 msec is needed per beacon. For both ends of communication, per a total time of 8.54 msec, a neighbor moves only 0.265 m, i.e., it is within $v$’s communication range.
speeds, and lanes, it will be uneasy for any outsider to track the vehicles or correlate their old and new pseudonyms. In our protocol, once these neighbors cooperate and form the dynamic mix zone, they alternate their authentication and start using group signatures GS and/or hybrid authentication of [39] to hide vehicle $v$’s pseudonym change; any receiving vehicle $w$ does not recognize the identity of $v$ nor links the used group signature and pseudonym to any previously broadcasted beacon by $v$. Our protocol does not need to use any silent periods nor encryption in creating the dynamic hiding crowd; it simply utilizes the new authentication as a cover when changing pseudonyms.

### 4.2 Proposed Mixing Crowd Mechanism

As with the standard pseudonyms approach of PKI [15], in compliance with DSRC specifications [7], vehicle $v$ should change its current pseudonym approximately every 60 sec; therefore, any $pk_v^i$ would be used for around 600 broadcasts if the considered beaconing rate $T$ is 10 msg/sec. Taking such considerations into account, our scenario starts when $v$’s pseudonym is about to expire (this is what incentivizes $v$ to dynamically call for zone formation); at that moment before $v$ changes $pk_v^i$, it broadcasts a create − cover message:

$$m_v^i = \langle create - cover, duration \rangle, \text{sig}_{sk_v^i}(h_1(m_v^i)|T_v^i), \text{cer}_{CA}(pk_v^i), \quad (4.1)$$

where duration is the expected lifetime of the created zone. To make sure that $v$’s neighbors do not miss (4.1), $v$ sends more than one request. When $v$’s immediate one-hop neighbors receive the request and verify its validity, each of them ($w$ for example) switches to use its own group-signature’s secret key (assuming all vehicles
are cooperative) to sign any future $m_w^i$, leading to a group signature authentication:

$$m_w^i, \operatorname{sig}_{gsk_w}(h_1(m_w^i)|T_w^i). \quad (4.2)$$

Having all entities in the created zone use GS authentication including $v$, $v$ can freely change its pseudonym to $pk_w^{i+1}$ and starts watching neighbors hoping at least one cooperates and changes its pseudonym and then its lane and/or speed; a neighbor $w$ cooperates if its current pseudonym $pk_w^i$ is about to expire and the duration term, included in the received $v$’s create−cover message, is greater than the $pk_w^i$’s remaining lifetime. Once vehicle $w$ replaces $pk_w^i$, $w$ announces this to the zone’s neighbors by signing the new pseudonym $pk_w^{i+1}$ itself prompting the hybrid authentication of [39]:

$$m_w^{i+1}, \operatorname{sig}_{sk_w^{i+1}}(h_1(m_w^{i+1})|T_w^{i+1}), \operatorname{sig}_{gsk_w}(pk_w^{i+1}). \quad (4.3)$$

When $v$ starts receiving (4.3) before duration ends, $v$ sends couple of end−cover messages

$$m_v^{i+1} = <\text{end-cover}>, \operatorname{sig}_{sk_v^{i+1}}(h_1(m_v^{i+1})|T_v^{i+1}), \operatorname{sig}_{gsk_v}(pk_v^{i+1}) \quad (4.4)$$

to all neighbors in a hybrid authentication mode; otherwise, if no cooperation occurs before the zone’s life expires, $v$ sends an extension request with the same duration parameters using the group signature authentication as in (4.5):

$$m_v^{i+1} = <\text{extend-cover, duration}>, \operatorname{sig}_{gsk_v}(h_1(m_v^{i+1})|T_v^{i+1}). \quad (4.5)$$

When $v$’s neighbors receive the end−cover(s), they automatically revert back to their PKI authentication and dissolve the dynamic crowd. To better understand the proposed mechanism, Fig. 4.1 exemplifies a timeline followed by three vehicles $v$, $w$, and $u$ within the dynamically-created crowd representing zone’s originator,
a neighbor who changes its pseudonym, and a member who does not change any pseudonym, respectively. From Fig. 4.1, it is seen that the group signature authentication is used directly after \( v \) initiates the dynamic crowd and remains in use unless other pseudonym changes occur within the crowd; hereabouts, hybrid authentication starts to securely communicate the new pseudonyms. The \( end - cover \) sent by \( v \) also use the hybrid authentication with the new pseudonym \( pk_{v+1} \) and \( v \)'s group secret key \( g_{sk_v} \); thus, originators of \( end - cover \) messages cannot be identified. *If the vehicles simply ignore \( v \)'s \( end - cover \) messages, they use the group or hybrid signature for the rest of the network lifetime.* This does not weaken the security nor affect the computation; however, it affects the communication efficiency since extra communication payload has to be carried throughout the network lifetime. According to [39], the hybrid scheme performance with some optimization can be dramatically improved to approximate the baseline pseudonyms performance; therefore, adopting only the hybrid authentication in creating the cover becomes more efficient. In this sense, we enhance our scheme by eliminating the GS authentication leaving only the baseline pseudonym and hybrid authentications, as depicted in Fig. 4.2.

### 4.2.1 Modifying The Created Crowd

After the initial formation of any hiding zone by \( v \), vehicles might dynamically leave and join the groups. Leaving event might happen when the vehicle is revoked by CA or when it becomes outside the range of the initiator \( v \). In either case, \( w \) reverts to be a single vehicle with a PKI-authentication. On the other hand, since all vehicles are assumed to be cooperative, once any vehicle \( z \) is within the range of the initiator \( v \)'s different-authentication broadcast, it broadcasts a \( join - cover \)
Figure 4.1. The Proposed Scheme Timeline for Three Vehicles: v, w, and u
Figure 4.2. The Timeline After Eliminating The Group Signature Phase
under PKI-authentication; only \( v \) understands the request and answers it with an acknowledgement \( \text{join − ack} \). Once receiving the acknowledgment, \( z \) switches to the group signature and/or hybrid authentication. If any group member receives the \( \text{join − ack} \) message, it simply ignores such acknowledgement.

4.3 Considered Attacker, Model Validity, and Correctness Proofs

In face of eavesdropper \( AT \), presented in Chapter 3, we want to make sure that our model is valid and correct. In terms of trustability, the mixture ensures that only legitimate members can verify the received beacons since they acquired \( pk_{CA} \) and \( g_{pk_{GM}} \) at time of registration. Also, vehicles cannot deny that they sent a beacon since any message is signed using a secret key such as \( sk_v \) or \( g_{sk_v} \) that specifically belongs to the signing vehicle. The freshness and integrity of messages is guaranteed by including timestamps and hashes in beacons. Expired or misused certificates in our scheme is revoked using a mixture of PKI’s CRL and GS’s RL. The CA/GM is assumed to instantly discover and identify a misbehaved OBU. Once discovered, the central authority adds that vehicle’s credentials to the CRL+RL list that has the form of \(< \{ cer_{CA}(pk_v^{i})_{\text{serial}} \}, g_{sk_v}, revoke_{date}, CRL_{\text{generation date}} \> \). Since the authority periodically distributes the revocation list via RSUs, all of the network members will stop trusting the misbehaving vehicle. Besides these basic security requirements, the main goal of our mixing scheme is to guarantee the

- **Conditional Privacy**: although vehicles must be publicly untraceable, the CA/GM must be able to identify, track (for accountability purposes), and gather the entire
messages vehicles broadcast. Our scheme’s conditional privacy stems from inter-
mingling those of baseline, group signatures GS, and hybrid authentications. For
the former, CA easily opens the stored tuple \( < ID_v, \{ sk_v^i, pk_v^i, cer_{CA}(pk_v^i) \} > \)
to identify a liable vehicle \( v \); however, CA searches through a large database of
pseudonyms. Same procedure is applicable to group signatures where \( < ID_v, g_{sk_v} > \)
is used for finding the \( v \)’s identity. Since the hybrid authentication is a mixture of
these two, certainly the conditional privacy is achieved by using \( < ID_v, \{ sk_v^i, pk_v^i, g_{sk_v} \} > \)
to uncover the true signer identity \( v \). As a result, our scheme’s trusted
authority hosts a holistic tuple \( < ID_v, \{ sk_v^i, pk_v^i, cer_{CA}(pk_v^i) \}, g_{sk_v} > \) for mapping.

- **Unlinkability**: attacker \( AT \) passively receives beacons in the form of (2.3) from
many vehicles before the super-anonymous cover is created. However, \( AT \) cannot
verify the \( cer_{CA}(pk_v^i) \) since he has no \( pk_{CA} \), but he can recognize the different
signatures in \( cer_{CA}(pk_v^i) \). The advantage of using group and hybrid authentications
is that \( AT \) will not be able to differentiate the signatures anymore nor he can verify
signatures since he does not have the \( g_{pk_{GM}} \) key; the privacy manifests in having
any vehicle in the group signing \( m_v^i \) and the results (4.2, 4.3) are certifiable as
generated from a certain group. The identity of the generator is hidden, i.e., there
is no identity exposure and all vehicles seem to have same credentials.

The correctness of the scheme is easy to be demonstrated. By showing that every
generated signature is valid whenever the pseudonym or the group secret key is not
revoked, the mixing strategy is said to be correct as in:

\[
[pk_v^i \notin CRL] \cap [\text{verify}_{pk_v^i}(\text{sig}_{sk_v^i}(h_1(m_v^i)|T_v^i)) = \text{valid}] \cap [h_1(m_v^i) = \text{Correct}],
\]

(4.6)
\[ \text{verify}_{g_{pkGM}}(\text{sig}_{g_{skv}}(h_1(m^i_v)|T^i_v))) = \text{valid} \cap [h_1(m^i_v) = \text{Correct}] \cap [g_{skv} \notin RL], \text{and} \]

(4.7)

\[ \text{verify}_{g_{pkGM}}(\text{sig}_{g_{skv}}(pk^i_v)) = \text{valid} \cap [h_1(m^i_v) = \text{Correct}] \cap [pk^i_v \notin (CRL + RL)]. \]

(4.8)

Having presented the scheme and proved its validity, we need to numerically analyze and evaluate its effectiveness and practicality in the light of three chosen performance metrics.

4.4 Numerical Analysis and Discussion of Effectiveness

In accordance with the IEEE 1609.2 standard’s specifications [7], the ECDSA is used in our scheme for signing any \(m^i_v\); for the group signature, the short signature of Boneh and Shacham [1] is adopted. Although signature over messages has short lifetime (100 msec considering beaconing rate of 10 msg/sec) in comparison with the CA’s signature over the certificated pseudonyms as well as the GS that have longer lifespans, we follow the up-to-date elliptic curves recommendations of 256 bit keys, i.e., \(L = 128\) bit security level. An ECDSA of \(L = 128\)-bit strength is employed to sign messages; similarly, 128 bit security level is used for certificates and group signatures.

Table 4.2 [1,19,39,82] shows that ECDSA involves shorter keys and shorter signatures as well as less complex operations in comparison with GS; the scalar multiplication in ECDSA is its most time-consuming operation in comparison with group-signature of [1] where pairing is its slowest operation. We refer the reader to Table 4.1 and 3.2 for a complete list of the computational complexities of operations with respect to
$q$ (the order of the base point $G$ on the used elliptic curve) of a $2L$ bit length. By utilizing Tables 4.2, 4.1, and 3.2 besides the ECDSA and group signature’s relevant data from [39, 83, 84] over a 2.4GHz MacBook Pro machine, we could analytically assess the effectiveness of our technique in terms of the following metrics:

<table>
<thead>
<tr>
<th>Table 4.1</th>
<th>USED CRYPTOGRAPHIC OPERATIONS AND THEIR COMPLEXITY OVER $E$: $(\hat{a} \text{ and } \hat{b} \in \mathbb{Z}_q^*) \text{ and } (\hat{P} \in G)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation</strong></td>
<td>$C_{Cy}$ <strong>in terms of</strong> $O(</td>
</tr>
<tr>
<td>Modular Addition (MA): $(\hat{a} + \hat{b})$</td>
<td>$O(\log_2 q)$ → $O(L)$</td>
</tr>
<tr>
<td>Modular Exponentiation (ME): $(\hat{a}^\hat{b})$</td>
<td>$O((\log_2 q)^3)$ → $O(L^3)$</td>
</tr>
<tr>
<td>Scalar Multiplication (SM) on $E$: $(\hat{a}\hat{P})$</td>
<td>$44L(MM) → O(L^3)$</td>
</tr>
<tr>
<td>Hash (HS)$^9$: ${0, 1}^* → \mathbb{Z}_q^*$</td>
<td>Negligible</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4.2</th>
<th>ECDSA vs. GS [1]: PERFORMANCE AND KEY LENGTHS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scheme</strong></td>
<td><strong>Strength</strong> $L$ (bit)</td>
</tr>
<tr>
<td>ECDSA-256</td>
<td>128</td>
</tr>
<tr>
<td>GS [1]</td>
<td>128</td>
</tr>
</tbody>
</table>

### 4.4.1 Computation Time (msec)

Computation time refers to the time needed to sign sent messages and to verify received certificates and signatures. For example, in the baseline pseudonyms, the sender signs every sent beacon with ECDSA $L = 128$-bit signature and at the receiver side, there are two verifications, the long-term ECDSA $L = 128$-bit certificate followed by another $L = 128$-bit ECDSA verification. Considering the data

$^9$The message digest is truncated so that the bit length of the hash is the same as the bit length of $q$ (the order of $G$).
from [39], Table 4.3 shows that both of our scheme tracks (Fig. 4.1 and 4.2) incur more computation overhead throughout the created zone’s lifetime if no optimization is considered. For example, the average computation time is no less than 0.72 msec for signing and 6.5 msec for verifying in comparison with only 0.72 msec and 2.89 msec in the pseudonyms scheme. However, if we consider the [39]’s first optimization of sending only one certificate and verifying only once during the pseudonym lifetime, the normalization improves the computation; in the first track (both group signature and hybrid authentication are used (Fig. 4.1)), on average 3.25 msec is needed for signing and 5.79 msec for verifying. On the other hand, the enhanced track (Fig. 4.2) gives an approximate performance to that of the pseudonyms such as 0.73 msec is needed for signing outgoing messages and 1.55 msec is for verifying received signatures.

**Table 4.3**

**Computations Comparison Inside And Outside The Zone With And Without Optimization**

<table>
<thead>
<tr>
<th></th>
<th>Baseline Pseudonym</th>
<th>Group Signature</th>
<th>Hybrid Authentication</th>
<th>Baseline Pseudonym</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sign</td>
<td>Verify</td>
<td>Sign</td>
<td>Verify</td>
</tr>
<tr>
<td>Before Optimizing (msec)</td>
<td>0.72</td>
<td>2.89</td>
<td>5.79</td>
<td>5.06</td>
</tr>
<tr>
<td>After Optimizing (msec)</td>
<td>0.72</td>
<td>1.44</td>
<td>5.79</td>
<td>5.06</td>
</tr>
</tbody>
</table>
4.4.2 Storage Overhead (byte)

In Table 4.4, we list the predicted yearly Storage Cost $S_O$ for each authentication method. Since our scheme is a mixture of these three, we will need the highlighted colored parameters to be retained by every OBU in the network. Some of these parameters are single long-term entities while others have to be abundantly provisioned to the vehicles for their periodic short-term changing nature; if we consider that each pseudonym is used every 60 sec \cite{22} then $N = \frac{1 \text{ object}}{60 \text{ sec}} \times \frac{3600 \text{ sec}}{1 \text{ hour}} \times \frac{9 \text{ hour}}{1 \text{ day}} \times \frac{365 \text{ day}}{1 \text{ year}} = 197, 100$ objects are re-provided each year if the average daily vehicle usage is 9 hours. Thus, our scheme needs a total storage $S_O$ of 69.77 Mbyte in comparison to 25.4 Mbyte, 864 byte, and 57.1 Mbyte for baseline pseudonym, group signatures, and hybrid authentication, respectively. Since vehicles in VANETs are equipped with sufficient storage resources, 69.77 Mbyte per year is not a concern.

<table>
<thead>
<tr>
<th>Security Level $L=128$ bit</th>
<th>Object</th>
<th>Cost per Object (byte)</th>
<th>Number of Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Pseudonym</td>
<td>$sk_v^l$</td>
<td>32</td>
<td>$N$</td>
</tr>
<tr>
<td></td>
<td>$pk_v^l$</td>
<td>32</td>
<td>$N$</td>
</tr>
<tr>
<td></td>
<td>$pk_{CA}$</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>cer_{CA}(pk_v^l)</td>
<td>64</td>
<td>$N$</td>
</tr>
<tr>
<td>GS [1]</td>
<td>$g_{pk_{GM}}$</td>
<td>800</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$g_{sk_v}$</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>Hybrid Scheme</td>
<td>$sk_v^l$</td>
<td>32</td>
<td>$N$</td>
</tr>
<tr>
<td></td>
<td>$pk_v^l$</td>
<td>32</td>
<td>$N$</td>
</tr>
<tr>
<td></td>
<td>$g_{pk_{GM}}$</td>
<td>800</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$sig_{g_{sk_v}}(pk_v^l)$</td>
<td>225</td>
<td>$N$</td>
</tr>
</tbody>
</table>

$$353 \times N + 896$$
4.4.3 Maximum Number of Allowed Vehicles (With/Without Grouping)

In the basic pseudonyms authentication, with every sent beacon (of a 50 to 150 byte typical size range) there is an ECDSA-256 signature besides a 128 bit certificate of the 128 bit pseudonym itself. Therefore, from Tables 4.2 and 4.4, a total payload of 161 byte is communicated. For the group signature of [1], only one 128 bit signature is added, i.e., 225 byte; while for the hybrid authentication, one 128 bit ECDSA signature, one 128 bit group signature, and one 128 bit pseudonym are sent summing up to a total of 322 byte. This communicated payload is a concern in VANETs where the allowed data rate is between 3 and 27 Mbps; also, based on IEEE 1609 suite of standards in VANETs, the control wireless channel on which these safety beacons are transmitted is only 46 msec wide [7]. These two shortcomings limit the number of vehicles that can capture the medium. Since the baseline authentication has less overhead, it can accommodate more vehicles than the other two schemes (Fig. 4.3). What is interesting is the grouping effect, for example when group size is 4, the three schemes give comparable results when no freezing/contending is considered; in Fig. 4.4, all available leaders can capture the channel; however, in case a high priority is given to the leaders’ beacons, not all of the available members are able to capture the channel. When no priority is given to leaders, the number of capturing members and leaders curves are comparable across the three authentications, despite the fact that they are much less than the available number of vehicles.

In summary, these metrics show the effectiveness of the proposed scheme; the hiding zone does not add much overhead in comparison with the baseline authentication. Our analysis does not consider the competition for the control channel in DSRC [7];
therefore, in Chapter 5 we are building a similar grouping scenario in the presence of vehicles’ competition for the medium. Extra measures, such as lowering the beaconing rate in the formed group and reducing the authentication overhead, are taken to reduce the competition.

Figure 4.3. Schemes Comparison Before Grouping (No Freezing Delay)
Figure 4.4. Schemes Comparison After Grouping (No Freezing Delay)
Chapter 5: Grouping for Beaconing Efficiency Enhancement (G-BEE) in Vehicular Networks

To enhance the efficiency of beaconing in VANETs and to overcome the shortcomings mentioned in Section 2.3, we propose, in this joint work with Yuhui Feng and Professor Can Emre Koksal, our G-BEE cooperative beaconing strategy. G-BEE attempts to present an ideal setting for the CSMA/CA access mode; rather than having a dense network of short sessions, G-BEE makes vehicles form authenticated groups. Thus, we obtain a sparse network of longer sessions with the main load of V2I beaconing assigned to group leaders. This chapter is dedicated to describe the scheme and show how G-BEE substantially improves the performance of vehicular beaconing without degrading the level of security.

5.1 System Model, Problem Statement, and Contribution

Our network consists of \( n \) vehicles; each vehicle is equipped with an OBU. Vehicles are within the communication range of each other and are driving on a road where the infrastructure’s RSUs are optimally placed to give best coverage and least communication interference \([86]\). As it can be seen in Fig. 5.1, the channel model in

\footnote{Yuhui Feng was a graduate student at the Electrical and Computer Engineering Department, The Ohio State University at the time this work was done as part of his Masters thesis \([85]\).}
our system is a fully connected medium sharing model, in which simultaneous transmission from a pair of vehicles leads to a collision. We assume fixed transmission rate for all broadcasts at $r$ bits/sec and with a $\delta$ propagation delay. In general, the Power Gain (PG) between vehicle $v$ and vehicle $w$ is denoted as $pg_{v}^{w}$, and the power gain between vehicle $v$ and the $I$ infrastructure is denoted as $pg_{v}^{I}$. The transmission rates of vehicles are identical. We assume channels to be error-free to have less complicated analysis lest we lose insight; however, a generalization to a case with channel issues is straightforward. Each vehicle in the network periodically generates a beacon of size $R$, once every $T$ seconds, in the form of (2.3). Beacons are queued at each vehicle and transmitted when the vehicle captures medium. In this work, we consider two queuing models; in the first one, all beacons are buffered in the vehicle’s queue, vehicles need to send out its old beacon before trying to broadcast a new one, and there is no limit on the queuing buffer size. The second queuing model drops old beacons when new ones arrive, i.e., there is a finite buffer to queue beacons.

Figure 5.1. System Model
In our evaluations, we use the following performance metrics for each vehicle $v$:

- Average beacon delay $\bar{d}_v$: expected queuing delay of beacons.
- Beacon drop rate $\bar{\gamma}_v$: fraction of beacons dropped from queues.
- Beacon collision rate $\bar{\xi}_v$: fraction of beacons lost due to medium access.

Due to symmetry across vehicles, these quantities are identically distributed for all vehicles. Thus, we drop subscript $v$ for simplicity.

- Maximum availability $\mathcal{N}_C$: it is the maximum number of vehicles beyond which a finite beacon delay is not achievable.
- Throughput: it is the rate of successful beacon delivery over the channel; it can be defined as $W_v = \frac{1}{T}(1 - \bar{\xi})(1 - \bar{\gamma})$.
- Lowest achievable throughput defined as $W_{min} \triangleq \min_v(W_v)$.

In a conventional scenario (Fig. 5.1), when approaching any RSU, individual vehicles contend for CCH using CSMA/CA to beacon their statuses. The main objective of this work is to develop strategies that improve the independent vehicles’ beacons, when CSMA/CA-accessing is used in a broadcast setting; our scheme achieves:

- A higher maximum lowest achievable throughput $W_{min}$.
- Lower beacon delay $\bar{d}$ and drop rate $\bar{\gamma}$ at the same (or higher) lowest achievable throughput.
- Accommodate a higher number of vehicles at the same delay and same (or higher) lowest achievable throughput.
Our strategy space would include:

- Scheduling the vehicle that transmits at any given time.

- Deciding the route on which the transmitted beacons are forwarded to the infrastructure. Existing approaches rely on CSMA/CA to access the infrastructure directly on a single-hop V2I path. Our strategies will potentially exploit two-hop paths, combining V2V and V2I transmissions making use of the concept of grouping.

- Through our proposed G-BEE grouping strategy, we intertwine the notion of omitting/reducing the security overhead; we further enhance such reduction in order to not compromise the system security.

Having defined the system model, stated the problem we attempt to tackle, and highlighted our contribution, the following section introduces our grouping strategies.

5.2 G-BEE: Grouping for Beaconing Efficiency Enhancement

Our vehicles are PKI-entities that follow the steps of signing and verifying mentioned in 2.1.1. RSUs are no difference; they also obtain their own long-term fixed \( sk_l, pk_l, cer_{CA}(pk_l) \) security parameters. Our strategy considers forming groups in a fully distributed fashion. All vehicles in the network are considered to be cooperative and want to be within groups. The vehicles’ leaderships and memberships are decided and updated depending on channel estimations and signal strengths. In what follows, we first describe groups’ formation phase. Then, the continuously changing dynam-icity of the formed groups is outlined; we examine the (inter/intra)-communications of the groups. We prove that our scenario is secure and is not compromised when we consider authentication omission/reduction.
5.2.1 Group Formation

For a fine-grained description of this phase, we consider the example of Fig. 5.2 where every vehicle in our scheme has four parameters: Cluster Size (CS), Power Gain (PG) to represent the signal strength between any two vehicles\textsuperscript{11}, Group Power Gain (GPG) refers to the channel gain between any vehicle and the leader of its group\textsuperscript{12}, and Cluster Members Vector (CMV) to symbolize the information of the vehicle itself plus relative information of all members in its group if any. Before deployment, all vehicles are considered to be single leaders of $CS = 1$, $GPG = \infty$, and $CMV =$ Vehicle’s Identity. Our clustering scheme relies on the signal quality as a metric; we argue that the to-be-formed clusters will be sufficiently stable since we are using a gap threshold parameter. The used threshold ensures the stability of the created groups and eliminates any heavy fluctuation in group dynamics. Since vehicles prefer to be in groups, they can regularly broadcast create – group messages, i.e., they can add a create-group flag to their regular beacons and switch the flag on according to a specific rate of flag broadcast.

**Forming Group (create-group and create-ack):** Assuming vehicles $U$, $K$, and $J$ in Fig. 5.2 want to form groups, they will start beaconing create – group PKI-secured messages (5.1) that include their $CS$ and $CMV$ (5.2) parameters:

\[
\begin{align*}
< m^i_{initiator} = create – group, CMV, CS >, sig_{sk^i_{initiator}}(h_1(m^i_{initiator})) | T^i_{initiator}, cer_{CA}(pk^i_{initiator}) >.
\end{align*}
\]

\textsuperscript{11}Our assumption is that PG value is obtained from an existing related sensor.

\textsuperscript{12}GPG value is dynamically changed by the vehicle when it wants to join another group, when the vehicle sees that the channel gain between itself and the leader of the other group is greater than its intra-group GPG.
Figure 5.2. Grouping Scenario

$CMV = [initiator_{info}, member1_{\Delta initiaator_{info}}, ...]$.  

To make sure that neighbors do not miss (5.1), the initiating vehicles make sure to send more than one request. Hence, two types of vehicles receive the create – group:

- Vehicles that did not broadcast a create – group. For example, when vehicle $V$ receives the requests, $V$ would decide to join based on the signal strengths $pg_{VJ}$, $pg_{UV}$, and $pg_{VK}$ between $V$ and the requesting vehicles. For example, if $V$ wants to be within $J$’s cluster, $V$ sets its GPG to be $pg_{VJ}$ rather than $\infty$ and broadcasts create – ack message (5.3):

\[
<m_{V} = create – ack, CMV, CS, h_{1}(cer_{CA}(pk_{V}))>, \sigma_{sk_{V}}(h_{1}(m_{V})T_{V}^{i}), cer_{CA}(pk_{V}).
\]

- Vehicles that have just broadcasted a request. For example, if $U$ receives the requests of $J$ and $K$, $U$ checks the sending time of its create – group beacon by examining the $T_{U}^{i}$ timestamp of its request. Meanwhile, $U$ checks whether it received an acknowledgement from any other vehicle. If $T_{U}^{i}$ is fresh, meaning $U$ has just broadcasted a request, then $U$ ignores any other simultaneous create-group;
if $T_U^i$ is old enough and no acknowledgment has been received yet, then $U$ decides to join either $J$ or $K$ as is stated above.

Likewise, the receivers of the V’s create – ack beacons might be one of two types:

- A vehicle that just beaconed a create – group request. Such vehicle is waiting to receive an acknowledgement for its request; once received, the vehicle (e.g. $J$) validates the received acknowledgement by hashing its certified pseudonyms set $\{\text{cer}_{CA}(pk_J^i)\}$. $J$ uses such validation to discover whether the attached digest $h_1(\text{cer}_{CA}(pk_J^i))$ in (5.3) was meant to acknowledge $J$’s create – group message. If a match occurs, $J$ adds $V$ to its CMV and updates the CS size; otherwise, $J$ ignores the acknowledgement.

- A vehicle that did not send any create – group request. Such vehicle ignores any create – ack.

By the end of these back and forth broadcasts, our example in (Fig. 5.3) will have vehicles $J$ and $U$ becoming leaders of clusters with more than one member. $V$ and $Z$ are in $J$’s cluster while $O$ in $U$’s group. $K$ stayed as a single-member/single-leader group of $CS = 1$ without changing the GPG= $\infty$ value.

5.2.2 Group Dynamics and Modifications

Vehicles might willingly want to leave or in other instances desire to join groups even when they miss the initial requests made by other vehicles\textsuperscript{13}. Such modifications can be easily acted upon since our dynamic groups are founded on a voluntarily commitment and cooperation basis from the good neighbors of the initiating vehicles:

\textsuperscript{13}The created groups can be forcibly altered when any vehicle is evicted.
Figure 5.3. After Group Formation and Several Broadcasts: CMV of members is updated because of the received leaders’ beacons.

