ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ÉTUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : **3978**

THÈSE

en vue de l'obtention du : **DOCTORAT**

Structure de Recherche: Intelligent Processing Systems & Security (IPSS)
Discipline: Informatique
Spécialité: Cryptographie et Sécurité de l'information

Présentée et Soutenue le : ***05 Septembre 2024***

par:

Abdelkrim IMGHOURE

New Authentication Processes in Vehicular Ad-Hoc Network

Devant le JURY :

Mohammed BENKHALIFA	PES, Université Mohammed V, Faculté des Sciences - Rabat	Président du jury
Ghizlane ORHANOU	PES, Université Mohammed V, Faculté des Sciences - Rabat	Examinatrice/ Rapportrice
Khalid ZINE-DINE	PES, Université Mohammed V, Faculté des Sciences - Rabat	Examinateur/ Rapporteur
Youssef BENTALEB	PES, ENSA – Kénitra	Examinateur/ Rapporteur
Ahmed EL-YAHYAOUI	PH, Université Mohammed V, Faculté des Sciences - Rabat	Co-encadrant de Thèse
Fouzia OMARY	PES, Université Mohammed V, Faculté des Sciences - Rabat	Directrice de Thèse

Année Universitaire: 2023 – 2024

DEDICATION

This dissertation is dedicated to my family, whose unwavering support and encouragement have been my foundation throughout this journey. To my parents, thank you for instilling in me the values of perseverance and curiosity. Your sacrifices and belief in my potential have inspired me to reach for my dreams.

I also dedicate this work to all those who have inspired me along the way—mentors, friends, and colleagues—who have shaped my academic path and fueled my passion for knowledge. This achievement is a reflection of your influence and support. The changes made are minimal, focusing on enhancing clarity and maintaining a smooth flow.

ACKNOWLEDGMENTS

I would like to thank our research unit, Intelligent Processing Systems & Security (IPSS), for providing guidelines and material support to conduct this research work.

I would like to express my deepest gratitude to my advisors, Dr. Fouzia Omary and Dr. Ahmed El-Yahyaoui, for their unwavering support, guidance, and mentorship throughout my doctoral journey. Their expertise, patience, and encouragement have been invaluable in shaping my growth as a researcher and scholar. I am truly thankful for the opportunity to work under their supervision and learn from their vast knowledge and experience.

I would also like to extend my heartfelt appreciation to the chair of my dissertation committee, Dr. Mohammed Benkhalifa, for his insightful feedback, constructive criticism, and valuable time. His thoughtful suggestions and challenging questions have greatly improved the quality of my dissertation and pushed me to think more critically and deeply about my research.

I extend my thanks to the members of my dissertation committee, Dr. Ghizlane Orhanou, Dr. Khalid Zineddine, and Dr. Youssef Bentaleb, for their constructive feedback and valuable insights that have contributed to the refinement of this dissertation.

To my family and friends, words cannot express how thankful I am for your unwavering support and encouragement. You have been my rock throughout the challenges and triumphs of this journey. Your belief in me has given me the strength to persevere and reach this milestone.

I am truly grateful to everyone who has contributed to my success and growth during this remarkable journey. Thank you all for your invaluable support and for helping me achieve this significant accomplishment.

RÉSUMÉ

Dans les réseaux véhiculaires (VANET), les nœuds sont composés de véhicules et d'unités d'infrastructure routière (RSU) communiquant via des liens V2V et V2I. Assurer simultanément l'authentification et l'anonymat des nœuds représente un défi de sécurité majeur. Les VANETs doivent satisfaire des exigences de sécurité fondamentales : authentification, intégrité, non-répudiation, protection de la vie privée, traçabilité et résistance aux cyberattaques. Le temps d'authentification des signatures est aussi critique étant donné la nature dynamique des échanges entre nœuds mobiles. Bien que de nombreux travaux aient proposé des protocoles d'authentification conditionnelle préservant l'anonymat, ces protocoles existants présentent souvent des limitations en termes de non-répudiation entre l'autorité de confiance et les nœuds en cas de litige. Notre recherche vise à concevoir de nouveaux protocoles CPPA offrant un niveau de sécurité supérieur, en garantissant une non-répudiation renforcée entre la TA et les nœuds, tout en satisfaisant les exigences de sécurité connues pour les VANET. Chaque nouveau protocole exploite différentes approches cryptographiques (symétriques, asymétriques, hybrides) basées sur les courbes elliptiques ou les couplages. Ils visent à optimiser les temps d'exécution et à réduire les coûts de communication lors de l'authentification, en introduisant des techniques avancées comme la vérification par lot, les multi-signatures et l'agrégation de signatures.

Mots clés : Réseau Ad-hoc de Véhicules, Authentification, Anonymat, signature, Agrégation.