- **Leaving Group**: for any member vehicle $O$, in $U$’s group for example, $O$ might:

  - Unwillingly leave because it accelerated its speed and became outside the range of $U$’s beaconing:
    \[
    GPG_O < \text{Threshold}_\text{Leaving}.
    \]  
    (5.4)

  - Willingly leave because it receives a stronger signal from another group leader (e.g. from $J$) (5.5):
    \[
    GPG_O < \text{Threshold}_\text{Leaving} < p_{OJ}^J.
    \]  
    (5.5)

In either case, $O$ reverts back to be a single vehicle with $CMV = O$. Upon missing $O$’s beaconing for a certain period of time, $U$ considers $O$ outside its group, decreases $U$’s CS by 1, and updates its CMV. Leaders could also leave their groups unwillingly when they become far from the members who cannot receive their beacons anymore. For example, $U$ could leave when $p_{OJ}^U < \text{Threshold}_\text{Leaving}$. As a consequence, the members of $U$’s group switch back to be single vehicles, restore their before-deployment settings, and try to create or join groups.
• **Joining Group (join-group):** to join a group after missing the initial leaders’ create – group requests, any approaching vehicle, whether being a single or a member of another group, would join if the following joining condition is satisfied:

\[
p_{\text{vehicle}}^{\text{new leader}} > G_{\text{vehicle}} + \text{Threshold}_{\text{Joining}}.
\]  

If we consider vehicle \(O\) in our example, once (5.6) is satisfied, \(O\) decides to join an already created group (of \(J\)) by broadcasting a join – group request:

\[
< m_{O}^{\text{join - group}}, \text{CMV}, \text{CS}, h_{1}(\text{cer}_{CA}(pk_{J}^{i})), \text{sig}_{sk_{O}}(h_{1}(m_{O}^{i})T_{O}^{i}), \text{cer}_{CA}(pk_{O}^{i})>.
\]  

(5.7)

Only leader \(J\) recognizes the request since (5.7) includes \(h_{1}(\text{cer}_{CA}(pk_{J}^{i}))\). \(J\) first checks the signal strength of \(O\):

\[
p_{J}^{O} > \text{Threshold}_{\text{Permission}};
\]  

(5.8)

if (5.8) is satisfied, \(J\) automatically allows \(O\) to join \(J\)’s group by including \(O\) in its CMV and by incrementing \(J\)’s CS by 1 without any need to acknowledge that to \(O\). After receiving the new beacons of \(J\), \(O\) updates its CMV accordingly. Leaders do not join other groups unless their clusters are completely demolished when they themselves leave their groups or after their groups’ members leave.

With all measures considered, the leave and join activities do not require the vehicles to conduct any computations; yet, vehicles still need to broadcast extra messages as we have seen earlier.

**5.2.3 Addressed G-BEE Communication Scenario**

After thoroughly explaining group formation, dynamics, and modifications, we examine the (inter/intra)-communications of the created groups. Before grouping,
in a regular V2I situation, many individual vehicles contend to capture the single CCH channel. The purpose behind such contention is to directly communicate their information (large beacons) to the infrastructure over the one-hop links. With more vehicles, the infrastructure witnesses higher contention and lower throughput. Therefore, the main goal is to transform the many short-sessions (when high user-population communication settings are considered) of the existing approaches into fewer long-sessions of few active leaders. By dedicating the V2I communication to group leaders (assigning a high priority to their beacons\textsuperscript{14}), we allow all vehicles’ information to reach the infrastructure over a two-hop path through leaders (Fig. 5.4). This increases the network throughput significantly in 802.11p; the individual vehicles, which were not able to communicate their information to the infrastructure because of the limited channel capacity, are now able to send their information through leaders. Furthermore, since group members are relatively static to each other and highly mobile with respect to the stationary infrastructure, group leaders can use the relative members’ information in their beacons rather than whole members’ beacons. Thus, $I$ obtains an appropriate view of the network using the included short information of members even if such information is not up-to-date; beacon drop and collisions, which occur in the intra-group communication, do not affect the quality of the V2I leaders-sessions.

At the time of group formation/joining and any inter-group communication, any broadcasted beacon has to be fully PKI-secured so as to not compromise security. In other words, all beacons are in the form of (2.3) having attached the signature and the certificate parts. However, inside the groups themselves, we can safely lower the

\textsuperscript{14}Although RSUs still receive beacons from non-leaders, RSUs can be programmed to first process leaders beacons that have GPG= $\infty$. 

application layer overhead without breaching the security of the intra-group communications; since the authentication has already been ensured between group members and leaders at the time of joining groups, the authentication does not need to be renewed too frequently as long as members stay connected to their leaders. In Table 5.1, we enlist all possible forms of exchanged beacons before grouping, at moment of forming or joining groups, and after grouping with several types of authentication omissions/reduction for groups’ members and leaders. Our strategy utilizes a combination of such types of reductions/omissions leading to two versions of G-BEE: simple and enhanced. In the simple version, each group leader merely combines the received members’ beacons and sends them to $I$. The leader eliminates the repetition of the members’ headers of size $R_H$. To enhance such simple grouping form, an advanced form of G-BEE is presented; in which the CMV part (highlighted cells in Table 5.1) of the leader beacons is used where only hashed certificates of the group members are
kept. In this advanced case, following the DSRC specifications [7] of using the SHA-256 hash functions, size of certificates is reduced remarkably. Because of omissions, we need to prove that security is still achieved in the presence of the \( \mathcal{AT} \) attacker of Chapter 3.

### 5.2.4 Security Fulfillment, Functional Correctness, and Attacker Scenarios

Since G-BEE relies on the baseline PKI-pseudonyms authentication, it is easy to show its fulfillment of the needed VANETs security aspects, in both V2V inter-group and V2V/V2I intra-cluster communications. In Table 5.2, 5.3, and 5.4, we outline G-BEE security proofs at all phases of vehicular interactions. These proofs can straightforwardly prove the correctness of the scheme by merely fulfilling

\[
(p_{k_{v}}^{i} \notin CRL) \cap (\text{verify}_{p_{k_{v}}^{i}}(\text{sig}_{s_{k_{v}}}(h_{1}(m_{v}^{i})T_{v}^{i})) = \text{valid}) \cap (h_{1}(m_{v}^{i}) = \text{Correct}) \cap (h_{1}(\text{cer}_{CA}(p_{k_{v}}^{i}))) = \text{Correct}) \; (5.9)
\]

when omission is considered, we add a fourth condition of correctness to (5.9), by which we make sure the received certificate digest \( h_{1} (\text{cer}_{CA}(p_{k_{v}}^{i})) \) is correct. Furthermore, for CMV, validating that \( \{h_{1}(\text{cer}_{CA}(p_{k_{v}}^{i})))$$|_{1 \leq k \leq CS}\} = \text{Correct} \) is a necessity and can be added to (5.9) as well. On the other hand, these proofs can be used to show that G-BEE withstands attacker \( \mathcal{AT} \); since \( \mathcal{AT} \) did not register with CA before deployment, he does not possess the CA’s \( p_{k_{CA}} \) and he has no certified keys. Such an outsider can be easily spotted and isolated when he tries to breach into the network in one of the following ways:

- \( \mathcal{AT} \) passively receives other entities’ beacons: when \( \mathcal{AT} \) receives a beacon in the form of (2.3), he cannot verify the \( \text{cer}_{CA}(p_{k_{v}}^{i}) \) part since he does not have \( p_{k_{CA}} \).
Table 5.1
Exchanged Beacons And Communication in Our Scheme With Several Authentication’s Omission Types

<table>
<thead>
<tr>
<th>Before Grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt; m^i_{vehicle} = (CMV, CS) &gt;, \text{sig}<em>{sk</em>{vehicle}}(h_1(m^i_{vehicle})</td>
</tr>
<tr>
<td>CMV = [vehicle_{info}]</td>
</tr>
<tr>
<td>Group Formation and Joining</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Leader CMV = [leader_{info}]</td>
</tr>
<tr>
<td>(&lt; m^i_{leader} = (create - group, CMV, CS) &gt;, \text{sig}<em>{sk</em>{leader}}(h_1(m^i_{leader})</td>
</tr>
<tr>
<td>Member CMV = [member_{info}]</td>
</tr>
<tr>
<td>(&lt; m^i_{member} = (create - ack, CMV, CS, h_1(cer_{CA}(pk^i_{leader}))) &gt;, \text{sig}<em>{sk</em>{member}}(h_1(m^i_{member})</td>
</tr>
<tr>
<td>(&lt; m^i_{member} = (join - group, CMV, CS, h_1(cer_{CA}(pk^i_{leader}))) &gt;, \text{sig}<em>{sk</em>{member}}(h_1(m^i_{member})</td>
</tr>
<tr>
<td>After clustering</td>
</tr>
<tr>
<td>Leaders Beacons with Three CMV Options:</td>
</tr>
<tr>
<td><strong>CMV = [leader_{info}]</strong></td>
</tr>
<tr>
<td>(&lt; m^i_{leader} = (CMV, CS) &gt;, \text{sig}<em>{sk</em>{leader}}(h_1(m^i_{leader})</td>
</tr>
<tr>
<td>([leader_{info}, \sum_{k=1}^{CS} &lt; member_{k_{info}}, \text{sig}<em>{sk</em>{member_k}}(h_1(m^i_{member_k})</td>
</tr>
<tr>
<td>Members Beacons with Same CMV and Four Security Overhead Options:</td>
</tr>
<tr>
<td><strong>CMV = [member_{info}]</strong></td>
</tr>
<tr>
<td>(&lt; m^i_{member} = (CMV, CS) &gt;, \text{sig}<em>{sk</em>{member}}(h_1(m^i_{member})</td>
</tr>
<tr>
<td>After sending the first certificate, keep its digest in the subsequent Beacons for pseudonym’s lifetime:</td>
</tr>
<tr>
<td>(&lt; m^i_{member} = (CMV, CS) &gt;, \text{sig}<em>{sk</em>{member}}(h_1(m^i_{member})</td>
</tr>
<tr>
<td>After sending the first certificate, remove it from the subsequent Beacons for pseudonym’s lifetime:</td>
</tr>
<tr>
<td>(&lt; m^i_{member} = (CMV, CS) &gt;, \text{sig}<em>{sk</em>{member}}(h_1(m^i_{member})</td>
</tr>
</tbody>
</table>
• **\(\mathcal{AT}\) tries to replay the received beacons:** if \(\mathcal{AT}\) tries to resend the received beacons, other vehicles can easily spot the masquerading act using the attached \(T^i_u\) that does not match the current session timeline.

• **\(\mathcal{AT}\) creates his own \(sk^i_{\mathcal{AT}}, pk^i_{\mathcal{AT}}\) keys:** \(\mathcal{AT}\) creates his own keys and broadcasts his own beacon \(m^i_{\mathcal{AT}}, \text{sig}_{sk^i_{\mathcal{AT}}} \left( h_1(m^i_{\mathcal{AT}})T^i_{\mathcal{AT}} \right) \), \(cer_{\mathcal{AT}}(pk^i_{\mathcal{AT}})\) with a fabricated certificate \(cer_{\mathcal{AT}}(pk^i_{\mathcal{AT}}) = \text{sig}_{sk^i_{\mathcal{AT}}}(pk^i_{\mathcal{AT}})\). When network entities receive such beacon, they will not be able to use the CA’s \(pk_{CA}\) to verify the attached certificate and therefore easily identify \(\mathcal{AT}\) as being an outsider.

Having explained the scheme and analyzed its security, next section introduces our technical approach through which we can assess the gain of the presented G-BEE grouping strategy.

### 5.3 Performance Analysis

To analyze the performance of the presented strategy, we will derive the performance metrics defined in Section 5.1 by building three stochastic models. We build a Bianchi model characterizing the CSMA/CA broadcast to derive the beacon collision probability \(\bar{\xi}\). To describe each vehicle’s behavior in generating and broadcasting beacons, D/M/1 and D/M/1/1 queues are used to model the scenarios in which old beacon packets are kept and dropped, respectively. The D/M/1 queue assumes a deterministic beacon generation rate \(T\) and exponential service time \(\mu\), but infinite buffer to obtain the average beacon delay \(\bar{d}\); since this is unrealistic and the beacons suffers drop, we further develop the D/M/1/1 queue to derive the average beacon drop rate \(\bar{\gamma}\). Furthermore, in our analysis, the throughput \(W_v\) metric is precisely
Table 5.2
G-BEE Security Fulfillment Proofs

- Before grouping,
- At time of group formation and joining,
- After grouping for members intra-group V2V communications with no overhead-omission, and
- After grouping for leaders inter-group V2I without omitting overhead from the leaders beacons nor from their CMV

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Sending Vehicle $v$</th>
<th>Receiving Vehicle/Infrastructure $w$ (If $w$ is RSU, $w$ does the steps for the leader beacon and every member included in CMV)</th>
</tr>
</thead>
</table>
| Authentication and Trust         | • Send $cer_{CA}(pk_v^i)$                                                          | • Verify $cer_{CA}(pk_v^i)$ with $pk_{CA}$  
• Check $pk_v^i$ against CRL to prove that it is not a revoked certificate |
| Non-repudiation                  | • Send $sig_{sk_v}(h_1(m_v^i)|T_v^i)$                                              | • Verifying $sig_{sk_v}()$ with $pk_v^i$ proves that $v$ has an exclusive key related to $pk_v^i$ |
| Beacon Freshness                 | • $T_v^i$ is attached by $v$                                                        | • $w$ is sure that $m_v^i$ was not replayed |
| Beacon Integrity                 | • $h_1(m_v^i)$ accompanies $m_v^i$                                                  | • $w$ hashes $m_v^i$ and compares its digest with $h_1(m_v^i)$; if equal means the message has not been altered |
Table 5.3
G-BEE Security Fulfillment Proofs (continued 1)

- After grouping for intra-group V2V when the members beacons keep hashed certificate for a pseudonym’s lifetime \( m_v = (CMV, CS) \), \( \text{sig}_{sk_v}(h_1(m_v) | T_v^i) \), \( h_1(cer_{CA}(pk_v^i)) \) after sending the first certificate

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Sending Member ( v )</th>
<th>Receiving Member/Leader ( w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust and Authentication</td>
<td>• Send ( cer_{CA}(pk_v^i) ) at the beginning of ( pk_v^i )'s lifetime</td>
<td>• Verify ( cer_{CA}(pk_v^i) ) with ( pk_{CA} )</td>
</tr>
<tr>
<td></td>
<td>• Followed by ( h_1(cer_{CA}(pk_v^i)) ) attached in subsequent messages</td>
<td>• Check ( pk_v^i ) against CRL to prove that it is not revoked</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Hash ( cer_{CA}(pk_v^i) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Compare the received digest with ( h_1(cer_{CA}(pk_v^i)) )</td>
</tr>
<tr>
<td></td>
<td>Similar (Non-repudiation, Message Freshness, and Integrity) Checks</td>
<td></td>
</tr>
</tbody>
</table>

- After grouping for intra-group V2V when the members beacons remove certificate for a pseudonym’s lifetime \( m_v = (CMV, CS) \), \( \text{sig}_{sk_v}(h_1(m_v) | T_v^i) \) after sending the first certificate

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Sending Member ( v )</th>
<th>Receiving Member/Leader ( w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust and Authentication</td>
<td>• Only send ( cer_{CA}(pk_v^i) ) at the beginning of ( pk_v^i )'s lifetime</td>
<td>• Verify ( cer_{CA}(pk_v^i) ) with ( pk_{CA} )</td>
</tr>
<tr>
<td></td>
<td>• Subsequent messages only has ( \text{sig}_{sk_v}() )</td>
<td>• Check ( pk_v^i ) against CRL to prove that it is not revoked</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Still having the same ( pk_v^i ) to validate ( \text{sig}_{sk_v}() )</td>
</tr>
<tr>
<td></td>
<td>Similar (Non-repudiation, Message Freshness, and Integrity) Checks</td>
<td></td>
</tr>
</tbody>
</table>
Table 5.4  
G-BEE Security Fulfillment Proofs (continued 2)

- After clustering for leaders inter-cluster V2I when no overhead is omitted from their beacons \( m^i_v = (CMV, CS) \), \( \text{sig}_{sk_v}(h_1(m^i_v)T^i_v) \), \( cer_{CA}(pk_v^i) \) but from the included \( CMV = [v_{info}, \sum_{k=1}^{CS} \text{member}_{kinfo}, h_1(cer_{CA}(pk_{\text{member}_k}))] \) keeping only hashed certificates of members. Since other vehicles ignore CMV verification when receiving leaders beacons, we only look at V2I links

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Sending Leader ( v )</th>
<th>Receiving Infrastructure ( w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust and Authentication of Leaders</td>
<td>• Send ( cer_{CA}(pk_v^i) )</td>
<td>• Verify ( cer_{CA}(pk_v^i) ) with ( pk_{CA} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Check ( pk_v^i ) against CRL to prove that it is not revoked</td>
</tr>
<tr>
<td></td>
<td>Similar (Non-repudiation, Message Freshness, and Integrity) Checks</td>
<td></td>
</tr>
<tr>
<td>Authentication of CMV</td>
<td>• Send ( {h_1(cer_{CA}(pk_{\text{member}_k}))</td>
<td>1 \leq k \leq CS} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Parse the table to validate all received digests ( {h_1(cer_{CA}(pk_{\text{member}_k}))</td>
</tr>
</tbody>
</table>

- After clustering for leaders inter-group V2I with no overhead is omitted from their beacons but completely from the included \( CMV = [v_{info}, \sum_{k=1}^{CS} \text{member}_{kinfo}] \)

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Sending Leader ( v )</th>
<th>Receiving Infrastructure ( w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication of CMV</td>
<td>• Send nothing in CMV</td>
<td>• In this case, ( w ) just knows about the status of CS vehicles in the group of ( v )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Disadvantage: ( w ) cannot know whether any of CMV members are revoked</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Sending Leader ( v )</th>
<th>Receiving Infrastructure ( w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similar Authentication of Leaders, Non-repudiation, Freshness, and IntegrityChecks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
developed. For analytical tractability, in our analysis, we make the following two approximations:

- we assume that the packet service times are exponentially distributed, upon arriving at the head of the queue.
- to accurately model CSMA/CA, we need a giant two-dimensional Markov chain to represent the system; linking vehicles’ states together as a grid. However, because of the associated giant state space, we choose to break the chain into individual chains and analyze each user independently.

### 5.3.1 Collision Rate

In our one-dimension Bianchi Markov model [87] (Fig. 5.5), the state of each node in the chain, denoted by \( k \in (0, W_0 - 1) \), represents the backoff counter value. Idle State \( \bar{I} \) occurs when the capturing vehicle has no new beacon waiting for transmission in its buffer. The vehicle goes into \( \bar{I} \) state to allow other vehicles to compete for the channel. \( Pr_0 \) is the stationary probability that a vehicle’s buffer is empty, and \( Pr_I \) is the stationary probability that the channel is sensed idle by a vehicle. Considering \( \bar{b}_i \) to be the stationary probability for each backoff counter state in the chain and \( \bar{b}_I \) to be the stationary probability of state \( \bar{I} \), then

\[
\bar{b}_0 = \frac{(\bar{b}_0 + Pr_0)(1 - Pr_0)}{W_0} + \bar{b}_1 Pr_1, \tag{5.10}
\]

\[
\bar{b}_{W_0-1} Pr_I = \frac{(\bar{b}_0 + Pr_0)(1 - Pr_0)}{W_0}, \tag{5.11}
\]

\[
\bar{b}_k Pr_I = \frac{(\bar{b}_0 + Pr_0)(1 - Pr_0)}{W_0} + \bar{b}_{k+1} Pr_I, \quad k \in [1, W_0 - 2], \tag{5.12}
\]
\[ \bar{b}_f + \sum_{i=0}^{\infty} \bar{b}_i = 1. \]  
(5.13)

From (5.10)-(5.13) and assuming a total number of vehicles to be \( n \), we can obtain the following expression for \( Pr_I \), the probability that no vehicle is beaconing:

\[ Pr_I = (1 - \bar{b}_0)^n. \]  
(5.14)

Similarly, \( \bar{b}_0 \), the probability that a vehicle is transmitting a beacon, can be calculated from (5.10) as:

\[ \bar{b}_0 = \frac{2Pr_f(1 - Pr_0)}{(W_0 - 1)(1 - Pr_0) + 2Pr_f}. \]  
(5.15)

Hence, from (5.14) and (5.15), beacon collision probability \( \tilde{\xi} \), can be easily derived since \( \tilde{\xi} \) represents the probability of beacons colliding when being broadcasted during the same transmission timeslot:

\[ \tilde{\xi} = 1 - Pr_I = 1 - (1 - \bar{b}_0)^n. \]  
(5.16)

---

Figure 5.5. One Dimensional Bianchi Markov Model for CSMA/CA
5.3.2 Average Beacon Delay

To characterize beacons generation and broadcast in each vehicle, we build a discrete time D/M/1 queue. In comparison to [46–50], our D/M/1 infinite buffer queue (Fig. 5.6) has a periodic beacon generation rate $T$, i.e., a deterministic arrival time. For the sake of analytical tractability, a memoryless exponential beacon service rate $\mu$ is used.

![Figure 5.6. D/M/1 Model for Each Vehicle’s Beacons Buffer](image)

In Fig. 5.6, $\bar{v} \in (0, 1, 2, \cdots)$ represents the state of the each node, i.e., refers to the number of beacons in the vehicle buffer. $Pr_{ij}$ represents the transitions probability and $\Pi_i$ is the stationary probability of each state $\bar{v}$. Then, $Pr_{ij}$ can be found as:

$$
Pr_{ij} = \begin{cases} 
0 & i < j - 1 \\
\frac{\mu T^{i+1-j}}{(i+1-j)!} e^{-\mu T} & i \geq j - 1, j \neq 0 \\
1 - \sum_{j=1}^{\infty} Pr_{ij} & j = 0
\end{cases}
$$

where the stationary probability has the property of

$$
\sum_{k=0}^{\infty} \Pi_k = 1.
$$

Next, we combine the Bianchi’s model analyzed in the previous section with the D/M/1 model to evaluate $Pr_0$. This will provide us with the exact solution of the steady state probabilities of both Markov chains. To that end, we set $\Pi_0 = Pr_0$,
since both represent the probability of a node being idle. To derive $\Pi_0$, we use the Laplace-Stieltjes transform [88] to solve the following stationary state probabilities (obtained from (5.17)):

$$\Pi_j = \sum_{i=0}^{\infty} \frac{\mu^i T^i}{i!} e^{-\mu T} \Pi_{i+j-1}, \quad j = 1, 2, 3, ...$$  \hspace{1cm} (5.19)

and

$$\Pi_0 = \prod_{i=0}^{\infty} Pr_i \Pi_i.$$  \hspace{1cm} (5.20)

The solution of (5.19) and (5.20) has the following form:

$$\Pi_k = (1 - \omega) \omega^k, \quad k = 0, 1, 2, 3, ...$$  \hspace{1cm} (5.21)

where

$$\omega = e^{-(1-\omega)\mu T}.$$  \hspace{1cm} (5.22)

When the Markov chain is stable, the unique solution of $\omega$ is in the range $(0, 1)$ and $\Pi_0$ equals:

$$\Pi_0 = Pr_0 = 1 - \omega.$$  \hspace{1cm} (5.23)

By employing Little’s law, the expectation of the end-to-end delay $\bar{d}$, which is the time interval from the moment of beacon being generated until it is being received by receiver, can be found as\(^\text{15}\):

$$\bar{d} = \frac{1}{\Pi_0 \mu}.$$  \hspace{1cm} (5.24)

Since the average waiting time in the system includes the average service time $\frac{1}{\mu}$, we can derive $\frac{1}{\mu}$ as:

$$\frac{1}{\mu} = t_b + t_f + t_r = \beta E_c + t_r \left( \frac{1}{P_r} - 1 \right) E_c + t_r,$$  \hspace{1cm} (5.25)

\(^\text{15}\)As is defined in Section 5.1, the maximum availability $N_c$ metric will automatically be found using the average beacon delay $\bar{d}$.  

90
where $t_b$ is the average backoff time, $t_f$ is the average freezing time, $E_c$ is the expected value of backoff counter that equals $\frac{(W_0-1)}{2}$ for broadcast, $\beta$ is the timeslot duration within the backoff time, $R$ is beacon size, $r$ is the data transmission rate, $\delta$ is the propagation delay, DIFS is the Distributed Inter-Frame Space time, and $t_r$ is the transmission time that is calculated as:

$$t_r = \frac{R}{r} + DIFS + \delta. \quad (5.26)$$

### 5.3.3 Beacon Drop Rate

Due to the real-time nature of VANETs’ beaconing, if a new beacon is generated and a previous one has not been broadcasted yet, then it may be preferable to have the new beacon replaces the old one. Since there is no need to hold more than a single updated beacon in the queue, we use a D/M/1/1 queue (Fig. 5.7) to analyze the rate of beacon drops due to the arrival of the new updated beacons.

![Figure 5.7. D/M/1/1 Model for Each Vehicle’s Beacons Buffer](image-url)
Using the same notation of the D/M/1 queue, the transition probabilities can be written as:

\[
\begin{align*}
Pr_{00} &= \mu T e^{-\mu T}, \\
Pr_{01} &= 1 - e^{-\mu T}, \\
Pr_{10} &= 1 - (1 + \mu T) e^{-\mu T}, \\
Pr_{11} &= (1 + \mu T) e^{-\mu T}, \text{ and}
\end{align*}
\]

where \(Pr_{00}\) is the probability of at least one beacon is served and is departing, \(Pr_{01}\) is the probability that no beacon is served, \(Pr_{10}\) is the probability that at least two beacons are served and are departing, and \(Pr_{11}\) refers to the probability of having either no or one beacon served and departing. Along with \(\Pi_0 + \Pi_1 = 1\), one can find:

\[
Pr_0 = \Pi_0 = 1 - \frac{e^{-\mu T}}{1 - \mu T e^{-\mu T}}. \tag{5.28}
\]

Hence, the average drop rate \(\bar{\gamma}\) is found to be:

\[
\bar{\gamma} = Pr(\text{drop}) = \Pi_1 \cdot Pr_{01} = \frac{e^{-2\mu T}}{1 - \mu T e^{-\mu T}}. \tag{5.29}
\]

### 5.3.4 Throughput

The rate of successful beacon delivery over the channel in the classical individual scenario can be simply formulated as:

\[
W_v = \frac{1}{T} (1 - \bar{\xi})(1 - \bar{\gamma}). \tag{5.30}
\]

However, with the G-BEE clustering strategy where we have different roles for vehicles, this \(W_v\) will have two forms:

- A **V2I** throughput \(W_v\) defined as the number of valid not-necessarily-up-to-date vehicles’ beacons that the infrastructure obtains during each beacon generation
period $\mathcal{T}$. As a function of the drop and collision rates, $W_v$ can be found as:

$$W_v = n \cdot (1 - \bar{\gamma}) \cdot (1 - \bar{\xi}).$$  

(5.31)

Since drop and collisions can occur at the intra-cluster communication level, $W_v$ does not exhibit only the most up-to-date information. Therefore, the following throughput metric is needed.

- **An Updated G-BEE Throughput** $W_u$ defined as the number of valid up-to-date vehicles’ beacons that the infrastructure obtains from the system during each beacon generation period $\mathcal{T}$. For a total number of $n$ vehicles with $\mathcal{N}_L$ leaders/clusters each of $CS = \frac{n}{\mathcal{N}_L}$, $W_u$ is:

$$W_u = \mathcal{N}_L(1 - \bar{\gamma}_{\text{leader}})(1 - \bar{\xi}_{\text{leader}})[(\frac{n}{\mathcal{N}_L} - 1)(1 - \bar{\gamma}_{\text{member}})(1 - \bar{\xi}_{\text{member}}) + 1],$$  

(5.32)

where $\bar{\gamma}_{\text{leader}}, \bar{\xi}_{\text{leader}}, \bar{\gamma}_{\text{member}},$ and $\bar{\xi}_{\text{member}}$ are the drop rate and collision probability of any leader and any member, respectively.

### 5.4 Numerical Evaluations

Our evaluations are merely based on the analytical tools that we developed in Section 5.3. A reader may argue that a high-quality simulation approach would be a more realistic evaluation scenario. As a response, we emphasize that although many assumptions are involved, our analysis is built on a solid proven mathematical basis; it gives a general description of our system for any parameters values. By objectively observing the two approaches, for now, the analytical solution is sufficient and it serves our accuracy needs. Table 5.5 summarizes the parameters values set in our evaluation. Thorough discussions on the choice of these values can be found in [7,14, 93]
We assume a network of $n$ vehicles being evenly/optimally divided/partitioned into $N_L$ clusters of $N_L$ leaders; each cluster of size $CS = \frac{n}{N_L}$ and $(CS - 1)$ members. As we stated earlier in Section 5.2, we consider two grouping versions: simple and enhanced G-BEE. In the simple version, the total beacon size that the leader conveys to the infrastructure is $\frac{n}{N_L} R - (\frac{n}{N_L} - 1)R_H$ due to combining all of the $\frac{n}{N_L}$ group participants’ beacons of size $R$ excluding the beacon’s headers $R_H$ of the members. On the other hand, in the enhanced version, the total beacon size that the leader conveys to the infrastructure is $R + (\frac{n}{N_L} - 1)(R - R_H - R_{Signature} - R_{Certificate} + R_{Hash})$ due to replacing the beacons’ header overhead $R_H$, certificate overhead $R_{Certificate}$, and signature overhead $R_{Signature}$ with a hashed version of each member’s certificate $R_{Hash}$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beacon Generation Period $T$</td>
<td>100 msec</td>
</tr>
<tr>
<td>Contention Window $W_0$</td>
<td>512 µsec</td>
</tr>
<tr>
<td>Backoff Timeslot $\beta$</td>
<td>13 µsec</td>
</tr>
<tr>
<td>DIFS</td>
<td>58 µsec</td>
</tr>
<tr>
<td>Propagation Delay $\delta$</td>
<td>1 µsec</td>
</tr>
<tr>
<td>Data Transmission Rate $r$</td>
<td>6 Mbps</td>
</tr>
<tr>
<td>Total Beacon Size (including header) $R$</td>
<td>500 byte</td>
</tr>
<tr>
<td>Total (Physical and MAC) Header Size $R_H$</td>
<td>50 byte</td>
</tr>
<tr>
<td>Security Level $L$</td>
<td>128 bit</td>
</tr>
<tr>
<td>Signature Size $R_{Signature}$</td>
<td>64 byte</td>
</tr>
<tr>
<td>Certificate Size $R_{Certificate}$</td>
<td>96 byte</td>
</tr>
<tr>
<td>Hash Size $R_{Hash}$</td>
<td>32 byte</td>
</tr>
</tbody>
</table>

To assess the gain of our G-BEE schemes in comparison with the individual case where every vehicle tries to directly communicate its status to the infrastructure, we utilize the maximum availability $N_C$, average delay $\bar{d}$, collision probability $\bar{\xi}$, drop rate $\bar{\gamma}$, and throughput performance metrics we derived in Section 5.3. By plotting the curves with respect to the number of vehicles $n$ considering $N_L = 6$ evenly formed
groups, Fig. 5.8 shows that both G-BEE versions enact better effectiveness than the original individual case; the enhanced G-BEE is even more efficient than the simple grouping version since lower beacon size is communicated to the infrastructure. With grouping in Fig. 5.8a, we immediately notice an increase in the number of vehicles that can collectively capture the CCH channel without exceeding a finite average delay $\bar{\bar{d}}$. Through analysis, the maximum limit $N_C$ on $n$ is 64 when no grouping is considered (Fig 5.8a); beyond this 64, the vehicles’ queues become unstable and the average delay $\bar{\bar{d}}$ goes to infinity. Interestingly, both versions of our G-BEE scheme with and without drop increase this maximum availability $N_C$ to 72 and 84 vehicles respectively achieving a gain of 12.5% and 31.25% in comparison with the individual case. Moreover, the enhanced G-BEE achieves 40% improvement in comparison with the individual case when the same delay $\bar{\bar{d}}$ of 1 sec is considered. For the same number of vehicles, e.g $n = 62$, an average delay $\bar{\bar{d}}$ of 0.5 sec, 0.3 sec, 0.18 sec is experienced for the individual case, simple G-BEE, and enhanced G-BEE, respectively. Then in Fig. 5.8b we inspect the collision rate $\bar{\bar{\xi}}$’s gain. We see that grouping lowers $\bar{\bar{\xi}}$; both G-BEE versions enact an equal performance. For a low-density network of $n = 22$ and cluster size $CS = \frac{n}{6}$, a 25% gain is obtained, while for a high density network of 102 vehicles, 50% less collisions occur when G-BEE is used in comparison with the no grouping context. The same narration is applicable to the drop rate $\bar{\bar{\gamma}}$ where the enhanced G-BEE strategy outperforms both the individual and simple grouping cases with a 60% gain at high network density of $n = 102$.

Next, we show the effectiveness of the simple and the enhanced G-BEE with regard to the two derived throughput metrics $W_v$ and $W_u$. In Fig. 5.9, we plot these successful beacon delivery rates with respect to the number of vehicles $n$. A first observation
Figure 5.8. Delay Rate $\bar{d}$, Collision Rate $\bar{\xi}$, and Drop Rate $\bar{\gamma}$ Metrics Comparison for Individual No Grouping Scenario, Simple G-BEE, and Enhanced G-BEE for $N_G = 6$ Evenly Formed Groups.
is that the system throughput, whether before or after G-BEE, gradually increases with $n$ until it converges to a certain limit. Under the considered group size $CS = \frac{n}{6}$ for any $n$, both simple and enhanced G-BEE exhibit better throughput efficiency than the no grouping case. In both, the $W_v$ throughput, from the infrastructure’s perspective, exceeds the updated $W_u$ throughput, from the vehicles’ perspective, and the individual case throughput. This occurs because of the reduced number of active users that communicate with the infrastructure, hence significantly increasing the 802.11p network throughput. For example, when $n = 112$, the simple and enhanced G-BEE’s $W_v$ gain is 15% and 33%, respectively. Meanwhile, the $W_u$ in the enhanced G-BEE gives only 10% improvement and it approximates the individual case when simple G-BEE is considered. The reason is that $W_v$ is not as accurate as $W_u$; $W_v$ takes into account even the not-necessarily-up-to-date information delivered to the infrastructure. Therefore, $W_v$’s value\textsuperscript{16} is higher than $W_u$ that considers only the latest valid vehicles’ beacons. Another way to interpret these results is by combining beacons’ drop and collision rates into a loss rate metric calculated as $\frac{(n - \text{Throughput})}{n}$. Our enhanced G-BEE suffers 25% loss rate for a $W_u = 45$ in comparison with 61% loss rate in the individual case, i.e., the enhanced G-BEE strategy achieves as much as 92% improvement over the non-grouping scenario.

Furthermore, in Fig. 5.10, we plot our throughput curves against the number of groups/leaders when a fixed number of vehicles $n = 64$ exist in the network. Again, the G-BEE throughput $W_v$, of the simple and enhanced grouping schemes, exceeds the updated throughput $W_u$ and the throughput value of the individual case. However, the $W_v$ throughput gradually declines to approximate the values of $W_u$ when

\textsuperscript{16}The lowest achievable throughput metric defined in Section 5.3 resembles the G-BEE throughput $W_v$ in both G-BEE grouping versions.
the number of leaders approaches 40, i.e., smaller group sizes. In enhanced G-BEE, 
$W_v$'s gain drops from 39% when $N_L \simeq 6$ to 17% when the number of groups is 40.
In comparison, the simple grouping maintains less effective results than the enhanced G-BEE such that $W_v$ drops from 26.8% to 9% when $N_L \simeq 6$ and $N_L = 40$, respectively. The updated throughput $W_u$ metric shows a different tendency. In the simple G-BEE, $W_u$ is worse than the throughput of the individual case whenever $N_L \leq 20$, i.e., for $CS \geq 3$. Yet, it gives a gain of $\approx 4.8\%$ when $CS \approx 2$. The enhanced G-BEE apparently performs better such that it can encompass larger clusters (gives poor throughput in comparison with the individual context when $CS \approx \geq 12$). Furthermore, the enhanced version achieves a $W_u$ gain of 12% for the same cluster size $CS \approx \geq 2$.

![Figure 5.9. Throughput Metric Comparison for Individual Case, Simple G-BEE, and Enhanced G-BEE w.r.t. Number of Vehicles](image)

Throughout our analysis, we fixed the MAC and physical layers parameters such as data transmission rate, beacon generation rate, contention window size, etc. to the values given in Table 5.5. However, following the recent results of [60], we were encouraged to examine the drop rate $\bar{\gamma}$ w.r.t. several data transmission rates $r$ and
several beacon generation periods $T$ in Fig. 5.11 and 5.12, respectively. In comparison with the fixed 6 Mbps data rate adopted in the existing standards, $\bar{\gamma}$ rate drops without and with clustering as $r$ becomes higher such that the least drop is achieved at 12 Mbps transmission rate. Besides, as expected, the simple and enhanced clustering act superior to the no-grouping situation. For example, the enhanced G-BEE offers 51% decrement in the beacons’ drop rate when doubling the standard 6 Mbps transmission rate $r$; also, the simple G-BEE achieves 46.6% improvement in comparison to only 32% for the network when no grouping is considered. Likewise, when changing the beacon generation period $T$ from the adopted 100 msec to 150 msec, the drop rate is also decreased; without grouping, $\bar{\gamma}$ rate drops from 0.21 to 0.025. The simple grouping version’s beacon drop in vehicle queues is also cutback from 0.17 to 0.018; the enhanced formula achieves the best performance as expected and beacon drop almost diminishes at $T = 150$ msec. This desired trait comes with one drawback; lowering the frequency of beaconing leads to having more outdated vehicles’ status information broadcasted to the neighbors and to the infrastructure. This situation weakens the anticipated advantage of VANETs where real-time status of vehicles is
periodically broadcasted to avoid accidents and traffic congestion. One additional feature that grouping can offer is to beacon the leaders messages at a regular rate while only lowering the members beaconing frequency; in this case, the infrastructure does not miss much information while achieving less congestion at CCH.

Figure 5.11. Drop Rate $\bar{\gamma}$ Metric Comparison for Individual Case, Simple G-BEE, and Enhanced G-BEE w.r.t. Data Transmission Rate $r$ for $n = 64$ and $N_L = 6$ Leaders

Figure 5.12. Drop Rate $\bar{\gamma}$ Metric Comparison for Individual Case, Simple G-BEE, and Enhanced G-BEE w.r.t. Beacon Generation Period $T$ for $n = 64$ and $N_L = 6$ Leaders
Chapter 6: Pairingless Modified Efficient CPPA Scheme (PME-CPPAS) in Vehicular Networks

With the goal of utilizing a less bandwidth-demanding anonymous authentication framework, we develop, in this joint work with Faisal Alanazi\textsuperscript{17}, an IBC authentication scheme. Unlike the two available pairingless schemes \cite{2,3}, our pairingless contribution is TPD-free. Because no TPDs are used, our PME-CPPAS avoids the possibility of undergoing physical laser screening and side-channels attacks; thus, no attacker can pretend to be a registered vehicle nor can transmit a message he wants. As with the two available pairingless schemes of He et al. \cite{3} and Lo and Tsai \cite{2}, our scheme uses the IBS Schnorr signature \cite{77} as its basis\textsuperscript{18}. In Schnorr signature, the central PKG chooses an elliptic curve $E$ with large parameters ($a$, $b$, $q$, $p$, $G$, and $P$) and a hash function $h_1$. PKG randomly chooses a master private scalar key $sk_{PKG} \in \mathbb{Z}_q^*$ and calculates its master public point key $pk_{PKG} = sk_{PKG}P \in G$. PKG publishes these domain parameters to every registered user. Hence, after deployment, two phases of the scheme are conducted:

\textsuperscript{17}Faisal Alanazi is currently a PhD candidate at the Electrical and Computer Engineering Department at the Ohio State University.

\textsuperscript{18}In 1989, Schnorr presented an efficient short signature whose ECC version is known to be ECDLP secure in the random oracle model (see Section A.2.1).
• Signing at any registered sender $v$: to sign a message $m_v^i$, the sender (who is given the PKG’s $sk_{PKG}$ and $pk_{PKG}$ keys) picks a random private integer key $sk_v^i \in \mathbb{Z}_q^*$ and creates a random public point key $pk_v^i = sk_v^i.P \in G$. Hence, $v$ calculates two parts $(e_v^i, \sigma_v^i)$ of Schnorr signature such that $e_v^i = h_1(pk_v^i|m_v^i)$ or $e_v^i = h_1(m_v^i|pk_v^i)$ and $\sigma_v^i = sk_v^i + sk_{PKG}.e_v^i \mod q$, i.e., $sk_v^i = \sigma_v^i - sk_{PKG}.e_v^i \mod q$.

• Verifying at any registered receiver: receiving message $m_v^i$ and signature $(e_v^i, \sigma_v^i)$, the receiver (who has $p, q, P, pk_{PKG}$, and $h_1$) checks whether $h_1(m_v^i|pk_v^i) = h_1(m_v^i|\sigma_v^i.P - e_v^i.pk_{PKG})$, i.e., whether $\sigma_v^i.P$ equals $(pk_v^i + e_v^i.pk_{PKG})$. If it is valid, the receiver accepts $m_v^i$.

6.1 Two Available Pairingless IBC Authentication Schemes: A Motivating Comparison

Making use of a modified identity-based version of the simple Schnorr signature, both [2,3] build secure CPPA schemes. Both proposals have four main implemented phases: Generate/Setup, Extract, Sign, and Verify. The Generate phase is exactly similar to the generate phase of Schnorr where the domain parameters are generated. However, Lo and Tsai choose not to disclose the prime order $p$ of the underlying finite field $F_p$ while He et al. follow Schnorr and give it to vehicles. Also, [2] chooses to distinguish between the two trusted authorities TA and PKG such that TA is responsible for generating alias identities while PKG generates the alias secret keys.

On the other hand, rather than using one hash function, both schemes add two more one-way hashes to utilize them in the succeeding Extract phase. At the time of user registration with the PKG, every vehicle $v$ will show its real identity $v$ to obtain a TPD with stored security elements. At this phase, the two schemes show
an explicit difference; [2] prefers to keep the master private key secret; therefore, the \textit{Extract} function will be implemented offline to create specific alias identities for each registered real identity. On contrary, [3] chooses to store the master TA private key $sk_{TA}$ at the installed TPD with a user’s designated password $PWD_v$; thus, the \textit{Extract} phase is conducted online at the TPD itself. After deployment and when the vehicles are active, the \textit{Sign} and \textit{Verify} phases are carried out. To sign any outgoing message $m_v^i$, both schemes generate vehicle-specific private scalar key $sk_v^i$ and public point key $pk_v^i$ to use with the given alias identities inside the modified Schnorr signature. The sender broadcasts message $m_v^i$, vehicle alias identity $SID_v^i$, and its public key $pk_v^i$ as a part of the signature $\sigma_v^i$. Here, both He et al. and Lo and Tsai include a timestamp $T_v^i$ to withstand impersonation/replay attack. Once the broadcasted beacons are received, the receiver verifies the correctness of the received signature on $m_v^i$ and accepts $m_v^i$ only if the signature is valid. Although Schnorr signature offers a strong level of correctness with relatively fast verification, both [2, 3] choose to further expedite Schnorr’s verification; making use of Schnorr’s multi-signature property, they conduct a batch verification [70–72] by aggregating several signatures into a single validation test. Table 6.1 and 6.2 outlines the schematic description of the two schemes. By inspecting both pairingless schemes [2] and [3], it is easy to spot their differences (plus and minuses) that incentivize us to present our solution such that:

1. When any vehicle $v$ registers with the central authority by showing its real identity $v \in \{0, 1\}^*$, He et al. suggest to give the vehicle a certain password $PWD_v \in \{0, 1\}^*$. To activate signature generation phase, $v$ has to input its real given identity $ID_v$.
Table 6.1
Lo and Tsai’s Scheme [2] (V2I Setting)

Before Deployment: Setup at PKG/TA

\[ a, b, p, q, P, h_1, h_2, h_3 \]
\[ sk_{PKG}, sk_{TA} \in \mathbb{Z}_q^* \]
\[ pk_{PKG} = sk_{PKG}.P, pk_{TA} = sk_{TA}.P \in G \]
Publish \((q, P, pk_{PKG}, pk_{TA}, h_1, h_2, h_3)\)

Before Deployment: Registration and Extraction at PKG/TA

<table>
<thead>
<tr>
<th>Vehicle</th>
<th>PKG/TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v )</td>
<td>register</td>
</tr>
<tr>
<td>( ID_v )</td>
<td>( \longleftarrow )</td>
</tr>
<tr>
<td>( \hat{d}_i \in \mathbb{Z}_q^* )</td>
<td>( \longleftarrow )</td>
</tr>
<tr>
<td>( SID_{v}^{i,2} )</td>
<td>( \longleftarrow )</td>
</tr>
<tr>
<td>( SID_{v} = (SID_{v}^{i,1}</td>
<td>SID_{v}^{i,2}) )</td>
</tr>
<tr>
<td>( sid_{v}^{i,2} )</td>
<td>( \longleftarrow )</td>
</tr>
</tbody>
</table>

After Deployment

\[
Sign m_v^i \\
\quad sk_v^i \in \mathbb{Z}_q^* \\
pk_v^i = sk_v^i.P \quad \longrightarrow \quad pk_v^i \\
m_v^i \in \{0,1\}^* \quad \longrightarrow \quad m_v^i \\
T_v^i \quad \longrightarrow \quad T_v^i \\
SID_v^i \quad \longrightarrow \quad SID_v^i \\
sid_v^{i,1} \quad \longrightarrow \quad sid_v^{i,1} \\
sid_v^{i,2} \quad \longrightarrow \quad sid_v^{i,2} \\
\sigma_v^i = sid_v^{i,2} + h_3(sid_v^{i,1}|pk_v^i|SID_v^i|m_v^i|T_v^i).sk_v^i \mod q \quad \longrightarrow \quad \sigma_v^i \\
\sigma_v^i.P = ? sid_v^{i,1} + h_3(sid_v^{i,1}|pk_v^i|SID_v^i|m_v^i|T_v^i).pk_v^i + h_2(SID_v^i|sid_v^{i,1}).pk_{PKG} \]

104
### Table 6.2

**He et al.'s Scheme [3]**

<table>
<thead>
<tr>
<th>Before Deployment: Setup at TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a, b, p, q, P, h_1, h_2, h_3$</td>
</tr>
<tr>
<td>$sk_{TA} \in \mathbb{Z}_q^*$</td>
</tr>
<tr>
<td>$pk_{TA} = sk_{TA} \cdot P$</td>
</tr>
<tr>
<td>Publish $(p, q, pk_{TA}, h_1, h_2, h_3)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Before Deployment: Registration at TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Vehicle$</td>
</tr>
<tr>
<td>$v$</td>
</tr>
<tr>
<td>$\hat{b}^i \in \mathbb{Z}_q^*$</td>
</tr>
<tr>
<td>$ID_v = \hat{b}^i \cdot P$</td>
</tr>
<tr>
<td>$(ID_v, sk_{TA}, PW_{D_v}) \xleftarrow{} (ID_v, sk_{TA}, PW_{D_v})$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>After Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extraction at TPD</strong></td>
</tr>
<tr>
<td>Checking $ID_v$ and $PW_{D_v}$</td>
</tr>
<tr>
<td>$\hat{k}^i \in \mathbb{Z}_q^*$</td>
</tr>
<tr>
<td>$SID^{i,1}_v = \hat{k}^i \cdot P$</td>
</tr>
<tr>
<td>$SID^{i,2}<em>v = ID_v \oplus h_1(\hat{k}^i \cdot pk</em>{TA})$</td>
</tr>
<tr>
<td>$SID^i_v = (SID^{i,1}_v</td>
</tr>
<tr>
<td>$sid^i_v = \hat{k}^i + h_2(SID^i_v</td>
</tr>
<tr>
<td>$Sign m^i_v$</td>
</tr>
<tr>
<td>$sk^i_v \in \mathbb{Z}_q^*$</td>
</tr>
<tr>
<td>$pk^i_v = sk^i_v \cdot P$</td>
</tr>
<tr>
<td>$m^i_v \in {0,1}^*$</td>
</tr>
<tr>
<td>$T^i_v$</td>
</tr>
<tr>
<td>$SID^i_v$</td>
</tr>
<tr>
<td>$\sigma^i_v = sid^i_v +$</td>
</tr>
<tr>
<td>$h^3(SID^i_v</td>
</tr>
<tr>
<td>$\sigma^i_v \cdot P = SID^{i,1}_v + h^3(SID^i_v</td>
</tr>
</tbody>
</table>
and the correct password to be able to create pseudo identities and sign outgoing messages. Lo and Tsai as well as our approach do not use password checking step.

2. In terms of revealing more than needed, He et al. give each registered vehicle the master private key of the TA counting on the availability of the “secure” TPD. Also, they publish both prime orders $p$ and $q$ of the used elliptic curve. In comparison, Lo and Tsai keep the $sk_{PKG}$ and $sk_{TA}$ secret and do not publish the prime order $p$ of the underlying finite field. PME-CPPAS keeps the central authorities private keys hidden at authorities themselves. This is more preferable over He et al. [3] such that if any attacker captures a single TPD, he will not be able to compromise the whole system, i.e., the central authorities private keys. Furthermore, we keep the prime order, of the $P$ generator of the group of points on the used elliptic curve, secret to add extra measures of avoiding linkability issues as in Section 6.3.3.

3. From the anonymity point of view, He et al. provide better privacy since their pseudo identities and keys are generated dynamically inside the TPD after deployment, which ensures the unlinkability requirement. On the other hand, since [2] sets the fixed anonymous identities and keys at the extraction phase before deployment, there might be a chance of linkability at receivers, i.e., two sent messages by same sender could be ultimately linked to him. In comparison, PME-CPPAS has higher privacy and anonymity-level since rather than using only the fixed pseudo identities and keys of [2] or only the dynamic pseudo identities and keys of [3], our scheme uses two tiers of alias identities and keys; two first-tier fixed sets of identities and keys installed on the vehicle before deployment at time of registration, and two second-tier dynamic identities and keys used when broadcasting after deployment.
To complicate any attempt of revealing the real identities by any attacker, we use more one-way hash functions in the generation phase of these alias identities; Lo and Tsai use only two hash functions offline while He et al.’s scheme have to add more hashes since the extraction phase is conducted online at the TPD.

4. [2]’s scheme is designed for a two-hop setting: V2I and I2V; any vehicle’s broadcasted beacons via V2I are verified at the infrastructure and then conveyed via the I2V links to other vehicles. In contrary, He et al.’s scheme suits both one-hop V2V and V2I communications. Both schemes utilize batch verification of received signatures to expedite verification process. Similarly, PME-CPPAS is designed to verify single as well as multiple identity-based signatures whether the receiver is another vehicle or an infrastructure via one-hop links.

5. Unlike [2, 3], PME-CPPAS can be used for confidentiality purposes. We set up a shared-key between any two entities in the network. The resulted session key can be used in encrypting/decrypting any further sensitive messages.

Our scheme sums up the good traits of the two strategies and overcomes their shortcomings. The following section gives a detailed description of PME-CPPAS before proving its security soundness in Section 6.3 and analyzing its effectiveness in Section 6.4.

6.2 The Proposed PME-CPPAS Framework

Our network (Fig. 6.1) consists of a central CA/TA/PKG that sets all of the needed security parameters in the same way [2, 3, 77] do; given a security parameter $L$, central authority sets up the ECC curve domain parameters. The TA/PKG randomly
chooses a master private scalar key $sk_{TA/PKG} \in \mathbb{Z}_q^*$ and calculates its master public point key $pk_{TA/PKG} = sk_{TA/PKG}.P$. Next, authority selects three secure one-way collision-resistant hash functions $h_1$, $h_2$, and $h_3$ where:

$$h_1 : G \times G \rightarrow \mathbb{Z}_q^*,$$  \hspace{1cm} (6.1)  

$$h_2 : G \times \{0,1\}^* \rightarrow \mathbb{Z}_q^*,$$ \hspace{1cm} and 

$$h_3 : \{0,1\}^* \times G \times \{0,1\}^* \times \{0,1\}^* \rightarrow \mathbb{Z}_q^*. \hspace{1cm} (6.3)$$

After this Generate phase, when any vehicles manufacturer approach the CA/TA/PKG to register its vehicles, the authority will first publish to each registered OBU the generated security parameters. However, TA/PKG will neither publish its master secret key $sk_{TA/PKG}$ nor the prime order $q$ of $P$. The reason behind not publishing $q$ is to achieve the unlinkability security requirement as we explain in Section 6.3.3. Not publishing $q$ does not affect any computations at OBUs due to $G = E$’s cyclic property.

To obtain security parameters, each vehicle $v$ shows its real chassis number $v$ to the central authority. From $v$, TA/PKG generates a long term real point identity $ID_v \in G$ (6.4) and Extract two sets of first-tier fixed pseudo identity and key: point $SID_v^i$ (6.5) and scalar $sid_v^i$ (6.6) for that vehicle where

$$ID_v = \hat{b}.P \in G,$$ \hspace{1cm} (6.4)  

$$SID_v^i = \hat{k}.P \in G, \text{ and}$$ \hspace{1cm} (6.5)
\[
\text{sid}_v^i = \hat{k}^i + h_1(\text{ID}_v|\text{SID}_v^i).sk_{TA/PKG} mod q \in \mathbb{Z}_q^*.
\]

(6.6)

To keep record and trace every registree, the TA/PKG calculates:

\[
\text{ID}_v^{i,track} = (h_1(\text{ID}_v|\text{SID}_v^i))^{-1}.\text{SID}_v^i mod q \in G,
\]

(6.7)

where \((h_1(\text{ID}_v|\text{SID}_v^i))^{-1}\) is a co-prime inverse (computed using Extended Euclidean Algorithm) of \(h_1(\text{ID}_v|\text{SID}_v^i)\) modulus the order of \(P\); \(\text{ID}_v^{i,track}\) represents a means to trace misbehaving OBUs (in an agreement with some legal authority) and to locate the real identity of the liable sender as Section 6.3.3 explains. Once calculated, the TA/PKG stores the tuple \(< v, \text{ID}_v, \hat{b}^i, \{\hat{k}^i, \text{SID}_v^i, \text{sid}_v^i, \text{ID}_v^{i,track}\}>\) and offline submits \(< \text{ID}_v, \{\text{SID}_v^i, \text{sid}_v^i\}>\) to the related registree through a secure channel.\(^{19}\)

In our model, \(v\) does not disclose its \(\text{ID}_v, \text{SID}_v^i, \) and \(\text{sid}_v^i\) so as they are not leaked. Rather, \(v\) broadcasts second-tier dynamically derived versions of these first-tier fixed anonymous identity and key. At the end of the offline phase, the system is ready to be deployed on roads where vehicles can communicate and exchange the safety

\(^{19}\)\(v\) can easily check whether the created first-tier alias identity and key were generated by that legitimate TA/PKG by verifying whether each \(\text{sid}_v^i.P = \text{SID}_v^i + h_1.pk_{TA/PKG}\) is correct.
and non-safety messages via the V2V and V2I/I2V links. Here comes the role of our PME-CPPAS scheme to fulfill two of the main VANETs’ security requirements:

6.2.1 Authentication for Safety Messages: (Sign and Verify)

When any OBU\textsubscript{v} wants to broadcast a status beacon \(m\textsubscript{i,v}\), it first dynamically generates two vehicle-specific keys: private scalar key \(sk\textsubscript{i,v} \in \mathbb{Z}_q^*\) and public point key \(pk\textsubscript{i,v} = sk\textsubscript{i,v}.P\). Although the value of the generated \(sk\textsubscript{i,v}\) might lie beyond \(\mathbb{Z}_q^*\) because the order \(q\) of the elliptic curve is hidden and not given to vehicles, the produced public key will be a point on the same elliptic curve \(E\) due to the \(E\’s\) cyclic property.

OBU\textsubscript{v} will dynamically compute new second-tier anonymous identity \(hSID\textsubscript{i,v}\) and key \(H\textsubscript{i,v}\) (from the previously given alias identities and keys and from the created \(pk\textsubscript{i,v}\)) as:

\[
H\textsubscript{i,v} = h_1(ID\textsubscript{v}|SID\textsubscript{i,v}).h_2(pk\textsubscript{i,v}|T\textsubscript{i,v}) \in \mathbb{Z}_q^*, \tag{6.8}
\]

\[
hSID\textsubscript{i,v} = h_2(pk\textsubscript{i,v}|T\textsubscript{i,v}).SID\textsubscript{i,v} \in G. \tag{6.9}
\]

To \textit{Sign} message \(m\textsubscript{i,v}\), OBU\textsubscript{v} computes the \(\sigma\textsubscript{i,v} \in \mathbb{Z}_q^*\)’s part of Schnorr signature after attaching the current timestamp \(T\textsubscript{i,v}\) to avoid any replay attack:

\[
\sigma\textsubscript{i,v} = h_2(pk\textsubscript{i,v}|T\textsubscript{i,v}).sid\textsubscript{i,v} + h_3(H\textsubscript{i,v}|hSID\textsubscript{i,v}|pk\textsubscript{i,v}|m\textsubscript{i,v}|T\textsubscript{i,v}).sk\textsubscript{i,v} \tag{6.10}
\]

before broadcasting \(< m\textsubscript{i,v}, T\textsubscript{i,v}, H\textsubscript{i,v}, hSID\textsubscript{i,v}, \sigma\textsubscript{i,v}, pk\textsubscript{i,v} >\). Any legitimate receiver, who already has the TA/PKG’s domain parameters, first checks the freshness of \(T\textsubscript{i,v}\) before it can conduct the \textit{Verification} phase in one of two ways:

1. Single Verification: a receiver will accept \(m\textsubscript{i,v}\) if

\[
\sigma\textsubscript{i,v}.P = hSID\textsubscript{i,v} + h_3(H\textsubscript{i,v}|hSID\textsubscript{i,v}|pk\textsubscript{i,v}|m\textsubscript{i,v}|T\textsubscript{i,v}).pk\textsubscript{i,v} + H\textsubscript{i,v}.pk\textsubscript{TA/PKG} \tag{6.11}
\]
holds. Otherwise, the receiver rejects the message. In this type of verification, the receiver verifies each received signature individually, i.e., \( n \) verification are needed for \( n \) signatures from \( n \) members.

2. Batch Verification: receivers can expedite the verification of multiple signatures from multiple OBUs when they verify batch of signatures at once. For example, \( n \) tuples \( <m^k_1, T^k_1, H^k_1, hSID^k_1, \sigma^k_1, pk^k_1>, <m^k_2, T^k_2, H^k_2, hSID^k_2, \sigma^k_2, pk^k_2>, \ldots, <m^k_n, T^k_n, H^k_n, hSID^k_n, \sigma^k_n, pk^k_n> \) are received from \( n \) vehicles at time instance \( k \). In batch verification, to ensure that no modification attack \[75\] has been exerted on the received messages, the receiver runs a well-known cryptographic small exponents test \[89\]. In which, the receiver chooses a vector \( \alpha = \{\alpha^1, \alpha^2, \ldots, \alpha^n\} \) where each scalar \( \alpha^i \in \{0, 1\} \); \( t \) is the security parameter of the test and it is a small integer with small computation overhead \[2, 3, 75, 89\]. Also, \( \alpha^i \neq 0 \) is a requisite to exclude the un-verification of some of the received \( n \) signatures. The values of \( \alpha \) are included in the following verification condition

\[
\sum_{i=1}^{n} \alpha^i.v^k_i.P = \sum_{i=1}^{n} \alpha^i.hSID^k_i + \sum_{i=1}^{n} \alpha^i.h_3(H^k_i|hSID^k_i|pk^k_i|m^k_i|T^k_i).pk^k_i + \sum_{i=1}^{n} \alpha^i.H^k_i).pk^k_{TA/PKG}
\]

(6.12)

that which if it holds, \( m^k_i, i = 1, 2, \ldots, n \) are accepted by the receiver.