ABSTRACT

In VANET, vehicles and RSUs are considered nodes that exchange information using V2V and V2I communications. In this context, satisfying authentication and anonymity simultaneously is a significant security challenge. Additionally, VANET has to meet well-known security requirements, namely authentication, integrity, non-repudiation between nodes, privacy preservation, traceability, and resistance to several known attacks. During signature verification, execution time is also a challenging component, given that nodes exchange messages in a dynamic environment. Although many works are based on Conditional Privacy Preserving Authentication, existing protocols seem to have a limited level of security regarding non-repudiation between the TA and nodes. Our research aims to develop new protocols that provide a higher level of security compared to existing ones. Our new protocols utilize different cryptographic approaches, i.e., symmetric, asymmetric, and hybrid based on ECC and BP, while meeting the known security requirements in VANET. Furthermore, our protocols guarantee true non-repudiation between the TA and a node in a dispute and provide efficient execution time and communication costs during message transmission and signature verification. Each protocol also offers efficient authentication features such as batch verification of signatures, multi-signatures, and signature aggregation.

Keywords: Vehicular Ad-hoc Network, Authentication, Anonymity, Signature, Aggregation.

SUMMARY

In Vehicular Ad Hoc Networks (VANET), vehicles and Roadside Units (RSUs) are classified as nodes that engage in the exchange of information through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. This communication framework is essential for the development of intelligent transportation systems, as it enables vehicles to share critical information regarding traffic conditions, hazards, and other relevant data that can enhance road safety and efficiency. However, within this context, achieving both authentication and anonymity simultaneously is recognized as a significant security challenge. The dual requirement for vehicles to verify each other's identities while maintaining user privacy complicates the design of secure communication protocols.

Furthermore, VANETs have to satisfy a comprehensive set of well-known security requirements that are crucial for ensuring the integrity and reliability of the network. These requirements include authentication, which verifies the identity of the nodes involved in communication; integrity, which ensures that the information exchanged remains unaltered during transmission; non-repudiation, which prevents any party from denying the authenticity of their communications; privacy preservation, which safeguards sensitive user data; traceability, which allows for tracking of messages if necessary; and resistance to various known attacks, such as spoofing, eavesdropping, and denial-of-service attacks. Each of these elements plays a vital role in maintaining the overall security posture of the VANET, and failure to address them could lead to serious vulnerabilities.

During the verification of signatures, execution time emerges as a critical factor, particularly in a dynamic environment where nodes are constantly on the move and exchanging messages. The rapid pace at which vehicles operate necessitates efficient authentication processes to ensure that communications are both timely and secure. Despite the existence of numerous studies focused on Conditional Privacy Preserving Authentication (CPPA), many of the current protocols exhibit limitations in their security capabilities, particularly regarding non-repudiation between the Trusted Authority (TA) and the nodes. This gap in security highlights the need for further research and development in this area.

In light of these challenges, our research work aims to develop new protocols that provide a higher level of security compared to existing solutions. These new protocols will utilize a variety of cryptographic approaches, including symmetric, asymmetric, and hybrid methods based on Elliptic Curve Cryptography (ECC) and Bilinear Pairing (BP). By leveraging these advanced cryptographic techniques, our protocols will not only meet the established security requirements in VANET but also enhance the overall security framework.

Moreover, our protocols are designed to guarantee true non-repudiation between the Trusted Authority and a node in the event of a dispute. This feature is critical for ensuring accountability and trust within the network. In addition, we aim to provide efficient execution times and reduced communication costs during the transmission of messages and the verification of signatures. Each protocol will incorporate advanced authentication features, such as batch verification of signatures, multi-signature schemes, and aggregation of signatures. These features are intended to optimize the authentication process, allowing for quicker and more reliable communications among nodes.

For instance, the batch verification of signatures allows the Trusted Authority to verify multiple signatures simultaneously, significantly reducing the time required for authentication. Multi-signature schemes enable a group of nodes to collectively sign a message, enhancing the security and integrity of the communication. Similarly, signature aggregation techniques can combine multiple signatures into a single compact signature, minimizing the amount of data that needs to be transmitted and processed.

In conclusion, our research addresses the pressing need for enhanced security mechanisms in VANETs. By developing new protocols that prioritize both security and efficiency, we aim to contribute significantly to the field of vehicular networking. Our work not only seeks to improve the current state of VANET security but also lays the groundwork for future advancements in intelligent transportation systems, ultimately promoting safer and more efficient roadways for all users. The successful implementation of our protocols could lead to a more secure vehicular communication environment, fostering greater trust among users and paving the way for the widespread adoption of VANET technologies.