### 6.2.2 Confidentiality for Non-safety Messages: (Pairwise Key Establishment)

In addition to authentication, vehicles can use PME-CPPAS to dynamically establish a pairwise identity-based shared key in a Diffie-Hellman (DH) exchange fashion. Fundamentally speaking, any two OBUs, \( v \) and \( w \), which have previously received
each other beacons and were able to authenticate one another, have each other public
keys \( pk^i_v \) and \( pk^i_w \); therefore, in future when these two vehicles want to confidentialize
a specific data, they can easily create a pairwise scalar session key \( s_{v,w} = s_{w,v} \) that no
one else knows or is able to reveal. The key represents the x-coordinate of the point
\( S_{v,w} = S_{w,v} \) computed as:

\[
(s_{v,w} = sk^i_v pk^i_w) = (sk^i_w pk^i_v = S_{w,v}) \in G.
\] (6.13)

If \( v \) wants to secretly send \( m^i_v \) to \( w \), \( v \) will encrypt \( m^i_v \) using the created session key
\( s_{v,w} \) to obtain \( C_{m^i_v} \):

\[
C_{m^i_v} = enc_{s_{v,w}}(m^i_v) = m^i_v + s_{v,w}.P \in G.
\] (6.14)

Then OBU\(_v\) can broadcast or unicast \(< C_{m^i_v}, T^i_v, H^i_v, hSID^i_v, \sigma^i_v, pk^i_v, pk^i_w \>\). All
legitimate receivers will be able to check the freshness of \( T^i_v \) and can verify the sig-
nature; however, only OBU\(_w\) can decrypt the ciphertext \( C_{m^i_v} \) since it can successfully
compute the used pairwise key from (6.13). \( w \) then decrypts \( C_{m^i_v} \) to extract \( m^i_v \) as

\[
m^i_v = dec_{s_{w,v}}(C_{m^i_v}) = C_{m^i_v} - s_{w,v}.P \in G;
\] (6.15)

\( m^i_v \) can be the x-coordinate of the resulted decrypted message in (6.15). Having
explained the scheme, next section analyzes the scheme’s security.

### 6.3 PME-CPPAS’ Security Analysis

This section defines the considered attacker model following [3,90,91]. To show the
existential unforgeability of our PME-CPPAS scheme against such adaptive chosen
message attacker, we build our security proof utilizing the forking lemma and attack
game strategies of [91–93]. Afterwards, we demonstrate the PME-CPPAS scheme
Table 6.3
OUR PME-CPPAS Approach

<table>
<thead>
<tr>
<th>Before Deployment: Setup at TA/PKG</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Before Deployment: Registration and Extraction at TA/PKG</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vehicle</strong></td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>$v$</td>
</tr>
<tr>
<td>$\hat{b}^i, \hat{k}^i \in Z_q^*$</td>
</tr>
<tr>
<td>$ID_v$</td>
</tr>
<tr>
<td>$SID_v^i$</td>
</tr>
<tr>
<td>$sid_v^i$</td>
</tr>
</tbody>
</table>

After Deployment: Authentication Phase

<table>
<thead>
<tr>
<th>Sign $m_v^i$</th>
<th>Verify $\sigma_v^i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$sk_v^i \in Z_q^*$</td>
<td>$pk_v^i$</td>
</tr>
<tr>
<td>$pk_v^i = sk_v^i \cdot P$</td>
<td>$m_v^i$</td>
</tr>
<tr>
<td>$m_v^i \in {0,1}^*$</td>
<td>$T_v^i$</td>
</tr>
<tr>
<td>$H_v^i = h_2(pk_v^i</td>
<td>T_v^i) \cdot h_1(ID_v</td>
</tr>
<tr>
<td>$hSID_v^i = h_2(pk_v^i</td>
<td>T_v^i) \cdot SID_v^i$</td>
</tr>
<tr>
<td>$\sigma_v^i = h_2(pk_v^i</td>
<td>T_v^i) \cdot sid_v^i + h_3(H_v^i</td>
</tr>
<tr>
<td>$\sigma_v^i \cdot P \leftarrow hSID_v^i + h_3(H_v^i</td>
<td>hSID_v^i</td>
</tr>
</tbody>
</table>

$$
(\sum_{i=1}^{n} \alpha_i \cdot \sigma_v^i) \cdot P \leftarrow (\sum_{i=1}^{n} \alpha_i \cdot hSID_v^i) + (\sum_{i=1}^{n} \alpha_i \cdot h_3(H_v^i | hSID_v^i | pk_v^i | m_v^i | T_v^i) \cdot pk_v^i) + (\sum_{i=1}^{n} \alpha_i \cdot H_v^i) \cdot pk_{TA/PKG} $$$
ability of withstanding various types of attacks and satisfying essential security requirements of VANETs.

6.3.1 Adaptive Chosen Message Attacker (CMA)

The main building block of our authentication scheme is an identity-based signature; for any signature scheme to be secure, it has to be existentially unforgeable under a Chosen Message Attack (CMA) [90]. Therefore, in our model we consider an attacker $\mathcal{A}$, a probabilistic Turing machine, who mounts an adaptive chosen message attack within a bounded polynomial time $\tau$. The basic game behind his Existential Forgeability (EF) attack is shown in Fig. 6.2. Given the public parameters (generated using $\text{Setup}(1^L)$ oracle) of any signing vehicle and given access to three hash oracles $h_1$, $h_2$, and $h_3$, $\mathcal{A}$ can ask the signer/challenger $\mathcal{S}$ to sign several messages $m_i^{\mathcal{A}}$ of his choice by mounting a series of signing queries. He does this with the opportunity at each step $i$ to analyze the previous message-signature pairs $(m_{i-1}^{\mathcal{A}}, \sigma_{i-1}^{\mathcal{A}})$ and adapts his next query $m_i^{\mathcal{A}}$ accordingly. However, given that ability, $\mathcal{A}$ should not be able to produce a valid signature $\sigma_k^{\mathcal{A}}$ on a new message $m_k^{\mathcal{A}}$ that was not queried before, i.e., $k \notin \{i\}$. The probability of $\mathcal{A}$ being successful in producing a valid $\sigma_k^{\mathcal{A}}$ for the queried $m_k^{\mathcal{A}}$ is referred to as $Pr_{\text{EF-CMA}}(\mathcal{A})$ [91,93].

6.3.2 Security Claim and Proof

In the face of this severe attack, we claim that the proposed scheme is secure. The attacker cannot existentially forge a signature $\sigma_k^{\mathcal{A}}$ for message $m_{k\notin\{i\}}^{\mathcal{A}}$ in a given polynomial time $\tau$, i.e., $\mathcal{A}$’s $Pr_{\text{EF-CMA}}(\mathcal{A})$ is negligible. To attest to the validity of our signature-unforgeability claim, we are relying on the classical notion of security-by-reduction proofs; reduction makes it easier to prove the security of a cryptosystem.
relying on the hardness of its underlying computational problem. In this sense, we can reduce the problem of forging a signature in our scheme to solving its underlying ECDLP problem (Appendix A.2) in the random oracle model [92,94]. In other words, the PME-CPPAS signature is secure if solving its ECDLP problem is infeasible.

**Proof.** To prove our claim, as with any reduction technique, we start by constructing a Reduction Simulator $\mathcal{RS}$ who utilizes $\mathcal{AT}$’s forging ability to solve the ECDLP problem. The conspiracy of $\mathcal{RS}$ and $\mathcal{AT}$ begins when $\mathcal{RS}$ is given the PME-CPPAS’s system parameters $(p, P, \hat{P}, h_1, h_2, h_3)$. $\mathcal{RS}$’s goal is to find $\hat{x} \in \mathbb{Z}_q^*$ that produced $\hat{P} \in G$; $\hat{x}$ represents the solution of the ECDLP since $\hat{P} = \hat{x}.P$. Since $\mathcal{RS}$ runs $\mathcal{AT}$ on these parameters, $\mathcal{AT}$ starts $q_{h_1}$, $q_{h_2}$, $q_{h_3}$, and $q_s$ queries to the hash and sign oracles that are maintained by $\mathcal{RS}$, in order to produce a forgery with a $Pr_{\text{EF-CMA}}^{\text{PME-CPPAS}}(\mathcal{AT})$ probability in time $\tau$. $\mathcal{RS}$ simulates the hashing and signing oracles for $\mathcal{AT}$; first,
RS keeps a table/list in the form of \((m^\text{index}, h_1^\text{index}(), h_2^\text{index}(), h_3^\text{index}(), \text{sign}^\text{index}())\).

Whenever RS receives a hash or a sign query, RS checks the list; if the queried tuples are in the list, RS returns the stored assigned values, else RS dynamically populates the table with the new entries and responds with random values as follows:

- **Simulation of \(h_1\)-Oracle** as \(L_{h_1} = \langle ID_*^\text{index}, SID_*^\text{index}, h_1^\text{index}() \rangle\): when receiving \(\mathcal{A}T\)'s \(i^{th}\) query \((ID_{\mathcal{A}T}^i, SID_{\mathcal{A}T}^i)\), RS checks whether it exists in \(L_{h_1}\) first. If it is set, directly by a previous hash query, or by a signature query, RS sends the corresponding \(h_1^i = h_1(ID_{\mathcal{A}T}^i, SID_{\mathcal{A}T}^i)\) to \(\mathcal{A}T\); otherwise, RS generates a random \(h_1^i \in \mathbb{Z}^*_q\), adds \(<ID_{\mathcal{A}T}^i, SID_{\mathcal{A}T}^i, h_1^i>\) to \(L_{h_1}\), and sends the generated \(h_1^i\) to \(\mathcal{A}T\).

- **Simulation of \(h_2\)-Oracle** as \(L_{h_2} = \langle pk_*^\text{index}, T_*^\text{index}, h_2^\text{index}() \rangle\): upon receiving \(\mathcal{A}T\)'s \(i^{th}\) query \((pk_{\mathcal{A}T}^i, T_{\mathcal{A}T}^i)\), RS checks whether it exists in \(L_{h_2}\) first. If so, RS sends the corresponding \(h_2^i = h_2(pk_{\mathcal{A}T}^i, T_{\mathcal{A}T}^i)\) to \(\mathcal{A}T\); otherwise, RS generates a random \(h_2^i \in \mathbb{Z}^*_q\), adds \(<pk_{\mathcal{A}T}^i, T_{\mathcal{A}T}^i, h_2^i>\) to \(L_{h_2}\), and sends \(h_2^i\) to \(\mathcal{A}T\).

- **Simulation of \(h_3\)-Oracle** as \(L_{h_3} = \langle pk_*^\text{index}, hSID_*^\text{index}, H_*^\text{index}, m_*^\text{index}, T_*^\text{index}, h_3^\text{index}() \rangle\): upon receiving \(\mathcal{A}T\)'s \(i^{th}\) query \((pk_{\mathcal{A}T}^i, hSID_{\mathcal{A}T}^i, H_{\mathcal{A}T}^i, m_{\mathcal{A}T}^i, T_{\mathcal{A}T}^i)\), RS checks whether it exists in \(L_{h_3}\) first. If yes, RS sends the corresponding \(h_3^i = h_3(pk_{\mathcal{A}T}^i, hSID_{\mathcal{A}T}^i, H_{\mathcal{A}T}^i, m_{\mathcal{A}T}^i, T_{\mathcal{A}T}^i)\) to \(\mathcal{A}T\); otherwise, RS generates a random \(h_3^i \in \mathbb{Z}^*_q\), adds \(<pk_{\mathcal{A}T}^i, hSID_{\mathcal{A}T}^i, H_{\mathcal{A}T}^i, m_{\mathcal{A}T}^i, T_{\mathcal{A}T}^i, h_3^i>\) to \(L_{h_3}\), and sends \(h_3^i\) to \(\mathcal{A}T\).

- **Simulation of Sign-Oracle**: when \(\mathcal{A}T\) asks for a signature over message \(<m_{\mathcal{A}T}^j, T_{\mathcal{A}T}^j>\), RS first checks its stored table \((m_{\mathcal{A}T}^j, h_1^\text{index}(), h_2^\text{index}(), h_3^\text{index}(), \text{sign}^\text{index}())\) to see whether \(m_{\mathcal{A}T}^j\) has been previously queried for. If yes, RS returns the stored values of \(<pk_{RS}^j, hSID_{RS}^j, v_{RS}^j, H_{RS}^j, m_{\mathcal{A}T}^j, T_{\mathcal{A}T}^j>\) to \(\mathcal{A}T\); otherwise, RS
uniformly generates these values. First, \( \mathcal{RS} \) generates the signature \( \sigma_{\mathcal{RS}}^j \in \mathbb{Z}_q^* \). \( \mathcal{RS} \) chooses two random points \( ID_{\mathcal{RS}} \) and \( SID_{\mathcal{RS}}^j \), calculates \( h_1(ID_{\mathcal{RS}}|SID_{\mathcal{RS}}^j) \), and populates its \( L_{h_1} \) list with \( < ID_{\mathcal{RS}}, SID_{\mathcal{RS}}^j, h_1 > \). Similarly, \( \mathcal{RS} \) randomly chooses \( h_2 \in \mathbb{Z}_q^* \) and calculates \( H_{\mathcal{RS}}^j = h_2 \cdot h_1 \in \mathbb{Z}_q^* \) and \( hSID_{\mathcal{RS}}^j = h_2 \cdot SID_{\mathcal{RS}}^j \in G \) accordingly. Having these parameters, besides a random \( h_3 \), \( \mathcal{RS} \) calculates \( pk_{\mathcal{RS}}^j = h_3^{-1} \cdot (\sigma_{\mathcal{RS}}^j \cdot P - hSID_{\mathcal{RS}}^j - H_{\mathcal{RS}}^j \cdot \hat{P}) \). Both \( L_{h_2} \) and \( L_{h_3} \) can be updated with \( < pk_{\mathcal{RS}}^j, T_{\mathcal{AT}}^j, h_2 > \) and \( < pk_{\mathcal{RS}}^j, hSID_{\mathcal{RS}}^j, H_{\mathcal{RS}}^j, m_{\mathcal{AT}}^j, T_{\mathcal{AT}}^j, h_3 > \), respectively. Lastly, \( \mathcal{RS} \) sends the signature elements \( < pk_{\mathcal{RS}}^j, hSID_{\mathcal{RS}}^j, H_{\mathcal{RS}}^j, m_{\mathcal{AT}}^j, T_{\mathcal{AT}}^j, \sigma_{\mathcal{RS}}^j > \) to \( \mathcal{AT} \) in response to his query on \( < m_{\mathcal{AT}}^j, T_{\mathcal{AT}}^j > \). Once the signature is received by \( \mathcal{AT} \), he can easily verify that the signature is valid (6.16) in spite of its mere random-based construction by \( \mathcal{RS} \):

\[
\sigma_{\mathcal{RS}}^j \cdot P = hSID_{\mathcal{RS}}^j + h_3 \cdot pk_{\mathcal{RS}}^j + H_{\mathcal{RS}}^j \cdot \hat{P},
\]  

(6.16)

where \( h_3 \) is obtained by \( \mathcal{AT} \) because of his ability to query \( h_3 \)-oracle for the received \((H_{\mathcal{RS}}^j, hSID_{\mathcal{RS}}^j, pk_{\mathcal{RS}}^j, m_{\mathcal{AT}}^j, T_{\mathcal{AT}}^j)\) elements from \( \mathcal{RS} \).

Having obtained the answers of the hash and sign oracles, \( \mathcal{AT} \) outputs his (supposed) forgery on a random message \( m_{\mathcal{AT}}^k \) that has not been queried to the sign oracle before. Hereabouts, \( \mathcal{AT} \) can create his signature \( \sigma_{\mathcal{AT}}^k \) in many ways; in what follows, we introduce three possible ways of signature creation by \( \mathcal{AT} \) and discuss their details:

- **Method 1**: we fantasize a trick by which \( \mathcal{AT} \) can always produce a valid signature \( \sigma_{\mathcal{AT}}^k \) on \((m_{\mathcal{AT}}^k, T_{\mathcal{AT}}^k)\) when it is verified at \( \mathcal{RS} \). In this trick, \( \mathcal{AT} \) will first query \( h_1 \)-oracle on a random pair \((ID_{\mathcal{AT}}^k, SID_{\mathcal{AT}}^k)\) to obtain \( h_1^k \) value. Choosing a random \( sk_{\mathcal{AT}}^k \in \mathbb{Z}_q^* \), \( \mathcal{AT} \) calculates \( pk_{\mathcal{AT}}^k = sk_{\mathcal{AT}}^k \cdot P \in G \) and queries \( h_2 \)-oracle
on \((pk^k \at, T^k \at)\) to obtain \(h^k_3\). After calculating \(H^k \at = h^k_2 \cdot h^k_1\) and \(hSID^k \at = -H^k \at \cdot \hat{P} \in G\), \(\at\) can query \(h_3\)-oracle on \((pk^k \at, hSID^k \at, H^k \at, m^k \at, T^k \at)\) to obtain \(h^k_3\). The output signature \(\sigma^k \at\) is computed as \(h^k_3 \cdot s^k \at \in Z_q^*\). Once the tuple \((pk^k \at, hSID^k \at, H^k \at, m^k \at, T^k \at, \sigma^k \at)\) is received by \(\rs\) as a signature on \((m^k \at, T^k \at)\), the verification \(\sigma^k \at \cdot P = hSID^k \at + h^k_3 \cdot pk^k \at + H^k \at \cdot \hat{P}\) is valid, because of the tricks that \(\rs\) just played.

- **Method 2**: since \(\at\) can deduce the pattern of signing that \(\rs\) used from querying the \(\text{Sign}\)-oracle, \(\at\) will try to follow the same steps to create his signature. First, he randomly generates the signature \(\sigma^k \at \in Z_q^*\) and two points \(ID \at\) and \(SID^k \at\). Querying the \(h_1\)-oracle, \(\at\) obtains \(h^k_1\) from \(\rs\). In the same way, \(\at\) randomly chooses \(h^k_2\) and \(h^k_3\) \(\in Z_q^*\). Using them, he calculates \(H^k \at = h^k_2 \cdot h^k_1 \in Z_q^*\) and \(hSID^k \at = h^k_2 \cdot SID^k \at \in G\). Now, \(pk^k \at\) can be calculated as \(h^k_3^{-1} \cdot (\sigma^k \at \cdot P - hSID^k \at \cdot H^k \at \cdot \hat{P})\). The issue here is how to update both \(L_{h_2}\) and \(L_{h_3}\) at \(\rs\) with the values of \(< pk^k \at, T^k \at, h^k_2 >\) and \(< pk^k \at, hSID^k \at, H^k \at, m^k \at, T^k \at, h^k_3 >\), respectively. If such update is easy, \(\rs\) will be surely able to validate the signature tuple \(< pk^k \at, hSID^k \at, H^k \at, m^k \at, T^k \at, \sigma^k \at >\), which \(\at\) sends, as (6.16) shows.

- **Method 3**: this method is simply the reverse order of method 2. Rather than calculating \(pk^k \at \in G\) at the final steps, here it is randomly chosen at the beginning before the \(h_2\)-oracle is queried on \((pk^k \at, T^k \at)\). Similarly, the value of \(h^k_3\) is obtained from the \(h_3\)-oracle after calculating \(hSID^k \at\) and \(H^k \at\). Choosing \(\sigma^k \at \in Z_q^*\), the tuple \(< pk^k \at, hSID^k \at, H^k \at, m^k \at, T^k \at, \sigma^k \at >\) is sent to \(\rs\) as a signature.
on \((m^k_{AT}, T^k_{AT})\). The validity of the received signature, merely depends on the randomness of \(\sigma^k_{AT}, h^k_2, \) and \(h^k_3\) when is checked using (6.16).

In the fictitious arrangement of method 1, \(RS\) can never extract \(\hat{x}\). The reason is that the terms, of the validity test (6.16), cancel each other even when different hash values are obtained from the \(RS\) oracles. In this regard, the probability of solving ECDLP equals the probability of forgery \(Pr^{\text{EF-CMA}}_{\text{PME-CPPAS}}(AT)\). However, in methods 2 and 3, \(RS\) can extract \(\hat{x}\) making use of the forking lemma of Pointcheval and Stern [91, 92]. In order to solve the ECDLP problem by \(RS\), besides the first obtained signature tuple \(< pk^k_{AT}, hSID^k_{AT}, H^k_{AT}, m^k_{AT}, T^k_{AT}, \sigma^k_{AT} >\), another forgery is required with the same \(pk^k_{AT}\) key as part of the signature. Since \(AT\) can rewind/replay the call for \(h_1, h_2, h_3,\) and \(\text{Sign-}\)oracles a polynomial \((q_{h_1}, q_{h_2}, q_{h_3}, \text{and} \ q_s)\) number of times, the forking lemma’s basis is the following intuition. The attacker’s call for \(RS\)’s oracles can give similar results until some point \(i^* \in \{1..q_{h_1,2,3}\}\). Unlike the obtained oracle values in the first \((i^* - 1)\) attempts, from point \(i^*\) on, the results of the queries will fork/differ from the previously obtained values\(^{20}\). In this regard, different \(h^k_2\) and \(h^k_3\) values can be obtained, leading to different \(H^k_{AT}\) and \(hSID^k_{AT}\). As a result, \(AT\) outputs another valid signature tuple \(< pk^k_{AT}, hSID^k_{AT}', H^k_{AT}', m^k_{AT}, T^k_{AT}', \sigma^k_{AT}' >\) for \((m^k_{AT}, T^k_{AT}')\); Fig. 6.3 represents a schematic view of the forking lemma in our proof. In this case, the solution to the ECDLP, \(\hat{x}\), can be extracted as

\[
\hat{x} = \left[ \frac{(\sigma^k_{AT} - \sigma^k_{AT}').P - (h^k_2 - h^k_2').SID^k_{AT} - (h^k_3 - h^k_3').pk^k_{AT}}{(h^k_2 - h^k_2').h^k_{1}.P} \right]. \tag{6.17}
\]

The crucial concern at the moment regards the probability of solving the ECDLP. As we stated earlier, the reduction simulator, via this collusion with \(AT\), wants to

\(^{20}\)In our model, the value of \(h^k_1\) remains the same since the same \((ID_{AT}, SID^k_{AT})\) pair is sent to \(RS\) in all of the \(q_{h_1}\) queries.
Figure 6.3. **RS** Allows **AT** To Replay **RS**’s Oracles To Solve ECDLP Problem and Extract $\hat{x}$.
extract $\hat{x}$. Our aim is to prove that such $\mathcal{RS}$ reduction is loose, i.e., the probability of extraction is still negligible and it is much smaller than the $Pr_{\text{EF-CMA \text{PME-CPPAS}(\mathcal{AT})}}$ probability of forging a signature. Because the $\mathcal{AT}$’s queries of $h_2$- and $h_3$-oracles are forking at point $i^*$, we need to guess on which $q_{h_2}$ or $q_{h_3}$ or both ($q_{h_2}$ and $q_{h_3}$) queries the forking occurred. In this way, the probability of solving the ECDLP will be reduced by a factor of $(\frac{1}{q_{h_2}} + \frac{1}{q_{h_3}} + (\frac{1}{q_{h_2}} \cdot \frac{1}{q_{h_3}}))$ since the occurrence of these events are independent. By adding to this factor the queries that $\mathcal{AT}$ makes to the signing oracle, the final probability of breaking the hardness of ECDLP will be

$$Pr_{\text{EF-CMA \text{PME-CPPAS}(\mathcal{AT})}}(\cdot) \cdot \left( \frac{3.2^{2L} + 1}{2^{4L}} \right) \cdot Pr_{\text{EF-CMA \text{PME-CPPAS}(\mathcal{AT})}}(\cdot) \approx \frac{Pr_{\text{EF-CMA \text{PME-CPPAS}(\mathcal{AT})}}(\cdot)}{O(2^{2L})}.\quad \square$$

In comparison to [2–5], our proof is more solid and better built such that we clearly identify the attacker model, the game he plays, and the result of his attempted attack. Our PME-CPPAS’s proven-security facilitates explaining its ability to defy several types of attacks. First, it is obvious that our scheme is secure against replay attacks because of the inclusion of timestamps within the created messages and hashes. Second, our previous proof is strong evidence that our scheme is secure against the man-in-the-middle attack where attacker $\mathcal{AT}$ is unsuccessful in forging signatures even by playing those manipulation games. Third, for impersonation attacks, since $\mathcal{AT}$ cannot generate a valid signature on his own, he cannot mount impersonation attack. However, he might try the following trick to impersonate any vehicle $v$. When $\mathcal{AT}$ receives $<\mathcal{pk}_v^i, hSID_v^i, H_v^i, m_v^i, T_v^i, \sigma_v^i>$ from vehicle $v$, he embeds his
random scalar private key $sk_{AT}^i$ in the message before he broadcasts $< (sk_{AT}^i,pk_v^i)$, $(sk_{AT}^i.hSID_v^i)$, $(sk_{AT}^i.H_v^i)$, $m_v^i$, $T_v^i$, $(sk_{AT}^i,\sigma_v^i)$ > to his neighbors. Even with this trick, $AT$’s broadcasted message will be rejected since the verification (6.11) at any receiving vehicle is not valid; $h_3$ value is discrepant, so it makes the masquerading of $v$ unsuccessful. In the same way, any form of modification, which $AT$ might attempt over any vehicle $v$’s message with the aim of mounting a modification attack, is easily spotted because of the use of hashes in styling the (6.11) and (6.12) verification conditions.

### 6.3.3 Preserving VANETs’ Security and Privacy Requirements

Our proposed scheme fulfills the four main VANETs’ security requirements of trust, non-repudiation, freshness, and integrity. For the trustability aspect, when vehicle $v$ registers with the trusted authority, it obtains the alias key $\{sid_v^i\}$ set of secret scalars that are derived from the trusted authority $pk_{TA/PKG}$ key; therefore, only legitimate members, which have $pk_{TA/PKG}$, can verify the sender’s signature $\sigma_v^i$ as in (6.11). To ensure that senders will not repudiate/deny that they sent messages, signature $\sigma_v^i$ is formulated such that it is connected with the sender anonymous identities and public key $pk_v^i$. The freshness is achieved through the use of timestamps $T_v^i$ as we stated earlier and the integrity of message $m_v^i$ is ensured via the use of hash functions.

For the more advanced privacy requirements, we show how our PME-CPPAS scheme achieves:

- **Anonymity and Unlinkability:** on one hand, identity privacy is attained through the use of alias set of point $SID_v^i$ identities and scalar $sid_v^i$ keys. These first-tier fixed anonymous ingredients are given to vehicle $v$ by the trusted authority at time
of registration to conceal the given real identity $ID_v$. On the other hand, the vehicles themselves can dynamically generate second-tier anonymous identities $hSID_v^i$ and keys $H_v^i$ that are related to the dynamically generated public key $pk_v^i$ and to the first-tier alias ingredients without the help of TA/PKG. Because of changing these second-tier anonymous parameters and public key $pk_v^i$ with every broadcast, the signer’s anonymity is guaranteed. Such anonymity feature ensures that for every two received tuples $< pk_v^i, hSID_v^i, H_v^i, m_v^i, T_v^i, \sigma_v^i >$ at time $i$ and $< pk_v^j, hSID_v^j, H_v^j, m_v^j, T_v^j, \sigma_v^j >$ at time $j$, no receiver (other than TA/PKG) can link $pk_v^i$ to $pk_v^j$, $hSID_v^i$ to $hSID_v^j$, and $H_v^i$ to $H_v^j$.

b. Conditional Privacy and Traceability: although vehicles need to be untraceable by the public, TA/PKG must still be able to identify its registrees in case of dispute for accountability purposes. Since central authority stores the $ID_v$, $\hat{d}_i$, $\hat{k}_i$, $SID_v^i$, $sid_v^i$, and $ID_v^{i,track}$ sets of parameters, which were generated at the time of vehicle $v$’s registration, the identity of vehicle $v$ can be easily located by the central authority when $\sigma_v^i$ signature is suspected. From $H_v^i$, $hSID_v^i$, $pk_v^i$, and $T_v^i$, TA/PKG can find $h_1 = h_2^{-1}(pk_v^i|T_v^i).H_v^i \in Z_q^*$ and $SID_v^i = h_2^{-1}.hSID_v^i \in G$. Using them, TA/PKG calculates $h_1^{-1}.SID_v^i$ and looks for a matching stored $ID_v^{i,track}$ to locate the related $ID_v$. Entities other than TA/PKG cannot retrieve $ID_v$ since they neither have a stored $ID_v^{i,track}$ nor the large $q$ prime that is necessary for calculating the $h_2^{-1}$ modulus operation on curve $E$. Having comprehensively discussed our scheme and proved its security, below we introduce our technical approach through which we are assessing the resulted gain/effectiveness of PME-CPPAS.
6.4 Numerical Evaluations and Discussion of Gain

To investigate the suitability of PME-CPPAS for the real-time nature of VANETs, our numerical evaluations are presented. We show the effectiveness of our scheme in comparison with the other two pairingless approaches of [2, 3] as well as with other two pairing-based works of [4, 5]. At the outset and in accordance with the VANETs’s up-to-date standards specifications [7], we consider a network of $n$ vehicles that are registered with the central authority (following the phases of Table 6.3). The vehicles obtained their long-term real identities $ID_v$, all elliptic curve parameters, hashes ($h_1$, $h_2$, $h_3$), TA/PKG’s public key $pk_{TA/PKG}$, and their first-tier specific alias identities $SID^i_v$ and pseudo keys $sid^i_v$, where $v \in \{1..n\}$. After deployment, all vehicles are assumed to transmit their secure messages within the recommended $T = 100$ msec interval. We also assume the traffic, which each vehicle receives after deployment, to be equal to the number of received signatures $n$ from $n$ vehicles.