LIST OF ACRONYMS

AS	Application Server
BP	Bilinear Pairing
BN	Barreto-Naehrig
BLS	Barreto-Lynn-Scott
CCPPA	Certificateless Conditional Privacy-Preserving Authentication
CPPA	Conditional Privacy-Preserving Authentication
CLAS	Certificateless Aggregate Signature
CL-based	Certificateless-based
CL-PKC	Certificateless-based Public Key Cryptography
CSPRNG	Cryptographically Secure Pseudo Random Number Generation
DSA	Digital Signature Algorithm
DOS	Denial of Service
DSRC	Dedicated short-range communications
ECDSA	Elliptic Curve Digital Signature Algorithm
EUF-CMA	Existential Unforgeability under Chosen Message Attack
ECC	Elliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECCDHP	Elliptic Curve Computational Diffie–Hellman Problem
ECDLA	Elliptic Curve Discrete Logarithm Assumption
ECCDHA	Elliptic Curve Computational Diffie–Hellman Assumption
ECDHKE	Elliptic Curve Diffie-Hellman Key Exchange
GPS	Global Position System
HCBS	Hybrid Cryptography-based Scheme
HMAC	Hash-based Message Authentication Code
MTH	Map-To-Hash function
IoT	Internet of Things
ID-based	Identity-based
ID-PKC	Identity-based Public Key Cryptography
MANET	Mobile Ad-Hoc NETwork
MITM Attack	Man-In-The-Middle Attack
MNT	Miyaji-Nakabayashi-Takano
MultiSig	Multi-signature
OBU	On-Board Unit
PKI	Public Key Infrastructure

PK-Replacement	Public key replacement attack
RSA	Rivest–Shamir–Adleman
RSU	Roadside Unit
SXDH	Symmetric External Diffie-Hellman
SC-PKC	Self-Certified Public Key Cryptography
TPD	Tamper Proof Device
TPD	Tamper Proof Device
VANET	Vehicular Ad-Hoc NETwork
V2X	Vehicle-to-Everything
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
WAVE	Wireless Access in Vehicular Environments
XDH	External Diffie–Hellman
ECDSA-CCPPA	ECDSA-based Certificateless Conditional Privacy Preserving Authentication
ECDSA*-CCPPA	ECDSA*-based Certificateless Conditional Privacy Preserving Authentication
Schnorr-CPPA	Schnorr-based Conditional Privacy Preserving Authentication
CA	Certificate Authority
CDA	Credential Distribution Authority
EC	Elliptic curve
KGC	Key Generation Center
KDA	Key Distribution Authority
PKG	Private Key Generator
PoP	Possession of Proof
SL	Security Level
TA	Trusted Authority
TRA	Trace Authority
TL	Trust Level
VC	Vehicle

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

CHAPTER 1: GENERAL INTRODUCTION

This chapter introduces the background and the state-of-the-art of challenges related to anonymity and security in the context of VANET, the existing requirements in VANET, as well as our research motivations and major contributions.

INTRODUCTION

VANET is an advancement of MANET [1]. In the context of connected vehicles, VANET relies on V2X technology, which involves implementing a set of communication technologies among connected objects within the vehicular network [2]. V2X is considered a variant of IoT, enabling network nodes to communicate wirelessly in a dynamic environment through a multi-hop transfer protocol [3]. In VANET, vehicles and road infrastructures exchange various road-related applications with the rest of the network, such as road safety messages [4]. During traffic, the exchange of information between nodes aims to prevent potential events like accidents or congestion. Notably, autonomous vehicles require additional information from the network to detect other vehicles or pedestrians in their environment and make appropriate decisions accordingly. When driving, every vehicle regularly transmits real-time messages using an onboard unit known as an OBU. The transmitted messages encompass details regarding the vehicle's status, such as its location, speed, direction, and more [3], along with pertinent road-related information like weather conditions, construction zones, traffic congestion, and so forth. This approach allows drivers to receive safety-related messages, enabling them to proactively take precautions to prevent potential accidents, opt for alternate routes to alleviate traffic congestion, or yield to emergency services when necessary. In VANET, both vehicles and RSUs can transmit their messages to a TCC, which is considered as the management center of the network. Besides, RSUs act as intermediaries between the TCC and vehicles. When a RSU receives messages from vehicles within its coverage area, it can relay these messages to other RSUs in different regions or forward them to TCC. In this case, the TCC collects network information and uses it to manage traffic. For instance, it proceeds by adjusting traffic lights or recommending new routes to drivers to avoid potential traffic jams. Concerning network communication, VANET utilizes V2X technology to disseminate messages among network nodes [5], where V2X primarily encompasses V2V and V2I communications. By employing V2V communication, vehicles employ the DSRC protocol to broadcast messages to their surrounding vehicles, while they transmit messages to RSUs via V2I communication. Furthermore, these vehicular communications are deployed using IEEE 802.11p and WAVE standards. During message transmission in the network, an unauthenticated or an unauthorized vehicle could create traffic jams or even cause accidents by delivering false information to other vehicles. [6]. As a result, securing V2V and V2I communications among vehicles and roadside infrastructures is essential. In this context, the security requirements in VANET primarily involve ensuring the authentication of network participants, message integrity, protection of driver's private data, and non-repudiation in the event of a dispute between two network participants [7]. During exchange of messages, the authentication process in VANET ensures that only legitimate participants can engage V2V and V2I communications. Additionally, the

requirement related to message integrity ensures that messages are protected against alteration or modification by third parties. Plus, privacy protection ensures that the driver's personal information is not disclosed to third parties. Moreover, authentication safeguards against potential attacks such as location-based attacks [8]. Regarding the requirement related to non-repudiation, a transmitter cannot deny having signed a message in case of a dispute between a transmitter and a receiver [9]. Concerning network monitoring, it is desirable to designate a Trusted Authority which can be responsible for securing network communications using asymmetric cryptography and providing participants with pseudonyms. In addition, the TA should be able to identify a legitimate yet unauthorized vehicle that sends false information to the rest of the network. Practically, the authentication process should include certain parameters belonging to the authority to prove the legitimacy of participants during authentication. Meanwhile, participants should use valid pseudonyms to ensure anonymity. Consequently, ensuring both authentication and anonymity poses a significant security challenge in VANET, as vehicles need to be authenticated while keeping their identities secret to protect against potential attacks, such as: location-based attacks [10]. Regarding cryptographic techniques used in securing VANET communications, current cryptographic computations widely used are based on mathematical problems that are difficult to solve in a certain time. In practical applications, the established standardized solution relies on PKI. In this framework, a CA issues certificates to nodes to establish the connection between their public keys and pseudonymous. To authenticate a node, ECDSA algorithm is used as a standardized digital signature during vehicular communications. During the broadcast of a safety-related message, each vehicle should add a certificate and a digital signature for each transmitted message. The current implementation of this solution is defined in the international security standards ETSI-103-097 [11] and IEEE1609.2 [12], which provide guidelines on how to implement a digital certificate and an ECDSA signature for each transmitted message. Based on existing research in the literature, it is found that PKI is a trusted and a reliable authentication system in VANET communications.

PROBLEM STATEMENT

The current security challenge is finding a balance between authentication and anonymity during communications in VANET. Several research works propose CPPA protocols to preserve anonymity and achieve authentication. However, these research efforts have certain security limitations that make them vulnerable to various cyberattacks. Additionally, they face performance limitations during the signing and verification processes. In practice, the solution currently deployed, based on PKI, requires the insertion of a certificate and a signature for each transmitted message, introducing complexity in certificate management, distribution, and revocation. Moreover, adding a certificate to every message may risk overloading the bandwidth. Therefore, research efforts are focused on the development of new protocols that do not rely on certificates. An alternative solution called ID-based Protocol has been developed, where PKG issues private keys to nodes after verifying their identities, and where the public key is the user's identity. However, ID-based protocols seem to have a limited security since the authority provisioning the nodes with the private key holds significant power and could misuse valid private keys for its own interests by sending valid but falsified messages. In case of a dispute, it

cannot be proven whether the sent message originates from an unauthorized node or an unauthorized authority since both entities know the same private key.

To address the issues found in ID-based protocols, Al-Riyami and al. introduced a new protocol called a CL-based protocol, which constructs the private key as a combination of two keys: a partial key generated by the authority and a secret key generated by the node.

Regarding the security strength of authentication protocols, Girault defined three levels of trust, where higher TLs correspond to higher security strength [13]. According to this classification, an ID-based protocol is considered to satisfy a TL 1, while a standard CL-based protocol satisfies TL 2. In the case of PKI, it fulfills TL 3.

Based on our literature review, the existing protocols only satisfy levels 1 and 2 but not level 3. In our research work, we have developed new protocols that meet a TL 3 and fulfill the known security requirements in VANET. Furthermore, we have compared the performance of our protocols with several works to assess their efficiency.

MOTIVATIONS

In VANET, nodes continuously broadcast messages in dynamic and high-mobility scenarios, making the V2X network vulnerable to cybersecurity threats [14]. Consequently, real-world V2X communications necessitate a comprehensive set of security measures [15]. Currently, the prevailing standardized solution relies on a PKI model in which a CA delivers certificates to network nodes, confirming the legitimacy of possessing a pseudonym and a public key. Additionally, a transmitter node attaches a digital signature along with a certificate to prove its legitimacy. There are international security standards that provide guidelines for the software implementation of certificates and ECDSA digital signatures, that are defined in ETSI-103-097[11] and IEEE1609.2 [12]. However, it is acknowledged that the management process associated with certificate generation and distribution can be considered complex and challenging [16]. Additionally, adding a certificate to each transmitted message significantly increases the packet size, which could saturate the network bandwidth for a constrained bandwidth [17]. Furthermore, as the number of vehicles being sold increases each year, there is a risk of an increased number of distributed certificates. According to the literature, another drawback of deploying PKI in VANET is that sent messages can only be verified one by one, not in batches, resulting in a significant execution time in a highly dynamic environment [18]. In a dense road area, a receiving node receives multiple messages simultaneously. If a message is not processed within a given time, it is eventually abandoned when its waiting time expires or completed when it is received by an RSU [14]. To avoid losing the content of safety-related messages, a batch verification is central to several research works and is considered an alternative and effective solution to reduce message execution time and ensure message availability in VANET.