For the underlying cryptographic settings, we consider the elliptic curves recommendations of 256-bit public point keys, i.e., security level of $L = 128$ bit. Our pairingless PME-CPPAS scheme works on a non-singular prime field elliptic curve $E$ of prime $p$ at 256 bit and point order is $q = 256$ bit (we only consider the x-coordinate of the point). Since we are comparing with pairingless as well as pairing-based schemes, we need another designated pairing-friendly curve; pairing-friendly terminology refers to the curves that give non-degenerate pairings result as it is explained in appendix A.3. We choose supersingular curves ($y^2 \equiv (x^3 + x) \mod (p \equiv 3 \mod 4)$) or ($y^2 \equiv (x^3 + 1) \mod (p \equiv 2 \mod 3)$); such curves have a specific “distortion” map/rotation $\varphi$ to enable the non-degeneracy property for a symmetric pairing function $e : G \times G \rightarrow G_T$. $G_T$ is a subgroup (of a similar order to $G$) of a finite extension
field $F_{p^\varrho}$. The embedding $\varrho$ refers to extending the bit representation of the target group elements in $F_{p^\varrho}$; $\varrho$ can be 2, 3, 4, 6 and for any non-pairing operation, $\varrho = 2$ is sufficient in our scheme. Making $\varrho$ larger will not improve the scheme since we have to increase the prime $p$ of the base field. Nevertheless, since we are adopting such supersingular curve, we choose its prime to be Solinas primes (difference or sum of orders of 2). To achieve similar 128 bit security level, $p$ has to be 1582 bit, $q$ stays as 256 bit, co-factor is 1326 bit. Thus, extension field has 3072 bit elements when $\varrho = 2$. This feature ensures that the DLP in $F_{p^\varrho}$ is still hard\textsuperscript{21}. Table 6.4 summarizes the parameters values that are used in our evaluation. A thorough discussion on the choice of these values is in [7, 9, 11, 16].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$ Beacon Generation Period</td>
<td>100 msec</td>
</tr>
<tr>
<td>$L$ Security Parameter</td>
<td>128 bit</td>
</tr>
<tr>
<td>$t$ Security Parameter of The Small Exponent Test</td>
<td>5 bit</td>
</tr>
<tr>
<td>$p$ Finite Field Prime Order</td>
<td>$2^{2\times128}$ bit</td>
</tr>
<tr>
<td>$a, b, x, y$ Elliptic Curve Parameters</td>
<td>$\in F_{2^{256}}$ bit</td>
</tr>
<tr>
<td>#E Number of $E$’s Points</td>
<td>$(2^{256} + 1 \pm 2\sqrt{2^{256}})$</td>
</tr>
<tr>
<td>$q$ Order of an $E$’s Point</td>
<td>$2^{512}$ bit (x-coordinate has $2^{256}$ bit)</td>
</tr>
<tr>
<td>Hashes and Other Scalar Values</td>
<td>$\in \mathbb{Z}^{*}_{2^{256}}$ bit</td>
</tr>
<tr>
<td>$\varrho$ embedding degree</td>
<td>2</td>
</tr>
<tr>
<td>Solinas prime</td>
<td>$2^{1582}$ bit</td>
</tr>
</tbody>
</table>

To assess the gain of PME-CPPAS scheme in comparison with the other four approaches of [2–5], we utilize the Computation Complexity $C_C$ and Communication Overhead $C_O$ performance metrics as follows:

\textsuperscript{21}Another example, of pairing-friendly curves, is random non-supersingular curves. These curves are more efficient since they provide asymmetric pairing $e: \bar{G}_1 \times \bar{G}_2 \rightarrow \bar{G}_T$. Groups $\bar{G}_1$, $\bar{G}_2$, and $\bar{G}_T$ have same order, i.e., same number of elements, but $\bar{G}_1$ is composed of the points of elliptic curve over the base field $F_p$ while $\bar{G}_2$ and $\bar{G}_T$ are subgroups of the extension field $F_{p^\varrho}$, i.e., they have more number of bits to represent their elements. Shim [4] uses a good example of this kind of curves, which is the Miyaji-Nakabayashi-Takano (MNT) curves with $\varrho = 6$ [95].
6.4.1 Computation Complexity (bit)

Our computation complexity $C_C$ metric refers to the number of cryptographic operations needed to generate system parameters, extract alias identities and keys, sign outgoing messages, and verify received messages: individually and in batches. Since the generation/setup episode is conducted offline at PKG/TA, we do not include its calculations. Furthermore, we refer to the operations of both the extraction and the signing phases as “signature generation” because they are dynamically conducted for every to-be-broadcasted message. To achieve a fair assessment of PME-CPPAS scheme’s gain in comparison with [2–5], besides Table 3.2 and 4.1 in Chapter 3 and 4, we enlist in Table 6.5 the estimated big $O$ complexity notation of every used cryptographic operation in terms of $L$ and $t$ parameters. In Table 6.6, we straightforwardly count and identify the number of operations in each considered scheme.

<table>
<thead>
<tr>
<th>Used Cryptographic Operations and Their Complexity over $E$:</th>
<th>$O((t\log_2 q)) \rightarrow O(tL)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small Modular Multiplication (SMM): $(\hat{c} \hat{b})$</td>
<td>$O((t\log_2 q)) \rightarrow O(tL)$</td>
</tr>
<tr>
<td>Small Modular Addition (SMA): $(\hat{c} + \hat{c})$</td>
<td>$O((t\log_2 q)) \rightarrow O(tL^2)$</td>
</tr>
<tr>
<td>Small Scalar Multiplication (SSM) on $E$: $(\hat{c}\hat{P})$</td>
<td>$O((t\log_2 q)) \rightarrow O(tL^2)$</td>
</tr>
<tr>
<td>Hashing to Point (HP): ${0,1}^* \rightarrow G$</td>
<td>$O((t\log_2 q)) \rightarrow O(tL^2)$</td>
</tr>
<tr>
<td>Negation (NOT): $\overline{a}$ or $\overline{\hat{P}}$</td>
<td>Negligible</td>
</tr>
</tbody>
</table>

Based on such analysis, by plotting the curves for the signature generation phase, in Fig. 6.4 our authentication scheme enacts better effectiveness and less complex computations than Shim’s [4] and Bayat et al.’s [5]; since less scalar multiplication $SM$ operations are used and because Shim’s and Bayat et al.’s are pairing schemes, i.e., they use larger prime field; PME-CPPAS achieves a gain of 99.9% in comparison with

\[^22\text{Map to Point is a probabilistic function defined by Boneh and Franklin [96] to map arbitrary string \{0,1\}^* values into elliptic curve points. Since there are many ways to do it, its computation complexity is undecided, [97] tried to implement deterministic forms of it.} \]
### Table 6.6
Comparing PME-CPPAS Scheme with [2–5] In Terms of Computation Complexity $C_c$
(Offline Operations are Grayish-highlighted)

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Generate (Offline at PKG/TA)</th>
<th>Signature Generation (Extract and Sign Phases)</th>
<th>Single Verify (Online at Receiver)</th>
<th>Batch Verify (Online at Receiver)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shim [4]</td>
<td>In V2I: (2)SM</td>
<td>Offline at PKG/TA in V2I: (6)SM + (2)HS + (1)MM + (1)MA + (2)NOT + (1)PA</td>
<td>At TPD: (3)SM + (1)HS + (1)PA</td>
<td>At RSU in V2I: (3)WP + (n + 1)SM + (2n)HS + (n)MM + (n − 1)MA + (3n − 2)PA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In V2I: N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In I2V: N/A</td>
<td>Offline at PKG/TA in I2V: (2)SM + (1)HS + (1)MM + (1)MA</td>
<td>At TPD: (3)SM + (1)SM + (1)HS + (1)PA</td>
<td>At TPD in I2V: (3)WP + (2)SM + (n + 1)HS + (2n)MM + (n − 1)MA + (2n − 1)PA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In I2V: (2)PA</td>
<td></td>
</tr>
<tr>
<td>Bayat et al. [5]</td>
<td>(2)SM</td>
<td>At TPD: (6)SM + (1)HS + (1)HP + (2)NOT + (1)PA</td>
<td>At TPD: (3)WP + (1)SM + (1)HS + (1)PA</td>
<td>At TPD: (3)WP + (n)SM + (n)HS + (n)HP + (3n − 3)PA</td>
</tr>
<tr>
<td>He et al. [3]</td>
<td>(1)SM</td>
<td>At TPD: (4)SM + (2)HS + (2)NOT + (1)MM + (1)MA + (1)PA</td>
<td>At TPD: (3)SM + (1)SM + (2n)HS + (n)SSM + (2n − 2)MA + (2n)PA</td>
<td>At TPD: (n + 2)SM + (3n)SSM + (2n + 1)SM + (n − 1)PA + (n − 1)SMA</td>
</tr>
<tr>
<td>Lo and Tsai [2]</td>
<td>In V2I: (2)SM</td>
<td>Offline at PKG in V2I: (5)SM + (2)HS + (1)MM + (1)MA + (2)NOT + (1)PA</td>
<td>In V2I: (1)SM + (1)HS + (1)PA</td>
<td>At TPD in I2V: (n + 3)SM + (n + 1)HS + (n)SSM + (n − 1)PA + (n − 1)SMA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In V2I: N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In I2V: N/A</td>
<td>Offline at PKG in I2V: (1)SM + (1)HS + (1)MM + (1)MA</td>
<td>In I2V: (1)SM + (3)SM + (1)HS + (2)PA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In I2V: (2)PA</td>
<td></td>
</tr>
<tr>
<td>PME-CPPAS</td>
<td>(1)SM</td>
<td>Offline at PKG: (3)SM + (1)MA + (1)HS + (1)MM + (1)MA</td>
<td>At OBU: (2)SM + (3)SM + (1)HS + (2)PA</td>
<td>At OBU: (n + 2)SM + (n)HS + (3n)SSM + (2n − 2)MA + (2n)PA + (n)SSM</td>
</tr>
</tbody>
</table>
these two pairing-based schemes for $L = 128$ security level. Moreover, our scheme achieves 63.3% improvement in comparison with He et al.’s approach; although [3] has less $SM$ operations in the signing phase, the overhead comes from the extraction of alias identities and keys that occurs at the TPD of the sender. However, in comparison to Lo and Tsai’s signing phase, our scheme experiences slightly more computation complexity $C_C$ than theirs; the reason is their use of a single scalar multiplication $SM$, whether in the V2I or the I2V version. Finally, when is compared to PKI, PME-CPPAS achieves 33.33% less signing computation cost.

![Figure 6.4. A Comparison of Signature Generation (Extraction and Signing) Phase](image)

As we stated earlier, when the broadcasted message $<m^i_v, T^i_v, pk^i_v, hSID^i_v, H^i_v, \sigma^i_v>$ is received, it can be verified either individually or in batches. We inspect in Fig. 6.5 the gain of our scheme in the single verification case. Noticeably, our PME-CPPAS, He et al.’s, and Lo and Tsai’s schemes are equally lowering the cost of the computation $C_C$. All three approaches enact 53.84% less $C_C$ overhead than PKI and 99% less computation than both [4] and [5]. Such gain comes from the elimination of the complex and time-consuming pairings operation on elliptic curves; basically, this is the main motivation behind developing pairingless schemes.
Figure 6.5. Single Verification Phase Comparison

To appraise the effectiveness comparison in the batch verification case, we plot the curves of Fig. 6.6 when $L = 128$ with respect to the number of vehicles/signatures $n$ at a designated value of the small scalar exponent test parameter $t$ equals 5. For a low density network of $n = 5$, PME-CPPAS, along with [3] and V2I version of [2], gives 99% gain over Shim V2I approach and Bayat et al.’s scheme for $L = 128$. In comparison to PKI, PME-CPPAS is still 11.9% less efficient than its verification phase. For higher traffic density of 200 signatures, PME-CPPAS, He et al.’s, and Lo and Tsai’s equally score 99% better than Shim’s and Bayat et al.’s for $L = 128$. The reason is that even pairing-based schemes have a sizable pairing operation, batch verification accumulates an $n$ power of scalar multiplications thus slows down the performance. Still, PKI is 89.7% less complicated. A suggested solution is to either eliminate the batch verification or conduct a modified form of the small exponent test to reduce the $O(L^3)$ to $O(t.L^2)$ or to $O(t^2.L)$; this is one of our future extensions.
6.4.2 Communication Overhead (byte)

The other metric that we utilize to assess the performance of our scheme is the communication overhead $C_O$. $C_O$ is the size of the beacon to be transferred including the security part. In the existing adopted standards of VANETs, PKI framework entails a high communication overhead as we stated in Table 2.1. Therefore, identity-based authentication schemes have been introduced to overcome such limitation. In this sense, we want an authentication scheme that has smaller security portion size to be attached with the plain safety message. As can be seen in Fig. 6.7, on one hand, in comparison with the PKI-based authentication when $L = 128$, PME-CPPAS has smaller beacon size; 328 byte in PME-CPPAS compared to 360 byte in PKI. PME-CPPAS has similar $C_O$ to He et al.’s [3]. In comparison to Lo and Tsai’s V2I version, PME-CPPAS achieves a 8.8% improvement at $L = 128$. On the other hand, our approach has 72.4% less $C_O$ than Shim V2I [4] and 58.69% Bayat et al. [5]. The reason is that the security payload in [4] consists of five elliptic curve point elements and in [5] it has three supersingular elliptic curve point elements; in comparison, we have only two points and two scalars (all over a non-singular curve of $p = q = 256$ bit).
To spotlight the effect of the size of the communicated payload on VANETs, we can estimate the maximum number of vehicles $N_C$ that can simultaneously share the wireless medium. Because the current VANETs’ standards limit the allowed data transmission rate between 3 and 27 Mbps and the width of the CCH on which the safety beacons are transmitted to 46 msec [7], the number of vehicles that can capture the medium is restricted. In this sense, smaller beacon size gives room to more vehicles to share the medium. In Fig. 6.8, the curves of the number of the medium-capturing vehicles are comparable in PME-CPPAS and He et al.’s; both are better than PKI and Lo and Tsai’s. The difference is noticeable when we compare with Bayat et al.’s, and Shim’s.

![Figure 6.7. Comparing Communication Overhead $C_O$ (bytes)](image1)

![Figure 6.8. Number of handled Vehicles $N_C$ Comparison](image2)
In summary, this full computation-wise and communication-wise metrics comparison shows the effectiveness of the proposed authentication approach; PME-CPPAS is better in both the signing and single verification phases. In batch verification, it gives comparable performance to [3] and [2]. Since PME-CPPAS eliminates the need for bilinear pairing, it conducts less computations in comparison with Shim [4] and Bayat et al. [5] pairing-based CPPA schemes.
Chapter 7: Context-Aware Authentication Interchange Scheme (CAAIS) in Vehicular Networks

The availability of GS authentication that has better anonymity than PKI on one hand and the IBC authentication that has less overhead than PKI on the other hand, have motivated us to present our CAAIS scheme. In CAAIS, we interchange between the three frameworks to match the vehicles’ context with the goal of optimizing performance of the network in terms of computation, communication, and verification costs. To the best of our knowledge, no study has been dedicated to investigate interchanging between the three frameworks in VANETs; there is one effort on mixing GS and PKI to produce a hybrid authentication with better performance [39]. In another literature, Sun et al. [98] studies teaming up GS and IBC to achieve better privacy and authentication for vehicles. Before we explore the presented approach, we first introduce our new identity-based authentication framework that will serve the role of the identity-based authentication framework in CAAIS.

7.1 New IBC Authentication Framework

The available pairingless IBC authentication schemes [2, 3, 6] can be good candidates to utilize in CAAIS; they have a comparable computation and communication
costs. However, each scheme has its own not very well-tuned traits; therefore, we derive a neater and a better calibrated pairingless IBC authentication version. Our preferred version is a fine-tuned mixture of [2,3,6] that achieves better vehicle anonymity by dynamically randomizing the assigned alias identities. Before we describe the suggested authentication framework, we list in Table 7.1 the main spotted differences between it and the schemes of Lo and Tsai [2], He et al. [3], and our previous work of PME-CPPAS [6].

### Table 7.1

**Spotted Differences Between The New IBC Authentication and [2,3,6]**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Addressed setting</td>
<td>Only V2I and I2V</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
</tr>
<tr>
<td>Addressed authority</td>
<td>TA and PKG</td>
<td>TA</td>
<td>TA and PKG</td>
<td>TA and PKG</td>
</tr>
<tr>
<td>Publishing the secret key of authority</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>How is the real identity (ID_v), of each registree vehicle, is assigned</td>
<td>Not addressed</td>
<td>Not addressed</td>
<td>Addressed</td>
<td>Addressed</td>
</tr>
<tr>
<td>Number of alias identities and who generates them</td>
<td>(SID_{v,1}^i) at (v)</td>
<td>(SID_{v,1}^i) and (SID_{v,2}^i) at (v)</td>
<td>(SID_v^i) at TA</td>
<td>(SID_{v,1}^i) and (SID_{v,2}^i) at (TA)</td>
</tr>
<tr>
<td>Number of alias keys and who generates them</td>
<td>(sid_{v,1}^i) and (sid_{v,2}^i) at (v)</td>
<td>(sid_v^i) at (v)</td>
<td>(sid_v^i) at TA</td>
<td>(sid_v^i) at PKG</td>
</tr>
<tr>
<td>Beaconing the alias key</td>
<td>(v) sends (sid_{v}^{i,1})</td>
<td>Kept private</td>
<td>Kept private</td>
<td>Kept private</td>
</tr>
<tr>
<td>Direct use of given alias identities</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

As in [3,6], the framework is suggested for any setting: V2V, V2I, and I2V. Our authentication comprises four phases: Setup, Alias Identities and Keys Generation and Extraction, Signing, and Verification (single and batch). We prefer the style of [2] of distinguishing between the two different authorities TA and PKG, where
TA creates the alias identities and the PKG derives the alias keys. However, to simplify description, we consider TA and PKG to have the same secret and public keys; rather than \((sk_{TA}, pk_{TA})\) and \((sk_{PKG}, pk_{PKG})\), we have \((sk_{TA/PKG}, pk_{TA/PKG})\).

At \(L\) security level, the setup phase starts at TA/PKG by choosing elliptic curve parameters: \(2L\) bit prime \(p\), Galois field \(F_p\), and elliptic curve equation \(E : y^2 \equiv (x^3 + ax + b) \mod p\), where \(a, b, x, y \in F_p\). On such \(E\), a group \(G\) of \((x, y)\) points is chosen; if generator \(P\) of the group equals the generator of \(E\), hence \(G = E\) and \(P\) will have an order \(q\) of \(|q| = 2L\) bit-length. After generating \(sk_{TA/PKG} \in \mathbb{Z}_q^*\) and \(pk_{TA/PKG} = sk_{TA/PKG}.P \in G\), TA/PKG will have \((p, q, P, pk_{TA/PKG})\) as well as five hash functions \(h_1, h_2, h_3, h_4, h_5\) (all map \(\{0, 1\}\) to \(\mathbb{Z}_q^*\)) to be given to every registree \(v\). In contrast to He et al. [3], the TA/PKG in our approach as well as in [2, 6] keeps \(sk_{TA/PKG}\) private rather than distributing it to vehicles; He et al. [3] publishes the key to vehicles supported by the availability of secure TPDs, yet neglecting the chances of physical attacks. Along these elliptic curves parameters, vehicle \(v\) also obtains a real point identity \(ID_v\); unlike [2, 3], as in 6, we precisely describe how the real identity \(ID_v\) is calculated as

\[
ID_v = \hat{b}^i.P \mod q \in G, \tag{7.1}
\]

where \(\hat{b}^i \in \mathbb{Z}_q^*\).

Next, TA conducts the pseudo identities generation phase where \(v\) obtains two sets of first-tier alias identities; one set have randomly generated points \(SID_v^{i,1}\) while the second set of points \(SID_v^{i,2}\) is derived from the real appointed identity \(ID_v\), the first alias identity, and TA’s secret and public keys with the help of \(h_1\) as

\[
SID_v^{i,2} = ID_v \oplus h_1(sk_{TA}, SID_v^{i,1}, pk_{TA}) \mod q \in G. \tag{7.2}
\]
In this regard, our scheme contrasts [3, 6] since it generates double alias identities. In comparison to [2], the expense of generating $SID_v^{i,2}$ is offloaded on TA, hence more anonymity is added to the generated identity and the contained clutter and derangement of [2] is removed. Succeeding the alias identity generation is the pseudo key extraction phase. In our scheme, $sid_v^i$ represents a set of alias keys in $Z_q^*$ to be extracted at PKG from the calculated set of anonymous identities $SID_v^i = (SID_v^{i,1} | SID_v^{i,2})$ and from $sk_{PKG}$ with the help of $h_2$:

$$sid_v^i = \hat{d}_i + h_2(SID_v^i).sk_{PKG} \mod q. \quad (7.3)$$

As in [3, 6], we adhere to generating only a single set of keys $sid_v^i$. Despite the fact that Lo and Tsai’s scheme [2] generates double sets of pseudo keys, they always need to plainly beacon one key with every sent message to have valid verification at receivers. Such feature vitiates their double keys betterment, weakens the unlinkability of the scheme, and adds extra communication overhead; in other words, creating single key and keeping it undisclosed is better than doubling the keys, yet exposing one of them.

By the end of these setup and extract phases, TA/PKG can keep record of every registered vehicle $v$ in the form of $< v, ID_v, \{SID_v^{i,1}, SID_v^{i,2}, sid_v^i, \hat{b}_i, \hat{d}_i, ID_v^{i,track}\} >$. Each $ID_v^{i,track}$ is calculated as

$$h_1^{-1}(sk_{TA}.SID_v^{i,1}, pk_{TA}).ID_v \mod q \in G \quad (7.4)$$

to be used by TA/PKG for tracing liable vehicles as we see in Section 7.4. After deployment, the signing phase at senders is carried out online whenever $v$ wants to broadcast a beacon; to sign a message $m_v^i \in \{0, 1\}^*$ at $T_v^i$ timestamp, $v$ dynamically generates secret $sk_v^i \in Z_q^*$ and public $pk_v^i = sk_v^i.P \in G$ keys to ensure anonymity.

From the given fixed first-tier $SID_v^{i,1}$ and $SID_v^{i,2}$ alias identities, $v$ extracts second-tier
dynamic alias identities using $h_3$ and $h_4$ hashes with $T_i^v$ timestamp:

$$h_{3,4}(SID_i^v|T_i^v) = h_3(SID_i^{i_1}|T_i^v)|h_4(SID_i^{i_2}|T_i^v).$$

(7.5)

By adding such hashing-flavor to the created alias identities, we increase the randomness of the generated signature $\sigma_i^v$. Making use of the given alias key, the dynamically generated secret and public keys, and hashed alias identities,

$$\sigma_i^v = sid_i^v + h_5(pk_i^v|h_{3,4}(SID_i^v|T_i^v)|m_i^v|T_i^v).sk_i^v \mod q$$

(7.6)

represents the $v$’s scalar signature on message $m_i^v$ that is broadcasted by $v$ in the $<m_i^v, T_i^v, pk_i^v, SID_i^v, \sigma_i^v>$ beacon.

At receivers, verification phase of received beacons can be carried out either individually or collectively. To verify the validity of a single received tuple $<m_i^v, T_i^v, pk_i^v, SID_i^v, \sigma_i^v>$ by any receiving end $w$ whether it is a vehicle or an infrastructure, $w$ has to have valid central authority parameters $(p, q, P, pk_{TA/PKG}, h_1, h_2, h_3, h_4, h_5)$. After checking the freshness of the received timestamp $T_i^v$, using these parameters $w$ can easily verify whether

$$\sigma_i^v.P \equiv SID_i^{i_1} + h_5(pk_i^v|h_{3,4}(SID_i^v|T_i^v)|m_i^v|T_i^v).pk_i^v + h_2(SID_i^v).pk_{TA/PKG}$$

(7.7)

holds. For a collective verification of $n$ received signatures from $n$ vehicles, $w$ can sum up the validation process of (7.7) in one single equation

$$\left(\sum_{i=1}^{n} \alpha_i^{i}.\sigma_i^v\right).P \equiv \left(\sum_{i=1}^{n} \alpha_i^{i}.SID_i^{i_1}\right) + \left(\sum_{i=1}^{n} \alpha_i^{i}.h_5(pk_i^v|h_{3,4}(SID_i^v|T_i^v)|m_i^v|T_i^v).pk_i^v\right)$$

$$+ \left(\sum_{i=1}^{n} \alpha_i^{i}.h_2(SID_i^v).pk_{TA/PKG}\right),$$

(7.8)

where $\alpha_i^{i}\{|1\leq i \leq n\}$ is a small integer ($\neq 0$) of length $t$-bit. The use of $\alpha_i^{i}$ is necessary to disclose any modification that might happen to the received beacons [75]; $\alpha_i^{i}$ is
part of a cryptographic small exponents test [89] of security parameter equals $t$. In addition, the small size of $t$ helps reducing the computation complexity of the used cryptographic operations in (7.8) as in Section 7.5. By finalizing the details of the verification phase, we have provided in this section a thorough description of our new IBC authentication scheme. Table 7.2 sketches the details of the proposed framework phases. Proving the security of the new IBC authentication framework is postponed to Section 7.4. Also, we are adding one more comparison, in terms of computation complexity, between this IBC authentication and [2, 3, 6]’s in Section 7.5 to enhance our claim of introducing a neater authentication framework. Next section utilizes this secondary contribution to serve the role of the needed identity-based security framework under the new CAAIS umbrella.