In this regard, several research works propose CPPA protocols to ensure secure communications and interesting performance to protect the privacy of participants [16]. However, we show that the existing works have a limited security regarding the non-repudiation requirement between the authority and network nodes. As an alternative, our research focuses on developing new

protocols that do not require certificate deployment, provide a level of security equal to PKI, and allow the implementation of different types of encryptions, i.e., symmetric, asymmetric, and hybrid. Moreover, we propose solutions to reduce message verification time using batch verification, signature aggregation, verification of an aggregated signature, and multi-signature processes. Regarding security proof, we demonstrate that our protocols meet security requirements in VANET and are EUF-CMA due to the difficulty of solving the ECDL and CDDH problems in polynomial time.

CONTRIBUTIONS

The gap found in the existing research works is related to security strength, execution speed during message authentication, and packet size of messages which result in security and performance issues within a dynamic environment. In this regard, we have developed 4 protocols with different cryptographic algorithms that overperform the existing protocols in the literature. Our schemes can be summarized as follows:

1. *ECDSA*-based scheme:* Our first protocol is a new asymmetric conditional anonymity-preserving authentication protocol based on the CL-based concept. It uses ECC and avoids mapping the hash function to a point and pairing. ECDSA*-based scheme achieves a TL 3 and meets all security requirements in VANET. Furthermore, it allows nodes to authenticate using the ECDSA* algorithm. In IoT, ECDSA*-based scheme can be used in VANET for a single message and batch verification. Thus, a RSU can assist VCs in verifying their messages in congested traffic areas. Our evaluation results show that ECDSA*-based scheme outperforms many protocols. Additionally, the ECDSA*-based scheme incurs lower communication costs compared to the studied CL-based protocols.

2. *Schnorr-based scheme:* Our second protocol is new asymmetric CPPA based on the CL-based concept and ECC cryptography. It enables VCs and road infrastructures to authenticate using the Schnorr algorithm. To authenticate the same safety-related message from different transmitters, a RSU can carry out a MultiSig and aggregate the signatures and send a single signature to a TCC. Moreover, a RSU can carry out a batch verification when receiving different safety-related messages from different senders and send one aggregated signature to a TCC. Our protocol achieves VANET requirements and turns out resilient to the rogue attack.

3. *CL-based Aggregate Signature (CLAS):* Our third protocol is a third asymmetric CL-based aggregate signature protocol based on pairing, with a TL 3, providing a strong non-repudiation and resistance to cyberattacks involving the replacement of *pk*s. Additionally, our protocol offers an authentication process during V2V and V2I communications. When a RSU receives multiple messages from VCs, it performs a batch verification to ensure that all signatures are valid, reducing computation time compared to verifying messages one by one. Afterward, the RSU aggregates the signatures and sends the aggregate signature to other RSUs and a TCC.

4. *HCBS-CPPA:* Our fourth protocol is a new CPPA protocol based on the CL-based concept and ECC cryptography with hybrid encryption. This protocol also satisfies all

VANET security requirements. On the one hand, it uses asymmetric encryption to provide non-repudiation. On the other hand, it employs symmetric encryption to offer a lightweight authentication process. We show that HCBS-CPPA is resilient against memory-based DoS attacks, as a verifier deploys certain security mechanisms related to a time interval to counter a potential "pollution attack".[19]. This latter attack occurs when an adversary attempts to flood a VC or road infrastructure with invalid messages. Finally, our simulation shows that our HCBS-CPPA protocol requires less execution time than many CL-based protocols and incurs lower communication costs during the authentication process.

THESIS ORGANIZATION

This thesis comprises seven chapters. The purpose of each chapter is detailed as following:

Chapter 1 introduces the background and the state-of-the-art of challenges related to anonymity and security in the context of VANET, as well as our research motivations, major contributions, and thesis organization.

Chapter 2 gives an introduction of the existing requirements in VANET. It also lists the existing cryptographic encryptions used to authenticate nodes in the network. We distinguish between Symmetric Encryption, Asymmetric Encryption, and Hybrid Encryption. In addition, we introduce cryptographic computations based on elliptic curves and mapping which are based on complex computational problems. This chapter also gives the existing public key cryptography-based approaches in the field of VANET, mainly certificate-based PKI, ID-based, CL-PKC and SC-PKC approaches.

Chapter 3 details our work results related to CL-based protocol with ECDSA authentication. It gives the VANET architecture of our protocol, cryptographic operations used during the authentication process and results from simulations.

Chapter 4 details our work results related to CL-based protocol with Schnorr authentication. It gives the VANET architecture of our protocol, cryptographic operations used during the authentication process and results from simulations.

Chapter 5 details our work results related to CL-based protocol with mapping cryptography for authentication. It gives the VANET architecture of our protocol, cryptographic operations used during the authentication process and results from simulations.

Chapter 6 details our work results related to CL-based protocol with hybrid cryptography for authentication. It gives the VANET architecture of our protocol, cryptographic operations used during the authentication process and results from simulations.

Chapter 7 gives a general conclusion of our four CL-based protocols for authentication, where each has its own security features and performance according to the studied use cases in VANET.

CHAPTER 2: BIBLIOGRAPHIC REVIEW OF AUTHENTICATION IN VANET

In this chapter, we introduce VANET security requirements as well as the existing cryptographic encryptions that allow to ensure a secure authentication based on known complex mathematical problems.

2.1 Introduction of security requirements in VANET

During V2V and V2I communications, the VANET network should meet a set of security requirements to protect participants from fake messages originating either from unauthorized participants within or external to the network. Consequently, security measures should allow nodes to ignore any messages from nodes external to the network while also rejecting any message that does not pass the authentication process from an internal node. Furthermore, the authority should be capable of regularly distributing pseudonyms and monitoring messages that are transmitted within the network. It should also trace a legitimate but unauthorized participant in case of fraudulent actions. In this way, the authority can revoke a node when it requests new pseudonyms, rendering it unable to participate in future network communications. VANET security requirements can broadly be defined as follows:

Authentication. Authentication is a security requirement that ensures that only messages from legitimate participants can be accepted after verification. It involves including a digital signature with the message to attest that the sender is a legitimate node which possesses a valid private key. Since the authentication process should meet tight time constraints, VCs should be able to transmit and process messages within a reasonable timeframe. Otherwise, messages will be discarded because an OBU has limited resources. This is especially important when certain messages need to be transmitted over long distances or when vehicles are traveling on a high-speed highway, which may require additional transmission time for message processing [15].

Integrity. When a message is transmitted within VANET, it should include cryptographic evidence to confirm its integrity, ensuring that it has not been tampered with or altered during its transmission from the sender to the receiver. This is crucial for protecting against an adversary who might intercept the message and attempt to modify it in transit.

Non-repudiation between two nodes. In a dispute between a transmitter node and a receiver node, the use of an asymmetric cryptography ensures this requirement. With asymmetric cryptography, the transmitter cannot deny having signed the message. Additionally, the transmitter node uses a valid pseudonym, which is provided by the authority to sign its message in VANET. This combination of cryptographic techniques helps establish accountability and traceability in case of disputes or security incidents within the network.

Protection of privacy. When a VC transmits a message to another VC or RSU, it should use a pseudonym provided by the authority rather than its actual identity. This practice is crucial for preventing potential cyberattacks, such as location-based attacks, where an adversary could

exploit a VC's true identity for harmful purposes. Using pseudonyms enhances the security and privacy of communications within the VANET by concealing the true identity of the sender.

Traceability. In VANET, an authority should have the capability to monitor communications within the network. Furthermore, the management system should possess the capability to trace a legitimate yet unauthorized node by employing its pseudonym to unveil its actual identity. The authority publishes public parameters that only a legitimate node can use to encrypt its messages and prove its legitimacy.

In our research, our protocols empower the authority to identify an unauthorized node through the utilization of two key components: its pseudonym and public key. This novelty is not present in the studied protocols in research that only use the pseudonym component to identify a node. By incorporating both the pseudonym and public key, our protocol provides an enhanced method for the authority to detect and trace unauthorized nodes, thereby improving the security and accountability of the VANET network.

Unlinkability. During the dissemination of messages within the network by a node, each message should include a pseudonym to conceal the sender's identity. Furthermore, messages transmitted by the same node should contain pseudonyms that are distinct from one another. The purpose of pseudonym unlinkability is to prevent an adversary from linking pseudonyms and tracking a driver through a location-based attack. This safeguard is essential for preserving the privacy and security of participants in the VANET network.

Additional security measures in VANETs. In our work, we analyze other security measures that we consider as additional measures because research works often do not include them in their security analysis. However, our research work has focused on proposing solutions to implement these security measures.