7.2 Considered Network Model and Assumptions

Our network (Fig. 7.1) is assumed to have a central authority that is comprised of a CA to generate certificates, a TA to generate alias identities, a GM to generate group signature ingredients, and a PKG to create pseudo keys. The CA/TA/GM/PKG co-operate before and after the deployment of the network to serve the needs of $n$ registered OBU’s and infrastructure RSUs. Before deployment, these authorities agree on the used elliptic curve parameters mentioned in Section 7.1 at security level $L$. With these parameters, CA/TA/GM/PKG authorities create their own secret and public keys:

- $CA : \quad sk_{CA} \rightarrow pk_{CA} = sk_{CA} \cdot P,$
- $TA : \quad sk_{TA} \rightarrow pk_{TA} = sk_{TA} \cdot P,$
- $PKG : \quad sk_{PKG} \rightarrow pk_{PKG} = sk_{PKG} \cdot P,$ and
- $GM : \quad gsk_{GM} \rightarrow gpk_{GM} = gsk_{GM} \cdot P.$
Table 7.2
New IBC Authentication Scheme

<table>
<thead>
<tr>
<th>Before Deployment: Setup at TA/PKG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a, b, p, q, P, h_1, h_2, h_3, h_4, h_5$</td>
</tr>
<tr>
<td>$sk_{TA/PKG} \in \mathbb{Z}_q^*$</td>
</tr>
<tr>
<td>$pk_{TA/PKG} = sk_{TA/PKG}.P$</td>
</tr>
<tr>
<td>Publish $(p, q, P, pk_{TA/PKG}, h_1, h_2, h_3, h_4, h_5)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Before Deployment: Registration at TA/PKG and Alias Identities and Keys Extraction at TA/PKG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Vehicle \rightarrow TA/PKI$</td>
</tr>
<tr>
<td>$v$</td>
</tr>
<tr>
<td>$\hat{b} \in \mathbb{Z}_q^*$</td>
</tr>
<tr>
<td>$ID_v \leftarrow \hat{b}.P$</td>
</tr>
<tr>
<td>$\hat{d} \in \mathbb{Z}_q^*$</td>
</tr>
<tr>
<td>$SID_i^{i,1} \leftarrow SID_i^{i,1} = \hat{d}.P$</td>
</tr>
<tr>
<td>$SID_i^{i,2} \leftarrow SID_i^{i,2} = ID_v \oplus h_1(sk_{TA}.SID_i^{i,1}, pk_{TA})$</td>
</tr>
<tr>
<td>$SID_v^i = (SID_v^{i,1}, SID_v^{i,2})$</td>
</tr>
<tr>
<td>$ID_v^{i,\text{track}} = h_1^{-1}(sk_{TA}.SID_i^{i,1}, pk_{TA}).ID_v \mod q$</td>
</tr>
<tr>
<td>$sid_v^i \leftarrow sid_v^i = \hat{d} + h_2(SID_v^i).sk_{PKG} \mod q$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>After Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Signing \rightarrow Verifying Signature$</td>
</tr>
<tr>
<td>$sk_v^i \in \mathbb{Z}_q^*$</td>
</tr>
<tr>
<td>$pk_v^i = sk_v^i.P$</td>
</tr>
<tr>
<td>$m_v^i \in {0, 1}^*$</td>
</tr>
<tr>
<td>$T_v^i$</td>
</tr>
<tr>
<td>$SID_v^i$</td>
</tr>
<tr>
<td>$h_3,4(SID_v^i</td>
</tr>
<tr>
<td>$\sigma_v^i = sid_v^i + h_5(pk_v^i</td>
</tr>
<tr>
<td>$\sigma_v^i.P \equiv SID_v^{i,1} +$</td>
</tr>
<tr>
<td>$h_5(pk_v^i</td>
</tr>
<tr>
<td>$h_2(SID_v^i).pk_{TA/PKG}$</td>
</tr>
<tr>
<td>$\left(\sum_{i=1}^{n} \alpha_i.\sigma_v^i \right).p \equiv$</td>
</tr>
<tr>
<td>$\left(\sum_{i=1}^{n} \alpha_i.SID_v^{i,1} \right) +$</td>
</tr>
<tr>
<td>$\left(\sum_{i=1}^{n} \alpha_i.h_5(pk_v^i</td>
</tr>
<tr>
<td>$\left(\sum_{i=1}^{n} \alpha_i.h_2(SID_v^i).pk_{TA/PKG} \right)$</td>
</tr>
</tbody>
</table>

139
When any vehicle manufacturer approaches the authorities to register any vehicle \( v \), under the consent of some legal authority, the registree obtains all of the ingredients of the three PKI, GS, and the new IBC authentication schemes besides the elliptic curve parameters \((p, q, P)\), the four hash functions \((h_1, h_2, h_3, h_4, h_5)\) that map to \(Z_q^*\), and the long term identity \(ID_v \in G\) from (7.1). Since both PKI and IBC use same \((sk_v^i \in Z_q^*, pk_v^i = sk_v^i \cdot P \in G)\), we can make any of the following assumptions. Either \( v \) itself generates the \((sk_v^i \in Z_q^*, pk_v^i = sk_v^i \cdot P \in G)\) offline when it registers with the central authority; \( v \) offline obtains their \(cer_{CA}(pk_v^i)\) from CA. In this assumption, \( v \) uses the same \((sk_v^i, pk_v^i)\) for both PKI and IBC; this saves storage space and computation overhead. The other assumption is to give \( v \) two different sets of keys: \((sk_v^i, pk_v^i, cer_{CA}(pk_v^i))\) obtained offline from CA for PKI authentication and \((sk_{ibc}^i, pk_{ibc}^i)\) set generated online by \( v \) as \(((\hat{d}^i \cdot sk_v^i), (\hat{d}^i \cdot pk_v^i))\) for IBC authentication where \( \hat{d}^i \in Z_q^* \). After deployment, vehicles start with the PKI security framework where they periodically change their short-lived certificated pseudonyms to deceive eavesdroppers. Along the mentioned cryptographic ingredients, every vehicle \( v \) is assumed to have:

Figure 7.1. Assumed Network Model

140
• A Context Timer $T_{cv}$ that when times out, $v$ checks its surrounding context. The initial value of $T_{cv}$ is set in compliance with the settings of DSRC [7]; we can make $T_{cv}$ hit zero every pseudonym lifetime, i.e., every 60 seconds [15]. Or, we can relate $T_{cv}$ with the traveled distance of $v$ [11]; if $v$ travels at 112 km/h on US highways, a PKI beacon of a 310 byte, transferred at 6 Mbps rate, needs 0.414 msec to reach the receiver ignoring any medium access contention. Adding the signing and verifying computation costs of 2.58 msec, each beacon needs a 2.99 msec communication time. For both ends of communication, per a total time of 5.988 msec, $v$ is moving only 0.18 m. If $v$ wants to examine its context every 10 m, $T_{cv}$ can be set to 332.6 msec.

• To illustrate vehicle $v$’s surrounding context, we assume the availability of a

  – Number of Neighbors Counter $N_C$: also, $N_C$ can refer to the number of beacons that $v$ receives during its context timer interval $T_{cv}$. $N_C$ is initialized to ($> 1$) value incrementing with every beacon reception after deployment. When $N_C$ exceeds a specific threshold $\xi_{N_C}$, $v$ switches to a lighter-weight authentication such as our new IBC scheme; otherwise, if $N_C$ is small, $v$ can accommodate heavyweight authentications such as PKI and GS.

  – Vehicle Anonymity $A_V$: our settings interprets vehicle anonymity in the terms of the total entropy of the received beacons within a $T_{cv}$ context timer interval as

$$A_V(T_{cv}) = \frac{\sum_{i=0}^{N_C \cdot T_{cv}} (pg_v^i \cdot \log \left( \frac{1}{pg_v^i} \right))}{\log(N_C \cdot T_{cv})},$$

where the power gain $pg_v^i$, between vehicles $v$ and $i$, refers to the signal strength of the received beacon from $i$ by $v$ within the $T_{cv}$ time interval. We assume $pg_v^i$ to be normalized within the $[0, 1]$ range and $A_V$’s initial value is obtained after an elapsed time of one beacon interval $T = 100$ msec. The value of $A_V$ increases by
the growth of the network. When the number of neighbors or received beacons is small and is under a designated threshold $\xi_{A_V}$, $v$ has to switch to a more anonymous authentication such as group signature authentication; once $N_C$ starts growing, the risk is lifted and $v$ can switch to an authentication that does not need to be super anonymous.

- Maximum Tolerated Communication Overhead $C_O$: is the multiplication of the allowed number of vehicles in the network and the broadcasted beacon size $R$. $C_O$ increases either when beacon size is large or more vehicles are presented in the network. However, due to the limited CCH 46 msec width [7], the number of vehicles that can capture the wireless medium is limited; therefore, the increment of $C_O$ is caused by larger beacon sizes, i.e., type of authentication scheme. CAAIS attempts to keep $C_O$ within a fixed limit $\xi_{C_O}$; we achieve this by letting the vehicles change their authentication to the lighter IBC framework when $\xi_{C_O}$ is reached; otherwise, each $v$ has the freedom to use any authentication. As with $A_V$, the initial value of $C_O$ is decided after an elapsed time of one $T$ beacon interval.

The assigned $<\xi_{N_C}, \xi_{A_V}, \xi_{C_O}>$ thresholds values are carefully chosen in accordance with DSRC specifications and with our own design preference, as in Section 7.5.

### 7.3 CAAIS’s Authentication Interchange Procedure

To fully and comprehensively describe CAAIS, we illustrate two scenarios in which vehicle $v$ can call the CAAIS authentication interchange procedure depending on the current status of $v$’s context. The initial values of the three $N_C$, $A_V$, and $C_O$ context parameters are decided before deployment. After deployment, they are updated with every new beacon arrival from neighboring vehicles/infrastructure. Since beacons are
periodically broadcasted every $T$ seconds, their arrival rates to $v$ are deterministic and equal $T$. We also assume the number of cars in $v$’s context to be random, but does not exceed the limit that will be set in Section 7.5. Thus, the total number of beacons received per second will be $N_C T$. Moreover, the obtained signal strength $p_{g_i}^v$ of the received beacons are normalized to random probability values between 0 and 1. When $T_{cv}$ times out, the updated $<N_C, A_V, C_O>$ parameters are compared to the assigned thresholds; $v$ decides to keep its current authentication or to change to an alternative framework depending on the result of the considered comparison.

### 7.3.1 CAAIS Without Grouping Scenario

In this first scenario, a general simple case is discussed; all vehicles in the network are assumed to be individuals, i.e., no groups/clusters/zones are formed and all vehicles’ beacons have the same priority. The Pseudocode Algorithm 1 is a self-explanatory outline of CAAIS in this case. The considered comparison starts at line 29 after updating all context parameters values and $T_{cv}$ is timing out. If the number of neighbors $N_C$ is greater than the designated threshold, $v$ switches to the lightweight IBC authentication to relax the overall network overhead. Otherwise, if the number of vehicles still within limits, the algorithm checks the total communication overhead $C_O$ from $v$’s point of view. If it is greater than the assigned threshold, the vehicle keeps its PKI authentication; otherwise, $v$ decides to switch to GS framework if the obtained degree of anonymity $A_V$ is very low (this means that $v$’s context is very scarce) or $v$ keeps its PKI framework.

**Algorithm 1 CAAIS Scheme For Individual Vehicles Scenario**

Before Deployment: Initial Settings at Vehicle $v$

Initialize:
1: p, q, P, E, G, h₁, h₂, h₃, h₄, h₅
2: IDᵣ
3: \{skᵣ, pkᵣ, cer_CA(pkᵣ)\}, pk_CA
4: gskᵣ, gpk_GM
5: \{SIDᵣ ← (SIDᵣ¹, SIDᵣ²), sidᵣ\}, pk TA/PKG
6: Tᵢn
7: Tᵥin
8: N₉in, AVin, COin
Input: ξN, ξAV, ξCO

After Deployment:
9: Initialize i
10: Generate mᵣ, Tᵣ
11: Auth (mᵣ, Tᵣ) ← PKI(mᵣ, Tᵣ)
12: while i < Run do
13: \(T_{CV} ← T_{CVin}\)
14: while \(T_{CV} > 0\) do
15: \(N_C ← N_{9in}\)
16: \(A_V ← A_{9in}\)
17: \(C_O ← C_{Oin}\)
18: for T ← 1, Tᵢn do
19: Broadcast \((mᵣ, Tᵣ, Auth(mᵣ, Tᵣ))\)
20: Receive \((m_w, T_w, Auth(m_w, T_w))\)
21: \(N_C ← N_C + 1\)
22: \(A_V ← A_V + (pq_w * log(\frac{1}{pq_w}))\)
23: \(T_{CV} ← T_{CV} - 1\)
24: Generate mᵣ, Tᵣ
25: end for
26: end while
27: \(A_V ← \frac{A_V}{log(N_C)}\)
28: \(C_O ← C_O + Auth(mᵣ, Tᵣ) * N_C\)
29: if \(N_C > ξN_C\) then
30: Auth(mᵣ, Tᵣ) ← IBC(mᵣ, Tᵣ) and \(N_C ← ξN_C\)
31: else if \(C_O ≥ ξCO\) then
32: Auth(mᵣ, Tᵣ) ← PKI(mᵣ, Tᵣ)
33: else if \(A_V < ξAV\) then
34: Auth(mᵣ, Tᵣ) ← GS(mᵣ, Tᵣ)
35: else
36: Auth(mᵣ, Tᵣ) ← PKI(mᵣ, Tᵣ)
37: end if
38: Record \(N_C, AV, CO\)
39: i ← i + 1

Elliptic Curve
Real ID
PKI
GS
IBC
Beaconing Rate: Initial Value
Context Checking Timer: Initial Value
Initial Context Parameters
Context Thresholds
Number of Runs
Start Context Checking
40: end while
41: procedure PKI \((m_v, T_v)\)
42: \(Auth(m_v, T_v)\) from (2.5)
43: return \(Auth(m_v, T_v)\)
44: end procedure
45: procedure GS \((m_v, T_v)\)
46: \(Auth(m_v, T_v)\) from (2.4)
47: return \(Auth(m_v, T_v)\)
48: end procedure
49: procedure IBC \((m_v, T_v)\)
50: either \(pk_{ibc} \leftarrow pk_v\)
51: or \(d \in \mathbb{Z}_q^*, pk_{ibc} \leftarrow d.pk_v\)
52: calculate \(\sigma_v\) from (7.6)
53: \(Auth(m_v, T_v) \leftarrow pk_{ibc}, SID_v, \sigma_v\)
54: return \(Auth(m_v, T_v)\)
55: end procedure

A simple illustrative example is given in Table 7.3 where \(v\) has three different cases of changing context. For \(\xi_{N_C} = 4, \xi_{A_V} = 4,\) and \(\xi_{C_O} = 0.8 * \xi_{N_C} * \text{Beacon Size}(GS),\) \(v\) decides to switch to another authentication scheme. When the number of neighbor is \(7 > \xi_{N_C},\) \(v\) chooses our new IBC framework. When \(N_C = \xi_{N_C} = 4,\) \(v\) checks the communication overhead and anonymity value and decides to keep its PKI authentication. When the obtained anonymity degree is so low and \(N_C = 2,\) \(v\) prefers to be in a more private atmosphere by switching to the super anonymous GS authentication.

### 7.3.2 CAAIS With Clustering Scenario

In this setup, vehicles can be either leaders \(CL\) whose beacons have higher priority to be sent on the wireless medium or members \(CM\) of groups. The leadership and membership roles are decided after deployment. However, before deployment, each vehicle \(v\) is assumed to be a single cluster leader \(CL\) of a Cluster Size \((CS)\) parameter equals 1. Vehicles prefer to have larger \(CS\) size. Also, each \(v\) will have a Number of Leaders parameter \(N_C\) with threshold \(\xi_{N_C}\) and Cluster Members Vector \((CMV)\) that
Table 7.3
THREE CASES OF CONTEXT CHANGING AROUND VEHICLE \( v \) FOR \( \xi_{N_C} = 4, \xi_{A_V} = 4, \) AND \( \xi_{C_O} = 0.8 \times \xi_{N_C} \times \text{Beacon Size}(GS) \)

<table>
<thead>
<tr>
<th>( N_C )</th>
<th>2</th>
<th>7</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_V )</td>
<td>( \sum_{i=1}^{2}(pg_{v_i}^l,\log(\frac{1}{pg_{v_i}^l})) ) / ( \log(2) )</td>
<td>( \sum_{i=1}^{7}(pg_{v_i}^l,\log(\frac{1}{pg_{v_i}^l})) ) / ( \log(7) )</td>
<td>( \sum_{i=1}^{4}(pg_{v_i}^l,\log(\frac{1}{pg_{v_i}^l})) ) / ( \log(4) )</td>
</tr>
<tr>
<td>( C_O )</td>
<td>( C_O \text{in} + (N_C * \text{Beacon Size}) )</td>
<td>( N_C &gt; \xi_{N_C} \rightarrow IBC )</td>
<td>( (N_C = \xi_{N_C}) \land (C_O &lt; \xi_{C_O}) \land (A_V &gt; \xi_{A_V}) \rightarrow PKI )</td>
</tr>
</tbody>
</table>

has the form of

\[
CMV = [CL_{v_{in,fo}}, CM_{1_{D_{CL_{in,fo}}}}, ...]. \tag{7.10}
\]

\( CMV \) includes the information of the cluster leader (its current hashed alias identity \( h_1(SID^v) \), current location, current speed, current direction, etc) as well as the relative information of all members in his cluster. The \( CMV \)’s initial value is one of the hashed alias identities \( h_1(SID^v) \) that was given to \( v \) at registration as part of the IBC authentication. When new vehicles join the cluster, as in Chapter 5, the set is updated with the pseudo identities of the joined members. These new parameters are given to \( v \) besides the cryptographic parameters of the three authentication frameworks, the three context checking parameters with their thresholds, and the power gain \( pg^w_v \) of the signal between \( v \) and other vehicle/infrastructure \( w \). When the network is active, vehicles start beaconing their new

\[
m^i_v = < \text{command}^i_v, CS, CMV, T^i_v, \text{Auth}(m^i_v, T^i_v) > \tag{7.11}
\]
styled beacons every $\mathcal{T}$ seconds. These clusters are formed and modified based on the received signal strength values and command $c_i$, value that can be either create$-group$, create$-ack$, or join$-group$; a full description of cluster formation and cluster modification is previously given in Chapter 5. Having formed and updated clusters, in Fig. 7.2, we tackle a simple example to help us delineate our CAAIS interchange protocol.

In this example, vehicles $v$, 2, and $n$ become CL leaders with cluster sizes 2, 3, and 1, respectively. For a leader vehicle $v$, which has set its $\mathcal{T}_{C_v}$ context timer to a certain initial value, $v$ updates its context parameters $<\mathcal{N}_c, \mathcal{A}_v, \mathcal{C}_o, \mathcal{N}_l>$ upon reception of new beacons in the form of (7.11). Received beacons might come from $v$’s CM members, from leaders 2 and $n$, from the CM members of the other two clusters, and from the infrastructure RSU. When $\mathcal{T}_{C_v}$ hits zero, $v$ checks its context as in Algorithm 2. When the network comprises large number of clusters (line 45 in Algorithm 2), our CAAIS chooses to use the IBC authentication for all inter-cluster and intra-cluster communications if the cluster size is large to minimize the overhead of such dense network; otherwise, for small clusters, $v$ specifies different authentications to increase the anonymity inside the cluster and to decrease the overhead of
communication between leaders and infrastructure. For the case of small number of
CL leaders/clusters (line 54 in Algorithm 2), our algorithm also chooses to alternate
the use of authentications to preserve v’s anonymity and efficiency.

Algorithm 2 CAAIS Scheme For Clustering Scenario

Before Deployment: Initial Settings at Vehicle v

Initialize:

1: \( p, q, P, E, h_1, h_2, h_3, h_4, h_5 \) \( \triangleright \) Elliptic Curve
2: \( ID_v \) \( \triangleright \) Real ID
3: \( \{ sk_v, pk_v, cerCA(pk_v) \} \) \( \triangleright \) PKI
4: \( g_{sk_v}, g_{pk_{GM}} \) \( \triangleright \) GS
5: \( \{ SID_v \leftarrow (SID_v^{i,1}, SID_v^{i,2}), sid_v \} \) \( \triangleright \) IBC
6: \( T_{in} \)
7: \( T_{C_v in} \)
8: \( CMV = h_1(SID_v^{i,1}) \)
9: \( N_{cin}, A_v, C_o, N_{cin} = 1, CSin = 1 \)

Input: \( \xi_{N_v}, \xi_{A_v}, \xi_{C_o}, \xi_{N_L} \)

After Deployment:

10: Group Formation Phase as in [12]
11: \( CS = 2 \)
12: \( CMV = [h_1(SID_v^{i,1}), h_3(SID_v^{i,1})] \)
13: if \( (CS > 1 \land |CMV| \neq 1) \lor (CS = 1 \land |CMV| = 1) \) then \( \triangleright \) CL Vehicle
14: Initialize \( i \) \( \triangleright \) Number of Runs
15: Generate \( m_v, T_v \)
16: \( \text{Auth} (m_v, T_v) \leftarrow \text{PKI}(m_v, T_v) | (\text{Target} = \phi) \)
17: While \( i < \text{Run} \) do
18: \( T_{C_v} \leftarrow T_{C_v in} \)
19: While \( T_{C_v} > 0 \) do
20: \( N_C \leftarrow N_{cin} \)
21: \( N_L \leftarrow N_{cin} \)
22: \( A_v \leftarrow A_{vin} \)
23: \( C_o \leftarrow C_{o in} \)
24: For \( T \leftarrow 1, T_{in} \) do
25: Broadcast \( (m_v, T_v, \text{Auth}(m_v, T_v)) \)
26: Receive \( (m_w, T_w, \text{Auth}(m_w, T_w)) \)
27: If \( \text{Target} = \phi \) or \( \text{Target} = h_1(SID_v^1) \) then
28: Accept Beacon
29: \( N_C \leftarrow N_C + 1 \) \( \triangleright \) Total Number of Neighbors
30: Else
31: Reject beacon
32: end if
33: if \((CS_W > 1) \lor (CMV_W = h_1(SID_W))\) then
34: \(N_L \leftarrow N_L + 1\) \(\triangleright\) CL Beacon
35: Record \(X(N_L) = h_1(SID_W)\)
36: end if
37: \(A_V \leftarrow A_V + (pg^w_v \times \log(\frac{1}{pg^w_v}))\)
38: \(C_O \leftarrow C_O + Auth(m_w, T_w)\)
39: \(T_{C_v} \leftarrow T_{C_v} - 1\)
40: Generate \(m_v, T_v\)
41: end for
42: end while

\(\triangleright\) Start Context Checking

43: \(A_V \leftarrow \frac{A_V}{\log(N_C)}\)
44: if \(N_L > \xi_{N_L}\) then
45: if \(C_O > \xi_{C_O}\) then \(\triangleright\) Dense Network
46: \(Auth(m_v, T_v) \leftarrow IBC(m_v, T_v)\mid\phi\)
47: else if \(A_V < \xi_{A_V}\) then
48: \(Auth(m_v, T_v) \leftarrow GS(m_v, T_v)\mid CMS_V\) \(\triangleright\) To \(v\)'s CMs
49: \(Auth(m_v, T_v) \leftarrow IBC(m_v, T_v)\mid X\) \(\triangleright\) To CLs
50: else
51: \(Auth(m_v, T_v) \leftarrow PKI(m_v, T_v)\mid CMS_V\) \(\triangleright\) To \(v\)'s CMs
52: \(Auth(m_v, T_v) \leftarrow IBC(m_v, T_v)\mid X\) \(\triangleright\) To CLs
53: end if
54: else
55: if \(A_V \leq \xi_{A_V}\) then
56: \(Auth(m_v, T_v) \leftarrow GS(m_v, T_v)\mid\phi\)
57: else if \(C_O \geq \xi_{C_O}\) then
58: \(Auth(m_v, T_v) \leftarrow IBC(m_v, T_v)\mid\phi\)
59: else
60: \(Auth(m_v, T_v) \leftarrow PKI(m_v, T_v)\mid\phi\)
61: end if
62: end if
63: Record \(N_C, A_V, C_O, N_L\)
64: \(i \leftarrow i + 1\)
65: end while
66: end if
67: procedure \(PKI (m_v, T_v)\)
68: \(Auth (m_v, T_v)\) from (2.3)
69: return \(Auth (m_v, T_v)\)
70: end procedure
71: procedure \(GS (m_v, T_v)\)
72: \(Auth (m_v, T_v)\) from (2.4)
73: return \(Auth (m_v, T_v)\)

149
74: end procedure
75: procedure IBC \((m_v, T_v)\)
76: either \(p_{kibc} \leftarrow p_k v\)
77: or \(d \in \mathbb{Z}_q^*, p_{kibc} \leftarrow \hat{d}.p_k v\)
78: calculate \(\sigma_v\) from (7.6)
79: \(\text{Auth}(m_v, T_v) \leftarrow p_{kibc}, \text{SID}_v, \sigma_v\)
80: return \(\text{Auth}(m_v, T_v)\)
81: end procedure

Cluster dynamics and modifications including how vehicles join/leave the created clusters are given in Chapter 5. Having presented the two CAAIS scenarios of authentication interchange, we prove the security of our approach in what follows.

7.4 Security Analysis of CAAIS

This section first demonstrates the correctness of CAAIS. Next, a valid security proof is built to confirm the existential unforgeability of the protocol against an adaptive chosen message attacker. To derive a solid proof, as in [2,3,6,90,91], a definition of the attacker model is given. Also, the forking lemma and attack game strategies [91–93] are utilized. Finally, we exhibit the CAAIS resistance against several types of attacks and its satisfaction of VANETs security requirements.

7.4.1 CAAIS Correctness Proofs

The correctness of a cryptographic signature-based authentication protocol can be defined as “the correct verification at receivers of the generated signatures by sender \(v\) using the sender non-revoked validation keys. Since CAAIS represents an interchange between PKI, GS, and IBC authentication frameworks, its correctness is derived from their cryptographic correctness as:

\[
\text{PKI Correctness} \quad [p_{k_v} \notin CRL] \cap [\text{verify}_{pk_v}(\text{sig}_{sk_v}(h_1(m_v^i)|T_v^i)) = \text{valid}] \cap [h_1(m_v^i) = \text{Correct}] \tag{7.12}
\]

150
**GS Correctness**

\[
\text{verify}_{g_{skv}}(\text{sig}_{g_{skv}}(h_1(m^i)|T^i_v)) = \text{valid} \cap [h_1(m^i_v) = \text{Correct}] \cap [g_{skv} \notin RL] \tag{7.13}
\]

**IBC Correctness**

\[
[SID^i_v \notin RL] \cap [h_2(SID^i_v) = \text{Correct}] \cap [h_3(SID^i_{v,1}|T^i_v) = \text{Correct}] \cap [h_4(SID^i_{v,2}|T^i_v) = \text{Correct}] \cap [\text{verify}_{(P,pk_{TA/PKG})}(\sigma^i_v) = \text{valid}] \tag{7.14}
\]

### 7.4.2 Attacker Assumptions

Since our CAAIS alternates between three signature-based authentication frameworks, the attacker \(AT\) we consider is an Adaptive Chosen Message Attacker (CMA) as in Chapter 6. For the PKI, GS, and IBC schemes to be secure, they have to be existentially unforgeable against this attack [90]. Because ECDSA and GS signatures are secure primitives [1,99], we choose not to repeat their proof to mainly focus on examining the unforgeability of our new IBC signature-based authentication developed in Section 7.1. Our probabilistic \(AT\) attacker mounts an Existential Forgeability (EF) attack, within a bounded polynomial time \(\tau\), through playing a game against signer/challenger \(SC\) vehicle as in Fig. 7.3. In his game, \(AT\) obtains the public parameters \((p,q,P,pk_{TA/PKG}, h_1, h_2, h_3, h_4, h_5)\) of Table 7.2; \(AT\) will have the four \(h_1, h_2, h_3, h_4, \) and \(h_5\) hash oracles. He will ask \(SC\) to sign several \(m^i_{AT}\) via several signing queries; at each step \(i\), \(AT\) can analyze the previous message-signature pairs \((m^{i-1}_{AT}, \sigma^{i-1}_{AT})\) to adjust the next query \(m^{i+1}_{AT}\). Nevertheless, \(AT\) should not be able to
produce a valid signature $\sigma^k_{\mathcal{AT}}$ for a message $m^k_{\mathcal{AT}}$ that was not queried before, i.e., $k \notin \{i\}$; $Pr^{\text{EF-CMA}}_{\text{CAAIS}}(\mathcal{AT})$ is the probability of $\mathcal{AT}$ succeeds producing $\sigma^k_{\mathcal{AT}}$ [91,93].

7.4.3 CAAIS Security Against EF-CMA Attack

We assert that our new IBC authentication scheme is secure in the face of the EF-CMA attack and $\mathcal{AT}$ cannot succeed in existentially forging a signature $\sigma^k_{\mathcal{AT}}$ for message $m^k_{\mathcal{AT}} \notin \{i\}$ in a limited polynomial time period $\tau$. Since we are reducing the problem of forging a signature to another of solving the underlying ECDLP problem in the random oracle model [92,94], we want to prove that our CAAIS signature is secure if the underlying ECDLP is unsolvable.
Proof. For reduction, there must be a reduction-simulator \( RS \) entity that employs \( AT \) to solve the IBC’s ECDLP. \( RS \) and \( AT \) begin their collaboration when \( RS \) obtains the system parameters \((p, q, P, \hat{P}, h_1, h_2, h_3, h_4, h_5)\) and gives them to \( AT \). \( RS \) facilitates/simulates \( AT \)’s \( q_{h_1}, q_{h_2}, q_{h_3}, q_{h_4}, q_{h_5} \), and \( q_5 \) queries to the hash and sign oracles to find \( \hat{x} \in Z_q^* \) of \( \hat{P} = \hat{x}P \) in time period \( \tau \) with high \( Pr_{EF-CMA}^{CAIIS}(AT) \) probability. At the outset, \( RS \) holds a \((m_{index}, h_{index}^1(), h_{index}^2(), h_{index}^3(), h_{index}^4(), h_{index}^5(), \text{sign}_{index}())\) lookup table. If the received query found to be in the list, \( RS \) returns to \( AT \) the assigned value; otherwise, \( RS \) adds the new query and responds to \( AT \) with a random value as follows:

- Simulating \( h_1\)-Oracle as \( L_{h_1} = <\Omega^i_{index}, h_1^i_{index}>\): if \( RS \) receives \( AT \)’s \( i \)th query of \( \Omega^i_{AT} \), \( RS \) checks whether it exists in \( L_{h_1} \). If it is set, directly by a previous hash query, or by a signature query as will be shown, \( RS \) sends the corresponding \( h_1^i = h_1(\Omega^i_{AT}) \) to \( AT \); otherwise, \( RS \) generates a random \( h_1^i \in Z_q^* \), adds \( <\Omega^i_{AT}, h_1^i> \) to \( L_{h_1} \), and sends the generated \( h_1^i \) to \( AT \).

- Simulating \( h_2\)-Oracle as \( L_{h_2} = <SID^i_{index}, h_2^i_{index}>\): upon receiving \( AT \)’s \( SID^i_{AT} \) query, \( RS \) checks whether it exists in \( L_{h_2} \). If yes, \( RS \) sends the corresponding \( h_2^i = h_2(SID^i_{AT}) \) to \( AT \); otherwise, \( RS \) generates a random \( h_2^i \in Z_q^* \), adds \( <SID^i_{AT}, h_2^i> \) to \( L_{h_2} \), and sends \( h_2^i \) to \( AT \).

- Simulating \( h_3\)-Oracle as \( L_{h_3} = <SID^i_{index,1}, T^i_{index}, h_3^i_{index}>\): receiving \( AT \)’s query \( (SID^i_{AT}, T^i_{AT}) \), \( RS \) checks whether it exists in \( L_{h_3} \). If yes, \( RS \) sends the corresponding \( h_3^i = h_3(SID^i_{AT}, T^i_{AT}) \) to \( AT \); otherwise, \( RS \) generates a random \( h_3^i \in Z_q^* \), adds \( <SID^i_{AT}, T^i_{AT}, h_3^i> \) to \( L_{h_3} \), and sends \( h_3^i \) to \( AT \).
• Simulating $h_4$-Oracle as $L_{h_4} = \langle SID_{\ast}^{index}, T_{\ast}^{index}, h_4^{index} \rangle$: when $RS$ receives $AT$’s $(SID_{AT}^{i,2}, T_{AT}^{i})$, $RS$ checks whether it exists in $L_{h_4}$. If yes, $RS$ sends the corresponding $h_4^i = h_4(SID_{AT}^{i,2}, T_{AT}^{i})$ to $AT$; otherwise, $RS$ generates a random $h_4^i \in Z_q^*$, adds $<SID_{AT}^{i,2}, T_{AT}^{i}>$, $h_4^i >$ to $L_{h_4}$, and sends $h_4^i$ to $AT$.