- ***Trust Level:*** It represents the security strength of an authentication protocol. In this context, we define three TLs according to the classification provided by Girault [13] :

i. *TL 1*. This level of trust is satisfied when the TA knows the private key of a user.

ii. *TL 2*. is achieved when the TA does not know the private key of the user. However, the TA can still encrypt and send messages under the identity of a legitimate user by creating a fake certificate or a false public key.

iii. *TL 3*. is satisfied when the TA cannot create a valid key pair for a user. Furthermore, it can be proven that an unauthorized TA has created a false public key in a dispute between the authority and the user. Thus, Level 3 enables non-repudiation between the authority and a user.

Unlike many ID-based protocols and normal CL-based protocols, which respectively satisfy TLs 1 and 2, we are developing an authentication strategy in our protocol that achieves a TL 3. Moreover, our strategy guarantees non-repudiation in the event of a dispute that could arise between the authority and a network node.

- ***Strong Non-Repudiation:*** In our research, we have introduced this requirement, which is defined as follows: Strong non-repudiation can be guaranteed when we can prove that either an unauthorized authority or node has falsified a valid but unauthorized public key

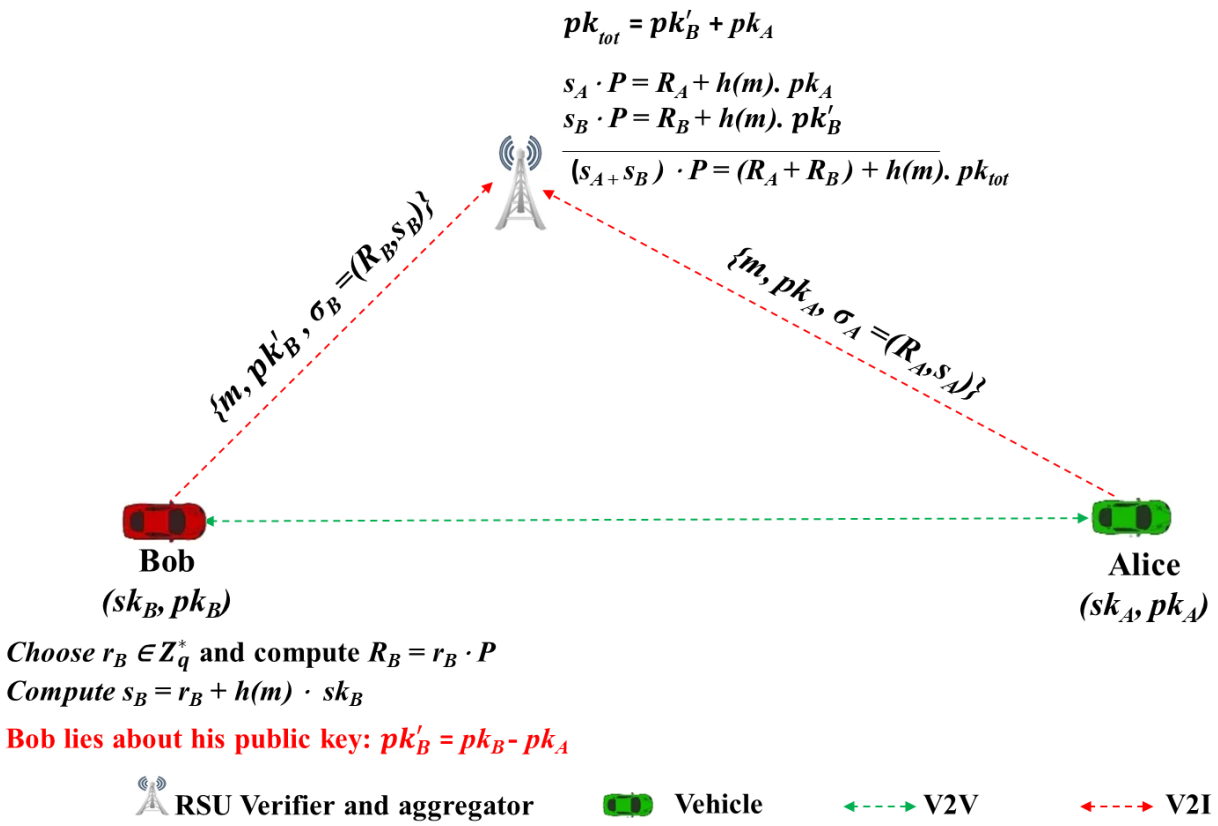
in case of a dispute. This requirement is typically satisfied when a protocol is categorized with a TL 3. Furthermore, this requirement implies that the protocol is resilient to cyberattacks related to PK-Replacement.