• Simulating $h_5$-Oracle as $L_{h_5} = \langle pk_{ibc,R}, h_3^{index}, h_4^{index}, m_i^{index}, T_i^{index}, h_5^{index} \rangle$: when $RS$ receives $AT$’s $(pk_{ibc,R}^{i}, h_3^{i}(SID_{AT}^{i,1}, T_{AT}^{i}), h_4^{i}(SID_{AT}^{i,2}, T_{AT}^{i}), m_i^{AT}, T_i^{AT})$, $RS$ checks whether it exists in $L_{h_5}$. If yes, $RS$ sends the corresponding $h_5^i = h_5(pk_{ibc,R}^{i}, h_3^{i}(SID_{AT}^{i,1}, T_{AT}^{i}), h_4^{i}(SID_{AT}^{i,2}, T_{AT}^{i}), m_i^{AT}, T_i^{AT})$ to $AT$; otherwise, $RS$ generates a random $h_5^i \in Z_q^*$, adds $pk_{ibc,R}^{i}, h_3^{i}(SID_{AT}^{i,1}, T_{AT}^{i}), h_4^{i}(SID_{AT}^{i,2}, T_{AT}^{i}), m_i^{AT}, T_i^{AT}, h_5^i >$ to $L_{h_5}$, and sends $h_5^i$ to $AT$.

• Simulating Sign-Oracle: $AT$ sends a signing query to $RS$ for $<m_{AT}^{j}, T_{AT}^{j}>$. $RS$ checks the $(m_i^{index}, h_1^{index}(\cdot), h_3^{index}(\cdot), h_4^{index}(\cdot), h_5^{index}(\cdot), sign^{index}(\cdot))$ table to look $m_{AT}^{j}$ up. If it has been previously queried for, $RS$ returns the stored values of $<pk_{ibc,R}^{j}, SID_{RS}^{j}, \sigma_{RS}^{j}, m_{AT}^{j}, T_{AT}^{j}>$ to $AT$. Otherwise, $RS$ has to uniformly/randomly generate these values. First, $RS$ will choose three random point identities $ID_{RS}^{j}, SID_{RS}^{j,1}$, and $SID_{RS}^{j,2}$. With these identities and $T_{AT}^{j}$, $RS$ calculates $h_2^{j}, h_3^{j}$, and $h_4^{j}$, and populates their associated lists. Next, $RS$ generates a random signature $\sigma_{RS}^{j}$ and random hashes ($h_1^{j}$ and $h_2^{j}$), all in $\in Z_q^*$. The $L_{h_1}$ list will be populated by $RS$ with a random $\Omega_{RS}^{j}$ and the generated $h_1^{j}$. Having these parameters, $RS$ calculates $pk_{ibc,R}^{j} = h_3^{j}(\sigma_{RS}^{j} \cdot P - SID_{RS}^{j,1} - h_2^{j} \cdot \hat{P})$. With $pk_{ibc,R}^{j}, L_{h_5}$ is updated. Finally, $RS$ will send the signature elements $<pk_{ibc,R}^{j}, SID_{RS}^{j}, \sigma_{RS}^{j}, m_{AT}^{j}, T_{AT}^{j}>$ to $AT$ in response to his query on $<m_{AT}^{j}, T_{AT}^{j}>$. When the signature is received
by $\mathcal{AT}$, he verifies its validity (7.15) to be correct:

$$\sigma^j_{\mathcal{RS}}.P = SID^j_{\mathcal{RS}}h^j_5.p_{ibc_{\mathcal{RS}}} + h^j_2.\hat{P}, \quad (7.15)$$

where $h^j_2$ and $h^j_5$ are obtained by $\mathcal{AT}$ because of his ability to query $h_2$ and $h_5$ oracles for the received tuple elements from $\mathcal{RS}$.

With the answers from the hash and sign oracles in hand, $\mathcal{AT}$ outputs his (supposed) forgery on a random message $m^k_{\mathcal{AT}}$ that has not been queried to the sign oracle before. Hereabouts, $\mathcal{AT}$ can create his signature $\sigma^k_{\mathcal{AT}}$ in many ways as we explain in Chapter 6:

- **Plot 1**: $\mathcal{AT}$ might always be able to derive a valid signature $\sigma^k_{\mathcal{AT}}$ on $(m^k_{\mathcal{AT}}, T^k_{\mathcal{AT}})$. In his con, $\mathcal{AT}$ queries $h_2$, $h_3$, and $h_4$-oracles on a random $(ID^k_{\mathcal{AT}}, SID^k_{\mathcal{AT}}, SID^k_{\mathcal{AT}})$ trio to obtain $h^k_2$, $h^k_3$, and $h^k_4$ values. $\mathcal{AT}$ calculates $p_{ibc_{\mathcal{AT}}} = -SID^k_{\mathcal{AT}} - h^k_2.\hat{P} \in G$, from which he exerts a brute force attack against himself to guess $s_{ibc_{\mathcal{AT}}} \in Z^*_q$. $\mathcal{AT}$ queries $h_5$-oracle on $(p_{ibc_{\mathcal{AT}}}, h^k_3(SID^k_{\mathcal{AT}}, T^k_{\mathcal{AT}}), h^k_4(SID^k_{\mathcal{AT}}, T^k_{\mathcal{AT}}), m^k_{\mathcal{AT}}, T^k_{\mathcal{AT}})$ to obtain $h^k_5$. He computes his signature $\sigma^k_{\mathcal{AT}}$ as $(h^k_5 - 1).s_{ibc_{\mathcal{AT}}} \in Z^*_q$. Once the tuple $(p_{ibc_{\mathcal{AT}}}, SID^k_{\mathcal{AT}}, m^k_{\mathcal{AT}}, T^k_{\mathcal{AT}}, \sigma^k_{\mathcal{AT}})$ is received by $\mathcal{RS}$ as a signature on $(m^k_{\mathcal{AT}}, T^k_{\mathcal{AT}})$, verification of (7.15) might be valid.

- **Plot 2**: by querying the Sign-oracle of $\mathcal{RS}$, $\mathcal{AT}$ can emulate a similar pattern of signing to devise his signature. He starts by choosing all random values of $ID^k_{\mathcal{AT}}$, $SID^k_{\mathcal{AT}}, SID^k_{\mathcal{AT}}, \sigma^k_{\mathcal{AT}}, h^k_1$, and $h^k_5$. From these values, he queries the $h_2$, $h_3$, and $h_4$ oracles of $\mathcal{RS}$ and calculates $p_{ibc_{\mathcal{AT}}} = h^{-1}_5.(\sigma^k_{\mathcal{AT}}.P - SID^k_{\mathcal{AT}} - h^k_2.\hat{P})$. If updating $L_h$ with $< p_{ibc_{\mathcal{AT}}}, h^k_3, h^k_4, m^k_{\mathcal{AT}}, T^k_{\mathcal{AT}}, h^k_5 >$ is easy, the validation of the signature tuple at $\mathcal{RS}$ is correct as in (7.15).
• **Plot 3**: this track only reverses the last two steps of plot 2. Instead of determining \( p^{k}_{ibc,AT} \) at the end, \( p^{k}_{ibc,AT} \in G \) is selected before the \( h_{5} \) oracle is queried on \( < p^{k}_{ibc,AT}, h^{k}_{3}, h^{k}_{4}, m^{k}_{AT}, T^{k}_{AT} >. \) Having all ingredients in hand, the signature is calculated by \( AT \) to be sent to \( RS \) for verification.

When analyzing the probability of success in extracting \( x' \) across the three plots, we realize several findings. On one hand, in the first plot, \( AT \) himself can never extract \( s^{k}_{ibc,AT} \) with probability more than \( \frac{1}{P} \), i.e., \( \frac{1}{2^{2t}} \). Even if we assume \( AT \) is able to find \( s^{k}_{ibc,AT} \), \( RS \) can never find \( x' \) since the terms of (7.15) cancel each other. On the other hand, schemes 2 and 3 can use the forking lemma of Pointcheval and Stern [91, 92] to extract \( x' \). In order for \( RS \) to solve the ECDLP problem, he requires two forgeries from \( AT \): \( (p^{k}_{ibc,AT}, SID^{k}_{AT}, m^{k}_{AT}, T^{k}_{AT}, \sigma^{k}_{AT}) \) and \( (p^{k'}_{ibc,AT}, SID'^{k'}_{AT}, m^{k'}_{AT}, T'^{k'}_{AT}, \sigma'^{k'}_{AT}) \) for the same \( p^{k}_{ibc,AT} \) and \( m^{k}_{AT} \). The second forgery can be easily obtained when \( AT \) rewinds the call for \( h_{1}, h_{2}, h_{3}, h_{4}, h_{5}, \) and \( Sign \) oracles a polynomial \( (q_{h_{1}}, q_{h_{2}}, q_{h_{3}}, q_{h_{4}}, q_{h_{5}}, \) and \( q_{s} \)) number of times. Such call gives similar results until some moment \( i^{*} \in \{1..q_{h_{1,2,3,4,5}}\} \). From point \( i^{*} \), the results of the oracle queries will fork/differ from the previously obtained values. As a result, \( AT \) outputs such other valid signature tuple. In Fig. 7.4, a schematic view of our proof forking lemma is presented. In this case, the solution to the ECDLP, \( x' \), can be extracted as

\[
\dot{x} = \left[ \frac{(\sigma'^{k'}_{AT} - \sigma^{k}_{AT}).P - (SID'^{k'}_{AT} - SID^{k}_{AT}) - (h'^{k'}_{5} - h^{k}_{5}).p^{k}_{ibc,AT}}{(h'^{k'}_{2} - h^{k}_{2}).P} \right].
\] (7.16)

What we need to prove is that the probability of extracting \( x' \) by \( RS \) is negligible and much smaller than the probability of forging a signature \( Pr^{EF-CMA,CAAIS}_{\text{CAAIS}}(AT) \) by \( AT \). In other words, we want to prove that \( RS \) is a loose reduction simulator. For this goal,
Figure 7.4. RS Allows AT To Rewind RS’s Oracles To Solve ECDLP Problem and Extract x.
we need to estimate the reduction factor; because the point of forking $i^*$ can occur on any $h_1$, $h_2$, $h_3$, $h_4$, and $h_5$ query, we develop Table 7.4 to help estimating which oracle call causes the forking. Hereabouts, the probability of solving the ECDLP will be reduced by \( \frac{(q_h)^2 + 3q_h q_b + 4q_h + q_b + 2}{(q_h)^2 q_b} \) in addition to the signing oracle queries of \( \frac{1}{q_s} \) probability. The final probability of breaking the hardness of ECDLP will be

\[
\frac{q_s ((q_h)^2 + 3q_h q_b + 4q_h + q_b + 2)}{(q_h)^2 q_b} P_{\text{CAAIS}}^{\text{EF-CMA}} (AT).
\] (7.17)

Since the hash and signature values fall in \( Z_q^* \) and $G$ and since \( |q| = 2L \), the security reduction ensures that as long as $L$ is large, $AT$ can never forge any signature with a probability larger than \( \left( \frac{(4)^{2L} + (5)^{2L} + 3}{2^{2L}} \right). P_{\text{CAAIS}}^{\text{EF-CMA}} (AT) \approx \frac{P_{\text{CAAIS}}^{\text{EF-CMA}} (AT)}{O(2^{2L})} \). \)

**Table 7.4**  
Probability of Querying The RS Oracles To Solve ECDLP  
(F=FALSE and T=TRUE)

<table>
<thead>
<tr>
<th>Probability Tree</th>
<th>Summation of Forking Oracles Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://example.com/diagram.png" alt="Probability Tree Diagram" /></td>
<td>1. ( Pr(h_5</td>
</tr>
<tr>
<td></td>
<td>2. ( Pr(h_5</td>
</tr>
<tr>
<td></td>
<td>3. ( Pr(h_5</td>
</tr>
<tr>
<td></td>
<td>4. ( Pr(h_5</td>
</tr>
<tr>
<td></td>
<td>5. ( Pr(h_5</td>
</tr>
<tr>
<td></td>
<td>6. ( Pr(h_5) \cap Pr(h_2) = \left( \frac{1}{q_{h_5}} \right) \cdot \left( \frac{1}{q_{h_5}} \right) )</td>
</tr>
<tr>
<td></td>
<td>7. ( Pr(h_2) = \left( \frac{1}{q_{h_5}} \right) )</td>
</tr>
<tr>
<td></td>
<td>8. ( Pr(h_5</td>
</tr>
<tr>
<td></td>
<td>9. ( Pr(h_5</td>
</tr>
<tr>
<td></td>
<td>10. ( Pr(h_5</td>
</tr>
<tr>
<td></td>
<td>11. ( Pr(h_5) = \left( \frac{1}{q_{h_5}} \right) )</td>
</tr>
<tr>
<td></td>
<td>12. ( 0 = \frac{(q_h)^2 + 3q_h q_b + 4q_h + q_b + 2}{(q_h)^2 q_b} )</td>
</tr>
</tbody>
</table>

Our given IBC proof as well as the proven-security of ECDSA and GS affirm that CAAIS can withstand: replay attack because of the inclusion of timestamps $T'_v$, 158
man-in-the-middle and modification/impersonation/masquerading attacks because of using hashing functions whether \( h_1 \) in PKI, \( h_1 \) of GS, or \( h_1 \) through \( h_5 \) in the new IBC scheme. These traits prevent \( AT \) from playing any manipulation game against network legitimate entities.

7.4.4 CAAIS Fulfills Essential Security and Advanced Privacy Requirements of VANETs

The mixture of the three frameworks in our scheme helps satisfying the essential requirements of security in VANETs. First, CAAIS ensures the trustability feature where only legitimate users, who have \( pk_{CA} \) of PKI, \( g_{pk_M} \) of GS, and \( pk_{TA/PKG} \) of IBC, can verify the received beacons. Second, CAAIS affords the freshness and integrity of messages via timestamps \( T_i \) and hashes \( h_1 \) through \( h_5 \). Further, CAAIS guarantees non-repudiation of senders by associating the created signatures with the \( sk_v \), \( g_{sk_v} \), and \( sk_{ibc_v} \) private keys that are given at time of registration. Also, revocation of expired credentials of vehicles or of credentials of misbehaving vehicles is achieved in our approach. Since CAAIS interchanges between PKI, GS, and IBC authentication frameworks, CAAIS has to maintain the revocation techniques of these three schemes. CAAIS adopts CRL of PKI, RL of GS, and RL of IBC approaches. In this regard, our trusted authorities CA/GM/TA/PKG have to periodically convey a signed up-to-date revocation list mixture to the vehicles and to the infrastructure. The created mixed \((CRL + RL^2)\) list’s entries can take the form of \(< \{Cer_{CA}(pk_{iv})_{serial\#}\}, g_{sk_v}, \{SID_{iv}\}, \text{revocation\_date}, (CRL + RL^2)_{generation\_date} >\). To keep the vehicles and infrastructure aware of any slight modification in their surrounding, a delta-\((CRL + RL^2)\) can be broadcasted between any two \((CRL + RL^2)\) announced instances to prevent escalating
Aside from security, CAAIS is able to achieve the following more advanced privacy requirements:

- (Conditional Privacy $\iff$ Traceability): vehicles of VANETs have to be untraceable by the public. Still, the trusted CA/GM/TA/PKG authorities, with which vehicles have registered before deployment, have to identify their members for accountability considerations. Our CAAIS establishes the conditional privacy through the three used frameworks. The CA/GM/TA/PKG in CAAIS hosts a mixed $< ID_v, \{sk_v, pk_v, cer_{CA}(pk_v)\}, g_{sk_v}, \{d^i, SID_v, sid_v, ID_v^{track}\} >$ mapping tuple for every registered vehicle. If the signature of the broadcasted beacon is suspected, when the PKI framework is active, the trusted authority compares the $cer_{CA}(pk_v)$ portion of the broadcasted beacon of (2.3) with the stored tuple to locate $ID_v$. Similarly, if the GS is active, the trusted authority can verify the $\sigma_{sk_v}(h_1(m_v^i)|T_v^i)$ portion of (2.4) to compare the extracted $g_{sk_v}$ with the stored tuple to find $ID_v$. When our IBC authentication is running, the beacon authentication has the form of $< pk_{ibc_v}, SID_v, \sigma_v >$. From $SID_v$, the trusted authority can directly find $ID_v$. However, to make sure that this $SID_v$ has not been altered, the trusted authority calculates $ID_v^{track}$ as in (7.4) and compares the result with the stored tracking identity before accepting $ID_v$. Entities other than CA/GM/TA/PKG cannot retrieve $ID_v$ since it is hard to reverse $SID_v$ without having $sk_{TA}$.

- (Anonymity $\iff$ Unlinkability): the anonymity of signer identity is guaranteed in our scheme through the use of periodic changing of pseudonyms of PKI and super anonymous group signature of GS authentication. Besides, the use, of alias identity $SID_v$ and random public keys $pk_{ibc_v}$ of IBC framework, vehicle $v$ anonymity is guaranteed when IBC is running. By warranting the anonymity of sender, we
ensure that any receiver cannot link the broadcasted beacons to each other or as being coming from the same sender.

Having fully explained CAAIS and proved its security, the following section introduces our technical approach, through which we evaluate the effectiveness of the two CAAIS versions in terms of the considered performance metrics.

7.5 Numerical Analysis and Discussion of CAAIS Performance (Comparison of Effectiveness)

Conforming to the existing regulations of DSRC standards [7], we list in Table 7.5 the values that represent our CAAIS model fundamental settings. For the size of the cryptographic parameters, the up-to-date elliptic curves recommendations of \( L = 128 \) bit security level is utilized for the three authentication schemes in our CAAIS. As we use the ECDSA signature in the PKI authentication, we endorse the group signature of [1] to be the GS framework. Also, we use the recommended CSMA/CA settings for medium access of a \( W_0 = 32 \) contention window size, a \( \frac{W_0 - 1}{2} = 16 \) backoff counter, an \( \beta = 13 \mu \text{sec} \) backoff timeslot, and a DIFS interval of \( 98 \mu \text{sec} \). The adopted beacon generation rate \( \mathcal{T} \) is \( 100 \text{ msec} \) at \( 6 \text{ Mbps} \) data transmission rate. The CAAIS particular settings are chosen such that the \( \mathcal{T}_{cv} \) context timer value is four times the \( \mathcal{T} \) beacon generation rate. The signal strength \( p_{g_{iv}} \) values of the received beacons by vehicle \( v \) are normalized in our simulations to probability values within 0 and 1 interval depending on the distance between sender and receiver. Since the \( < \xi_{c}, \xi_{A_v}, \xi_{C_o} > \) threshold values help fine-tuning the simulation results, we will discuss their settings later. Given these parameters, we can estimate the upper-layer application behavior of CAAIS. We conduct our simulating analysis using the
MATLAB environment [81] over a personal Macbook Pro machine of a 2.9 GHz Intel Core i7 processor and an 8 Gbyte DDR3 memory. To achieve a fair evaluation of the proposed CAAIS’s effectiveness in the two targeted scenarios, we consider the following metrics:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$ Beacon Generation Period</td>
<td>100 msec</td>
</tr>
<tr>
<td>$T_{Cv}$ Context Timer</td>
<td>$4T$ msec</td>
</tr>
<tr>
<td>$W_0$ Contention Window</td>
<td>32</td>
</tr>
<tr>
<td>$\beta$ Backoff Timeslot</td>
<td>13 $\mu$sec</td>
</tr>
<tr>
<td>DIFS</td>
<td>98 $\mu$sec</td>
</tr>
<tr>
<td>$r$ Data Transmission Rate</td>
<td>6 Mbps</td>
</tr>
<tr>
<td>$L$ Security Strength</td>
<td>128 bit</td>
</tr>
<tr>
<td>$p$ Finite Field Prime Order</td>
<td>$2^{128}$ bit</td>
</tr>
<tr>
<td>$a$, $b$, $x$, $y$ Elliptic Curve Parameters</td>
<td>$\in F_{2^{128}}$</td>
</tr>
<tr>
<td>$#E$ Number of $E$’s Points</td>
<td>$(2^{128} + 1 \pm 2\sqrt{2^{128}})$</td>
</tr>
<tr>
<td>$q$ Order of an $E$’s Point $</td>
<td>q</td>
</tr>
<tr>
<td>$t$ Security Parameter of The Small Exponent Test</td>
<td>5 bit</td>
</tr>
<tr>
<td>Scalars</td>
<td>$\in Z_{2^{256}}$ bit</td>
</tr>
<tr>
<td>SHA-256 Hash</td>
<td>$2^{256}$ bit</td>
</tr>
</tbody>
</table>

### 7.5.1 Computation Complexity (bit)

$C_C$ pertains to the number of conducted operations during the CAAIS signing and verifying phases. We connect the $C_C$ value with the number of bits of the domain parameters, namely the $L$ security strength of the scheme via the big $O$ notation; Table 3.2, 4.1, and 6.5 highlight the exact complexity of each involved cryptographic primitives. Since our scheme interchanges between PKI, GS, and IBC frameworks (Algorithm 1 and 2), CAAIS also interchanges between their $C_C$s shown in Table 7.6. From a coarse-grained stance, the three frameworks incur a comparable complexity of $O(L^3)$ because they use complex elliptic curves operations such as scalar multiplication, modular exponentiation, and Weil pairings. In this sense, CAAIS requires $C_C = O(L^3)$ in both of its sender-signing and receiver-verifying phases. However,
with a close fine-grained viewpoint, IBC shows fastest signing operation because of the inclusion of modular addition of \( \mathcal{O}(L) \) in comparison to the modular inversion of \( \mathcal{O}(L^2) \) in PKI, GS is the slowest for including eight modular exponentiation of \( \mathcal{O}(L^3) \) in comparison to only one scalar multiplication in PKI and IBC at the same order of complexity. At the receiving end, verification of IBC overthrows both PKI and GS for having only three scalar multiplications in comparison to six in PKI and seven approximate values in GS. In conclusion, when CAAIS uses the GS authentication, it takes longer time to generate and verify signatures at vehicles, but as we choose tighter threshold \( \xi_{N_C} \) on the allowed number of vehicles in the network, this fact is overlooked. For dense network status, CAAIS needs the fast signing and verifying IBC framework. We can also use the batch IBC verification to minimize the complexity by a factor of \( t \).

<table>
<thead>
<tr>
<th>Framework</th>
<th>Sign</th>
<th>Single Verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI</td>
<td>((1)SM + (1)MI + (1)MM + (1)HS)</td>
<td>((6)SM + (1)PA + (2)HS)</td>
</tr>
<tr>
<td>GS [1]</td>
<td>((2)WP + (8)ME)</td>
<td>((2 + 2</td>
</tr>
<tr>
<td>IBC</td>
<td>((1)SM + (1)MM + (1)MA + (3)HS)</td>
<td>((3)SM + (2)PA + (4)HS)</td>
</tr>
<tr>
<td><strong>IBC Batch Verification</strong></td>
<td>((N_C + 2)SM + (3N_C)SMM + (2N_C − 2)MA + (N_C)SSM + (2N_C)PA + (4N_C)HS)</td>
<td></td>
</tr>
</tbody>
</table>

### 7.5.2 Storage Overhead (byte)

CAAIS requires each vehicle to store couple long-term and short-term elements of the three authentication schemes. The long-term elements are fixed values and would not need to be updated or modified; however, short-lifetime elements have to be provided to vehicles in abundance; we refer to the number of elements that is
needed to be given to each vehicle $v$ as $N$. We can easily estimate the value of $N$ such that for a 1 minute [22] average lifetime of short-term elements and an average daily usage of vehicle to be 10 hours, $N$ will be $\frac{1 \text{ element}}{60 \text{ sec}} \times \frac{3600 \text{ sec}}{1 \text{ hour}} \times \frac{10 \text{ hour}}{1 \text{ day}} \times \frac{365 \text{ day}}{1 \text{ year}} = 219,000$.

Having $N$ and the relevant data of PKI and GS extracted from [1, 19, 39, 82–84], we show in Table 7.7 the predicted yearly Storage Overhead $S_O$ of our CAAIS scheme; CAAIS needs a 49 Mbyte per year. In comparison, PKI needs 28 Mbyte, GS requires 864 byte, and IBC necessitates 21 Mbyte, respectively. The sufficient storage resources at VANETs vehicles can effortlessly manage the 49 Mbyte CAAIS storage per year.

<table>
<thead>
<tr>
<th>Table 7.7</th>
<th>Authentication Schemes’ Yearly Storage Overhead $S_O$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>Overhead (byte)</td>
</tr>
<tr>
<td>Real Identity</td>
<td>$ID_v$</td>
</tr>
<tr>
<td>PKI</td>
<td>$sk_v^i$</td>
</tr>
<tr>
<td></td>
<td>$pk_v^i$</td>
</tr>
<tr>
<td></td>
<td>$pk_{CA}$</td>
</tr>
<tr>
<td></td>
<td>$\text{sig}<em>{sk</em>{CA}}(pk_v^i)$</td>
</tr>
<tr>
<td>GS [1]</td>
<td>$gp_{GM}$</td>
</tr>
<tr>
<td></td>
<td>$g_{sk_v}$</td>
</tr>
<tr>
<td>IBC</td>
<td>$pk_{TA/PKG}$</td>
</tr>
<tr>
<td></td>
<td>$sid_v^i$</td>
</tr>
<tr>
<td></td>
<td>$SID_v^{i,1}$</td>
</tr>
<tr>
<td></td>
<td>$SID_v^{i,2}$</td>
</tr>
</tbody>
</table>

$S_O = 224 \times N + 960$

### 7.5.3 Vehicle Drop Rate (vehicle)

To properly derive $\gamma_C$, we use the four performance metrics indicated in Section 7.2:

- Number of Neighbors $N_C$ (vehicle), Tolerated Communication Overhead $C_O$ (byte),
- Obtained Anonymity Degree $A_V$, and Number of $CL$ Leaders $N_L$.

Besides, an $N_M$ Number of $CM$ Members term, a $\gamma_C(\text{CL})$ leader drop rate, and $\gamma_C(\text{CL})$ member drop rate. Therefore, we first redefine and set their initial and threshold values such that:
1. **Tolerated Communication Overhead $C_O$ (byte) and Number of Neighbors $N_C$ (vehicle):** whether in its individual scenario or clustered scenario, CAAIS alternates between three frameworks, each has different authentication format and size (Table 7.8) to secure the beaconed message $(m^i_v, T^i_v)$; PKI sends two signatures and one public key, GS sends one group signature, and IBC communicates one signature, one public key, and two alias identities. The aforementioned security payload restricts the number of vehicles $N_C$ that can acquire the limited 46 msec wireless control channel [7]. In addition, the CSMA/CA backoff mechanism, with the considered settings of Table 7.5, will further bound the number of vehicles in the network. Fig. 7.5 shows the maximum number of individual vehicles $N_C$ that rendezvous on the wireless medium in the three frameworks for different $r$ Mbps data rates. Since IBC has the lowest $C_O$ load (Table 7.8), it accommodates more vehicles.

<table>
<thead>
<tr>
<th>Framework</th>
<th>Header and Message $(m^i_v, T^i_v)$ (byte)</th>
<th>Private Key (byte)</th>
<th>Public Key (byte)</th>
<th>Signature (byte)</th>
<th>Authentication Format and $C_O$ (byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI</td>
<td>150</td>
<td>32</td>
<td>32</td>
<td>64</td>
<td>$(sig_{sk_v}(h_1(m^i_v)</td>
</tr>
<tr>
<td>GS [1]</td>
<td>150</td>
<td>64</td>
<td>800</td>
<td>225</td>
<td>$sig_{sk_v}(h_1(m^i_v)</td>
</tr>
<tr>
<td>IBC</td>
<td>150</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>$(pk^i_{ibc_v}, SID^i_v, \sigma^i_v)$ 128</td>
</tr>
<tr>
<td>CAAIS Individual Scenario</td>
<td>150</td>
<td></td>
<td></td>
<td></td>
<td>Algorithm 1</td>
</tr>
<tr>
<td>CAAIS Clustering Scenario (CM Member)</td>
<td>$150 +</td>
<td>h_1(SID^i_{CM})</td>
<td>= 182$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAAIS Clustering Scenario (CL Leader)</td>
<td>$150 + (CS_{CL} - 1)h_1(SID^i_{CM}) = 150 + 32 (CS_{CL} - 1)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.8

**Keys and Signatures Lengths and Size of Authentication in PKI, GS [1], and Our IBC Authentication (Security Strength $L = 128$ bit)**
These maximum $N_C$ values facilitate for us the choice of $\xi_{N_C}$ and $\xi_{CO}$ thresholds. For simulation wealth, we choose $\xi_{N_C} = [N_{C\text{mean}}, N_{C\text{mean}}/2, N_{C\text{mean}}/4]$ and $\xi_{CO} = \xi_{N_C} \cdot \text{average}(C_{OIBC}, C_{OPKI}, C_{OGS})$, where $N_{C\text{mean}}$ is the average of $N_C$ in each authentication framework.

2. **Anonymity $A_v$ and $\xi_{A_v}$ Threshold**: the entropy of the $pg^i_v$ signal strength of the received beacons from vehicle $i$ contributes to the degree of anonymity of vehicle $v$ within the $T_{CV}$ context timer interval as stated in (7.9). The initial value of $A_v$ is obtained after a single beacon interval of 100 msec. For the anonymity threshold, our simulations coin it to be $\xi_{A_v} = [4, 3, 1.4]$ when $\xi_{N_C}$ is reached.

We want to ensure that these chosen threshold values are suitable for our scenario, i.e., comparable $N_C$, $C_O$, and $A_v$ results have to be achieved. In this sense, in Fig. 7.6, we compare CAAIS with the three authentications for about 20 runs of $T_{CV}$ context timer intervals. Hereabouts, we can easily see that in its individual case, CAAIS
achieves comparable anonymity level and number of neighbors performance for all \( \xi_{N_C} \) values. Moreover, for the overall beacon overhead \( C_O \), CAAIS introduces the same small overhead of our new IBC scheme; this makes it more effective than the other two pure authentications: PKI and GS. For example, CAAIS has on average 6\%, 3.4\%, and 8.76\% less overhead in comparison with pure PKI framework for the three values of \( \xi_{N_C} \) threshold, respectively. Compared with GS, CAAIS’s overhead is 22.69\%, 20.1\%, and 24.9\% less than GS overhead for all of the \( \xi_{N_C} \) threshold choices.

Having chosen such thresholds, we are led automatically to have less drop rate \( \gamma_C \) in comparison with PKI, IBC, and GS pure authentications. We define the drop rate here to be the total number of vehicles that cannot capture the wireless medium due to the limited CCH width and total communicated beacon size. Fig. 7.7 aids our claim where CAAIS outperforms the other authentications for the selected threshold values, especially when \( \xi_{N_C} = N_C/4 \); for example, although CAAIS’s \( \gamma_C \) approximates the drop rates of the pure authentications for \( \xi_{N_C} = N_{Cmean} \), CAAIS has an average of 43.4\%, 37.8\%, and 26.6\% gain in comparison with IBC, PKI, and GS frameworks when \( \xi_{N_C} = N_{Cmean}/2 \). For \( \xi_{N_C} = N_{Cmean}/4 \), CAAIS has an 80.3\% gain over IBC, a 78.55\% gain over PKI, and 74.3\% gain over GS framework.

After showing the effectiveness of CAAIS in an individual outline (blue cells in Table 7.8), we want to exhibit its performance when clustering is considered (yellow cells in Table 7.8). As the vehicles are being clustered (Section 7.3.2), we can estimate the maximum number of \( CL \) leaders and \( CM \) members that can communicate on the wireless channel. Since the settings of cluster size, beacon size, data rate, and channel width affect such maximum, in Fig. 7.8, we show the expected maximum number of \( N_L \) and \( N_M \) vehicles for cluster size \( CS = 4 \); We assume \( CL \) beacons to
Figure 7.6. \(N_C\), \(A_V\), and \(C_O\) Metrics Comparison for Individual CAAIS Scenario, IBC, PKI, and GS Frameworks

(a) \(N_C\) for \(\xi_{N_C} = N_{C_{\text{mean}}}\)

(b) \(A_V\) for \(\xi_{N_C} = N_{C_{\text{mean}}}/2\)

(c) \(C_O\) for \(\xi_{N_C} = N_{C_{\text{mean}}}/4\)
Figure 7.7. Drop γ_C Metric Comparison for Individual CAAIS Scenario, IBC, PKI, and GS Frameworks
have higher priority. At the outset, we aim to show that the clustered CAAIS does not decrease the number of vehicles who can capture the channel whether they are leaders or members and that CAAIS affords a comparable anonymity level and \( C_\Omega \) overhead. This prospect is attested to in Fig. 7.9 where a simple comparison of \( N_C, N_L, N_M, \) and \( A_V \) for \( \xi_{N_C} = N_{C_{\text{mean}}} \) is given. Furthermore, although that the \( C_\Omega \) beacon load of our CAAIS is comparable to PKI and IBC framework, it is about 12.3\%, 4\%, and 4\% less than the pure GS framework in the clustered scenario for the three values of \( \xi_{N_C} \) threshold. The main goal behind such preliminary results is to have a better drop rate. Since our scenario consists of leaders and members, our drop rate metric can be split into two sub-metrics: drop rate \( \gamma_C(CL) \) of leaders and \( \gamma_C(CM) \) of member vehicles. Our results in Fig. 7.10 show that when the CAAIS interchanges authentications, it actually saves vehicles from being unable to capture CCH. For example, CAAIS saves an average of 48.6\%, 32.2\%, and 26.5\% vehicles in general over IBC, PKI, and GS. The case of \( \xi_{N_C} = N_{C_{\text{mean}}}/4 \) exhibits the least drop rate outcome for CAAIS; in terms of overall vehicles drop rate, CAAIS achieves 68.3\%, 65.7\%, and 57.9\% gain over IBC, PKI, and GS authentications. For a fine-grained comparison: CAAIS achieves 82.4\%, 82.4\%, and 77.7\% less \( \gamma_C(CL) \) leader drop rate as well as 71.8\%, 68.9\%, and 63\% less member drop rate than that of pure IBC, PKI, and GS frameworks as is shown in Fig. 7.10c, 7.10f, and 7.10i.
Figure 7.8. Maximum $N_L$ and $N_M$ at Clustering of $CS = 4$
Figure 7.9. $N_C$, $N_L$, $N_M$, $A_V$, and $C_O$ Metrics Comparison for Clustered CAAIS Scenario, IBC, PKI, and GS Frameworks and $CS = 4$
Figure 7.10. $\gamma_C$, $\gamma_C(CL)$, and $\gamma_C(CM)$ Metrics Comparison for Clustered CAAIS Scenario, IBC, PKI, and GS Frameworks and $CS = 4$
Chapter 8: Conclusions and Future Work

The unique set of challenges in VANETs offers vast research venues. In this dissertation, we chose three major challenges and suggested some research directions that may help in overcoming these limitations. We have successfully shown in this dissertation how these problems can be addressed in the light of utilizing different concepts to afford the security, privacy, and availability of VANETs communications:

1. Concealing sensitive vehicular transactions is crucial; encryption has been considered the typical tool to achieve secrecy. Few efforts have focused on this topic beside the IEEE 1609.2’s ECIES recommendation. ECIES cannot operate without a preliminary step of authentication; therefore, we have been motivated to examine the suitability of new low-level state-of-the-art partial homomorphic encryption concept in building a comparable alternative to ECIES that affords dynamic one to one authentication. Our proposed construction creates encrypted keys and identities without openly exchanging preliminary information. With an acceptable performance in the context of two targeted scenarios, the obtained analytical results in terms of storage and computation time showed the feasibility of our model.

2. Location privacy preservation in vehicular networks is important. On one hand, under the PKI realm, changing pseudonyms has to be effective, i.e., the change must
occur while the vehicle is within a fixed or a dynamic mix zone that mainly involves encryption or silent periods. On the other hand, under the GS authentication realm, super anonymity is afforded for vehicles but at the expense of slow signature’s verification. Therefore, to benefit from the super anonymity of GS authentication, we suggested the idea of altering the authentication to form a dynamic mix crowd for an originating vehicle that needs to change its pseudonym. Since vehicles within the group are relatively static to each other, the slow performance of group signature would not affect the intra-communication within the zone. Our analysis showed the easiness of forming and demolishing the dynamic group as well as proved the validity of the model by satisfying the needed vehicular security requirements. Also, our proposal gave an acceptable computation and storage overhead even with the use of GS and hybrid authentication since it limited their usage to be throughout the group lifetime only.

3. To improve the performance of vehicular safety communication, we introduced a scheme that helps reduce the contention for the wireless medium and lowers the communicated authentication overhead without sacrificing the required security level. In our G-BEE framework, groups were formed in a fully-distributed fashion with memberships and leaderships decided and updated depending on the channel estimates and signal strengths among vehicles. By dedicating the V2I communication to the group leaders only, we transformed the brief-sessions of the high participant-density setting in the existing approaches, into long-sessions with only a few active vehicles. Without downgrading the security level, two versions of G-BEE were introduced to reduce the communicated overhead of authentication in beacons. We analyzed the gain of our G-BEE schemes via our own built Bianchi, D/M/1, and
D/M/1/1 mathematical models. The numerical evaluations showed the advantage of using G-BEE. Both adaptations performed better than the no grouping case in terms of the designated performance metrics. The enhanced version increased the maximum number of vehicles supportable by 40%. Moreover, a 92% lower beacon loss rate and a 33% higher beacon delivery rate were achieved.

4. With the goal of finding an efficient and privacy preserving identity based alternative authentication for VANETs, we built a pairingless IBC authentication with fast computations, smaller communication overhead, and more anonymous usage of pseudo identities. Our PME-CPPAS scheme does not rely on the certificates of PKI nor on the bilinear pairing of IBC. It further eliminates the need for expensive TPDs. In comparison with the available literature, our analysis showed that PME-CPPAS performs well and gave comparable results in both safety message signing and single verification phases of the authentication. In its batch verification phase, there is still a room for future improvements to achieve better results.

5. In VANETs, the number of vehicles, which can concurrently capture the wireless medium, is limited by the width of the CCH channel and by the type of the utilized authentication framework; some frameworks have less communication overhead while others have more anonymous traits; therefore, sticking to only one framework is unfair. In this sense, we introduced in this work a context-aware authentication interchange scheme that mixes/interchanges between three cryptographic frameworks. In our scheme, each PKI vehicle checks its surrounding every a specific time interval: if the vehicle finds that the neighborhood is crowded, the car switches to a lower overhead IBC authentication; if the surrounding is sparse, the vehicle
switches to a more anonymous GS authentication. Our mingled-authentications scheme showed a better performance than the case of using single pure authentication to secure the network throughout the network lifetime. In terms of six chosen performance metrics, our proposed CAAIS is effective; it is computably fast and preserves the required vehicle’s anonymity level. CAAIS suits both individual and clustered VANETs scenarios where it does not add overhead to the communication; CAAIS achieved less vehicle drop rate in comparison with other three IBC schemes.

Through our contributions, we want to shed lights on some ongoing issues in vehicular communications. Whether the goal is to have stronger security, more privacy, or better efficiency, instead of sticking to the adopted DSRC standards, we advocated an unbiased research philosophy that encourages incorporating new frameworks and concepts. Our future research will continue focusing on investigating, analyzing, and characterizing of cohesive solutions that befit VANETs requirements without escalating existing nuisances.
Appendix A: Cryptographic Concepts

Cryptography represents one of two major fields that have been utilized for achieving security and privacy purposes in communications and systems; unlike information-theory based methods, cryptography’s hardness comes from the computational complexity of the math structures used to build such cryptographic algorithms [20, 21]. This appendix would facilitate the understanding of the cryptographic concepts in Fig. A.1:

A.1 Cryptosystems Types

Until 1976, all designed cryptographic algorithms basically dealt with encrypting and decrypting data transferred between a pair or group of users using a similar shared
secret key. Being a block cipher, a stream cipher, or a hash function, symmetric algorithms are fast, require less processing, and assure privacy since there is no need for key exchange. However, because symmetric algorithms need key management procedures and that their keys’ length is critical for security, asymmetric algorithms were proposed in 1976 by Whitfield Diffie, Martin Hellman and Ralph Merkle. In the public key cryptography, each party would have two different keys such that one of them is kept secret while the other key can be publicly shared without affecting the security of the system. Since this type of cryptography is slow, requires long keys to preserve the desired level of security, and needs a preliminary step of announcing the keys, an idea of mixing the benefits of the two cryptosystems into one became necessary to boost the strengths and to eliminate the weaknesses [20,21].

A.2 Mathematics Behind Cryptography

Cryptosystems are designed in a way to be infeasible in practice to be broken by any attacker even though in theory it might be possible to crack such systems down. Prior to 1985, number theory concepts such as integer factorization, primes, and DLP were the basis on which most systems being built; keys in these systems were merely integers. However, to enhance the public key cryptography in face of ongoing powerful attacks, cryptographers have focused on boosting its robustness through utilizing mathematical structures that are more complex than the traditional algebraic mathematical groups of finite fields; in 1985, elliptic curves were employed in cryptography and later on in 1997 lattices became the new recruited soldier [20,21]. Since Elliptic Curve Cryptography ECC is utilized in the DSRC’s 1609.2 vehicular standard, we will dedicate the next subsection to explain its terminology.
A.2.1 Elliptic Curves Cryptography

Elliptic curves play a leading role in cryptography. A formal definition to an elliptic curve can be simply stated as:

**Definition A.1** (Elliptic Curve $E$ [20, 21]). The elliptic curve $E$ over $F_p$ (Galois field modulus large prime $p > 3$) is the set of all points $(x, y) \in F_p$ which fulfill the short Weierstrass equation $y^2 \equiv (x^3 + ax + b) \mod p$ together with an imaginary point at infinity $\mathcal{O}$, where $a, b \in F_p$ and the condition $4a^3 + 27b^2 \neq 0 \mod p$ must be satisfied to have no self-intersected smooth curves. Number of points on elliptic curve is bounded by Hasse’s theorem to $(p + 1 - t)$ where $t$, the Trace of Frobenius, satisfies $|t| \leq 2\sqrt{p}$.

There are several types of elliptic curves such as supersingular curves and Miyaji-Nakahayashi-Takano (MNT) curves besides the random curves. The supersingular curves, with $t$ equals multiple of $p$, have fast operation of multiplying a point by scalar and they offer symmetric pairing as well see below; this makes such curves more appealing in research. For example, $y^2 \equiv x^3 + x \mod (p \equiv 3 \mod 4)$ and $y^2 \equiv x^3 + 1 \mod (p \equiv 2 \mod 3)$ are common curves in cryptography. MNT curves on the other hand are non-supersingular curves, but they offer the asymmetric pairings operation and they come with a high embedding factor. In all types, the power of cryptography depends on the Elliptic Curves DLP (ECDLP) hardness that is stated as:

**Definition A.2** (ECDLP Problem [20, 21]). For any two points $\hat{P}$ and $\hat{Q}$ on the elliptic curve, it is computationally infeasible to know whether they are multiples of each other, i.e., it is hard to know $k$ in $\hat{Q} = k.\hat{P}$. 

180
A.2.2 Pairings on Elliptic Curves

In spite of the ECC strength, the availability of pairings operation on elliptic curves reduces the hardness of the ECDLP problem to a DLP over finite fields. Originating from such cryptanalysis purpose, pairings [100] has been suggested for building new cryptography concepts such as homomorphic encryption, identity based encryption, tripartite key agreement, and others. Pairings in general can be defined as:

**Definition A.3 (Pairings [100]).** Pairings is a function \( e \) that maps between elements of three cyclic groups \( G_1, G_2, \) and \( G_T \) of the same order. When two elements from two different additive groups \( G_1 \) and \( G_2 \) are mapped into the third multiplicative one \( G_T \), asymmetric pairings occurs \( e : G_1 \times G_2 \rightarrow G_T \). Otherwise if the two elements come from same additive group \( G \), it is called symmetric \( e : G \times G \rightarrow G_T \). Given \( a, b \in \mathbb{F}_p \) and \( \hat{P}, \hat{Q}, g \in G \), pairing has three fascinating properties:

- **Bilinearity:**
  - Additive form: \( e(a \hat{P}, b \hat{Q}) \rightarrow e(\hat{P}, \hat{Q})^{ab} = e(b \hat{P}, a \hat{Q}) = e(ab \hat{P}, \hat{Q}) = e(\hat{P}, ab \hat{Q}) \),
  - or
  - Multiplicative form: \( e(g^a, g^b) \rightarrow e(g, g)^{ab} = e(g^b, g^a) = e(g^{ab}, g) = e(g, g^{ab}) \).

- **Non-degeneracy:** Given \( \hat{P}, g \neq 0 \) then \( e(\hat{P}, \hat{P}) \neq 1 \leftrightarrow e(g, g) \neq 1 \).

- **Computability:** pairing function \( e \) must be efficiently computable.

The two most popular pairings/mappings are Tate’s and Weil’s. They can be used in pairing-based cryptography without the need to know their underlying complicated math [100, 101].
A.3 Cryptographic Primitives

Basically, any cryptosystem or crypto protocol when being built, to achieve confidentiality, integrity, authentication, etc., has to employ a combination of these three underlying building blocks or primitives: encryption algorithms, signature algorithms, and key exchange protocols [20, 21]. By limiting our explanation to the primitives adopted by the WAVE 1609.2 standard in VANETs, we see that for (a) confidentiality purposes, the symmetric Advanced Encryption Standard (AES) block cipher with 128 bit keys (in a CCM mode of authenticate with MAC then encrypt) and the hybrid elliptic curves ECIES encryption scheme are used, (b) authentication services, the ECDSA signature with 192 bit and 256 bit keys is adopted, and (c) the integrity of data, the SHA-2 family of hash functions is recommended specifically SHA-256 that produces 256 bit message digests [7, 14]. The key establishment protocol that is used in this document is the Blom key sharing protocol as explained below.

A.3.1 Blom Key Establishment Protocol

In [79], Blom presented a pioneering scheme for sharing a pairwise secret keys. In such scheme, with the help of an offline authority, any two nodes can establish a shared secret making use of two keying matrices:

- \( \Psi \) is a public matrix with elements generated by generator \( \psi \) from a Galois Field \( F_{p_{Blom}} \) modulus large prime \( p_{Blom} \):

\[
\Psi = \begin{bmatrix}
(\psi^1)^0 & (\psi^2)^0 & \cdots & (\psi^n)^0 \\
(\psi^1)^1 & (\psi^2)^1 & \cdots & (\psi^n)^1 \\
\vdots & \vdots & \ddots & \vdots \\
(\psi^1)^\lambda & (\psi^2)^\lambda & \cdots & (\psi^n)^\lambda 
\end{bmatrix}_{(\lambda+1) \times n}
\] (A.1)
• Υ is a symmetric secret matrix of random entries $\bar{r}_{i,j} \in F_{p^{Blom}}$:

$$
\Upsilon = \begin{bmatrix}
\bar{r}_{0,0} & \bar{r}_{0,1} & \cdots & \bar{r}_{0,\lambda} \\
\bar{r}_{0,1} & \bar{r}_{1,1} & \cdots & \bar{r}_{1,\lambda} \\
\vdots & \vdots & \ddots & \vdots \\
\bar{r}_{0,\lambda} & \bar{r}_{1,\lambda} & \cdots & \bar{r}_{\lambda,\lambda}
\end{bmatrix}_{(\lambda+1) \times (\lambda+1)}
$$

(A.2)

where $n$ is the network size and $\lambda$ is the maximum number of members to be compromised without causing the system’s collapse. Having $\Psi$ and $\Upsilon$, the online phase of the scheme is established as follows:

1. The offline third party provides each node with $\psi, \lambda, n$ to allow them to generate their related columns of $\Psi$.

2. The offline third party publishes to each node a related row from $(\Upsilon.\Psi)^{Tran}$; yet, it keeps $\Upsilon$ secret.

3. Whenever nodes are interested in communicating, they swap their identities to determine the shared secret key. Blom used the properties of matrix transpose to prove that the generated keys match [79]:

**Proof A.1** (Matrix Transpose Similarity [79]).

$$
Key \text{ Matrix} = (\Upsilon.\Psi)^{Tran} \cdot \Psi
$$

$$
= \Psi^{Tran} \cdot \Upsilon^{Tran} \cdot \Psi
$$

$$
= \Psi^{Tran} \cdot \Upsilon^{Tran} \cdot \Psi \ (\Upsilon \text{ is symmetric})
$$

$$
= \Psi^{Tran} \cdot ((\Upsilon.\Psi)^{Tran})^{Tran}
$$

$$
= ((\Upsilon.\Psi)^{Tran} \cdot \Psi)^{Tran}
$$

$$
=(Key \text{ Matrix})^{Tran}
$$

Because the nodes’ identities are plainly exchanged, privacy of communication is not preserved. Furthermore, when more than $(\lambda + 1)$ nodes collaborate, they can
obtain the shared keys, therefore it is referred to as $\lambda$-secure; the larger the $\lambda$ is the more resilient the network against outsiders is [79].

### A.4 Homomorphism

When any sender encrypts data, the receiver has to decrypt it to compute over the plaintext and at this point attackers can compromise the decrypted data as shown in Fig. A.2 where $n$ represents the number of $m_i$ messages and ciphertexts $c_i$, $enc/dec$ are encryption/decryption functions, $sk_{BGN}/pk_{BGN}$ are the private/public keys, and $f$ is a function such as addition $+$ or multiplication $\times$. Therefore, researchers have always attempted to Homomorphic Encrypt (HE) the data, i.e., conduct computations of function $f$ on the ciphertexts to obtain ciphered results which when unfolded match the outcome of operations performed on the plaintexts. The nature of $f$ decides the degree of homomorphism such that HE has a single $+$ or $\times$ operation, Somewhat HE (SHE) refers to limited number of $+$ and $\times$ operations in $f$, and when $f$ embodies an arbitrary number of additions and multiplications the term Fully HE (FHE) is used. Since the increase of the homomorphism degree escalates the scheme complexity, slows down its performance, and leads to an implementation impracticality, recent research focuses on methods that attempt to improve upon what’s available; such schemes are called Efficient FHE (EFHE). Despite their promising future, all homomorphic schemes have the undesired property of malleability where active attackers can convert a ciphertext into another whose decryption gives a plaintext related to the original; therefore, they must have computationally infeasible design [102–104].
A.4.1 BGN Cryptosystem

As can be seen in Fig. A.3, in 2005 Boneh et al. [10] presented an efficient and practical SHE cryptosystem BGN to evaluate dot products on ciphertexts providing that the inputs come from a small set. They utilized two algebraic concepts: (a) finite composite-order subgroup \( G_1 \) from a supersingular elliptic curve over prime field \( F_{p_{BGN}} \) and (b) modified Weil pairings between \( G_1 \) and \( G_2 \) (another subgroup from an extension field \( F_{p_{BGN}^e} \) related to \( F_{p_{BGN}} \)) in their system to build three algorithms:

1. Key Generation: From a security parameter \( L \), two public \( pk_{BGN} = (\bar{n}, G_1, P_{BGN}, \hat{P}_{BGN}, e, G_2) \) and secret \( sk_{BGN} = q_1 \) keys are generated where:

   - \( \bar{n} \): Order of subgroups to be generated obtained by multiplying two \( \frac{L}{2} \)-bit secret random primes \( q_1 \) and \( q_2 \).
   - \( G_1 \): \( \bar{n} \)-elements subgroup from \( p_{BGN} + 1 \) points in a supersingular curve \( E_{BGN} \):
     \[ y^2 \equiv x^3 + 1 \mod p_{BGN} \over F_{p_{BGN}} \]  where \( p_{BGN} = (\ell \in Z^+).n - 1 \equiv 2 \mod 3 \).
Figure A.3. Homomorphism’s Timeline

- $P_{BGN}$, $\hat{P}_{BGN}$: Random generators of $G_1$.
- $\hat{P}_{BGN}$: Generator of $G_{q_1} \subset G_1$ subgroup of order $q_1$ and $\hat{P}_{BGN} = \hat{P}^{(q_2)}$.
- $G_2$: An order $\bar{n}$ subgroup in extension field ($p_{BGN}^2$ points) $F_{p_{BGN}}^{(q_2)} = \{a_1 x + a_0\}$ where $x$ and $a_i$ are $\in F_{p_{BGN}}$.
- $e$: Modified Weil pairings $e : G_1 \times G_1 \rightarrow G_2$ to map between $G_1$ and $G_2$ as in:

**Definition A.4** (Modified Weil Bilinear Pairings [10]). For any two points $\hat{P}$ and $\hat{Q}$ in order $\bar{n}$ group $G_1$ of an elliptic curve $E_{BGN}$, the modified Weil pairing is defined as $e(\hat{P}, \hat{Q}) = \frac{(f_{\hat{P}}(Q + \hat{R}))}{(f_{\hat{P}}(\hat{R}))} \cdot \frac{(f_{\hat{Q}}(-\hat{R}))}{(f_{\hat{Q}}(\hat{P} - \hat{R}))}$ where $f_P$ and $f_Q$ are two rational functions with divisors $\text{div}(f_{\hat{P}}) = n.\hat{P} - n.\hat{O}$ and $\text{div}(f_{\hat{Q}}) = n.\hat{Q} - n.\hat{O}$ ($\hat{O}$ is the point of infinity) and $\hat{R}$ is a random point on $E_{BGN}$.

2. Encryption: To encrypt a message $m \in \{0, 1, ..., (< q_2)\}$ using $pk_{BGN}$, $m$ is transformed first to be $G_1$’s element $P_{BGN}^m$ before blinding it by a randomized form of $G_{q_1}$’s generator $\hat{P}_{BGN}$ [10]. The security of encryption is based on the hardness of the Subgroup Decision Assumption (SDA):
**Definition A.5 (SDA Problem [10]).** Given an encrypted element $\hat{S} \in G_1$, it is infeasible to infer whether $\hat{S}$ is in a subgroup $G_{q_1}$ within $G_1$. In other words, the adversary cannot decide to which group $\hat{S}$ belongs. Any attacker has a negligible advantage of distinguishability: $\text{Adv}_{AT}(L) = |Pr((\hat{S} \in G_1)) - Pr(\hat{S} \in G_{q_1})| \leq 0.5$.

To fulfill such requirement, factors $q_1$ and $q_2$ must be kept hidden and large leading to large group order (e.g. for $L = 128, n = 2^{1024}$) and large ciphertexts such that $C \in G_1$ of size (say 1 Kbit) $>> m$’s and in $G_2 \in F_{\varphi}^{\varphi BGN}$ would double-size $C$ (say 2 Kbit) because of $\varphi = 2$ [10, 100]$^{23}$.

3. **Decryption:** To get message $m$ back using $sk_{BGN}$, BGN computes $C^{q_1} = (P_{BGN}^m \cdot \hat{P}_{BGN}^{\varphi})^{q_1} = P^{q_1m} \cdot \hat{P}_{BGN}^{\varphi}$. Substituting $\hat{P}_{BGN} = \hat{P}_{BGN}^q$, the term $\hat{P}_{BGN}^{\varphi}$ dies out (identity of the group) in order $\bar{n}$ groups and we have $C^{q_1} = P_{BGN}^{q_1m}$ to retrieve $m$ from using the discrete logarithm $m = D\log_{BGN}(^{q_1m})$. To efficiently decrypt, $m$ must originate from small space to have small decryption time $O(\sqrt{|m| \leq q_2})$ [BGN uses Pollard’s Lambda Algorithm that has exponential computation time therefore $m$ should be small] [10]. It is worthy to mention here that the DLP is hard in $G_1$ and because pairings lessens its hardness, $G_2$’s size must be large [100].

With these three algorithms, Boneh et al. afford additive and multiplicative homomorphisms (Fig. A.4). For the additive homomorphism, multiplying two ciphertexts $C_1 = enc_{pk_{BGN}}(m_1)$ and $C_2 = enc_{pk_{BGN}}(m_2)$ in $G_1$ or $G_2$ gives a ciphertext representing the addition of the real two plaintexts and of a size no greater than $C_1$ and $C_2$ themselves thus BGN keeps a fixed sized ciphertexts independent from the number of conducted operations. On the other hand, the multiplicative homomorphism can

---

$^{23}$The value of $\varphi$ has to be carefully set; it has to be small enough to allow an efficiently computable pairings [100].
only be achieved making use of the modified Weil pairing $e : G_1 \times G_1 \to G_2$ such that to get the multiplication of two plaintexts, pair their corresponding $G_1$ ciphertexts before blinding it with a randomized element $\hat{P}_{BGN} = e(P_{BGN} , \hat{\gamma} P_{BGN})$ from $G_2$ where the result $C$ would live:

$$C = e(C_1, C_2) \cdot \hat{P}_{BGN} = e(P_{BGN}^{m_1} , \hat{P}_{BGN}^{r_1} , P_{BGN}^{m_2} \cdot \hat{P}_{BGN}^{r_2} \cdot \hat{P}_{BGN})$$

$$= e(P_{BGN}^{m_1}, P_{BGN}^{m_2}) \cdot e(P_{BGN}^{r_2}, \hat{P}_{BGN}^{r_1} \cdot P_{BGN}^{m_2} \cdot \hat{P}_{BGN}^{r_2}) \cdot \hat{P}_{BGN}$$

$$= e(P_{BGN}^{m_1, m_2}) \cdot e(P_{BGN}^{(m_1, r_2 + m_2, r_1 + f)}) \cdot e(\hat{P}_{BGN}^{(r_1, r_2)}) \cdot e(P_{BGN}^{(r_1, r_2, q_2)}) \cdot e(P_{BGN}^{r_1, r_2, q_2})$$

(A.3)

By decrypting $C$ with $sk_{BGN}$, in $C_{q_1} = (e(P_{BGN}, P_{BGN}^{(m_1, m_2)}) \cdot e(P_{BGN}, P_{BGN}^{(m_1, r_2 + m_2, r_1 + f)}) \cdot q_2 \cdot e(P_{BGN}, P_{BGN}^{r_1, r_2, q_2}))$, all terms of power $q_1, q_2$ would die out (identity of order $\bar{n}$ group) leaving only

$$C_{q_1} = e(P_{BGN}, P_{BGN}^{(m_1, m_2, q_1)})$$

where the DLP is used to easily obtain $m_1, m_2 = D\log_{e(P_{BGN}, P_{BGN}^{q_1})}$. Since there is no pairing on $G_2$, no more multiplications are allowed [10].

![Figure A.4. BGN Homomorphism](image)
Bibliography