- ***Database Security:*** We have introduced this requirement in our research and is defined as follows: During network management, the authority maintains a database containing the true identities of nodes and regularly provides nodes with pseudonyms after verifying their true identities. Given that the actual identities of nodes are regarded as sensitive information, our research comes up with a solution that safeguards the information of network nodes against potential cyberattacks, whether originating from within or outside the authority.
- ***Resilience against PK-Replacement Attack:*** In our research, we introduced this requirement, which is defined as follows: In the case of a protocol with a TL 2, a node can replace its original public key with a new one without any restrictions. As a result, the TA cannot reveal the public key of an unauthorized node. Moreover, it cannot be proven whether an authority or a node generated this new public key due to the lack of security restrictions. In our developed protocols, the public key is constructed in a way that ensures uniqueness through collaboration between the authority and node.
- ***Resilience against Impersonation Attack:*** Many works propose ID-based protocols. However, these protocols suffer from the impersonation problem. In an ID-based protocol, the authority calculates and provides private keys to network nodes after verifying their identities. Therefore, an unauthorized authority can use these private keys for harmful purposes by impersonating a legitimate node. Same thing in a normal CL-based protocol, in which impersonation attack is possible.
- ***Resilience against Single Point of Failure Related to Key Storage:*** This requirement has been added in our research and is defined as follows: In case network nodes store the same encryption keys, which is typical of symmetric encryption, it represents a security flaw. A compromise of a single VC is sufficient to compromise the entire network. This vulnerability is due to all network nodes are storing the same private keys. Therefore, it is preferable to use asymmetric encryption, where each node possesses unique key pairs.
- ***Resilience against MITM Attacks:*** This attack occurs when an adversary intercepts communication between a transmitter and a receiver node, and then gains access to sensitive data, which it can then forge for its own benefit. To prevent such an attack, the transmitter should include a signature in the transmitted messages, ensuring that the message has not been altered or impersonated. Note that message encryption is not necessary in VANET since the data is meant to alert vehicles and pedestrians to potential events, and confidentiality is not a requirement here.
- ***Resilience against Memory-Based DoS Attack:*** This attack occurs when a receiver is flooded with invalid messages, potentially compromising message availability. In the case of this attack, the OBU consumes a significant amount of resources, known as "pollution attacks" [19]. As a result, new legitimate messages cannot be stored in memory and may expire or be ignored. In our research, described as a hybrid protocol, when a RSU or VC receives messages, the receiver node uses time intervals to compel the transmitter not to overload its memory. Thus, an unauthorized transmitter cannot

flood a receiver node's memory with invalid messages, as the received messages outside these intervals would not be verified and would be abandoned.

- ***Resilience against Replay Attacks:*** This attack occurs when an adversary retransmits the same message later to convey valid but outdated information, potentially causing an unwanted event. To counter this attack, the transmitter should sign the message by encrypting the timestamp, making the generated signature unique and non-reusable.
- ***Availability and Batch Signature Verification:*** In VANET, message signing, and verification time should be acceptable given the limited resources of an OBU, such as CPU and memory. When the memory is full or the CPU is occupied with long and heavy computations, pending messages are abandoned and remain unverified. Therefore, it is desirable to verify messages using a batch verification, reducing the execution time of a series of messages, and preserving memory space.
- ***Multi-Signature and Aggregation of Signatures:*** MultiSig occurs when the receiver receives the same message from different transmitters. In the case of MultiSig, the receiver can aggregate all received signatures into a single signature and send the resulting signature to other nodes in the network, using signature aggregation. These computations reduce execution time and do not overload the network bandwidth.
- ***Resilience against Rogue Attack:*** This attack occurs during the MultiSig process, where an adversary can validate a message using a false public key, as shown in Figure 1.

These measures enhance the security and robustness of VANET, addressing various threats and challenges related to public communications.



m : **transmitted message**

P **: Base point**

(R_i, s_i) **:** **Schnorr signature**

(sk, pk) **:** **Key pair used in Schnorr signature by a VC**

Fig 1. Rogue Attack in VANET

2.2 Existing cryptographic encryptions in VANET

Here, we introduce three types of encryptions used in message signing: Symmetric Encryption, Asymmetric Encryption, and Hybrid Encryption.

Symmetric Encryption, also known as shared-key encryption, which is employed when both the sender and the receiver utilize an identical key for both encrypting and decrypting data.

Asymmetric Encryption: This encryption is defined by the involvement of both *sk* and *pk* during the encryption and decryption processes.

Hybrid Encryption: It can be defined as the use of both symmetric and asymmetric encryption simultaneously.

2.3 ECC

ECC is considered as a set of techniques used in PKC, where it deploys ECs over finite fields. By using shorter key sizes, ECC can provide a security equivalent to that provided by RSA. Furthermore, ECs are used in the construction of digital signatures. For instance, ECDSA is considered as an efficient algorithm that employs ECC and generate shorter key sizes [20]. From security perspective, a 256-bit elliptic curve-based key offers a comparable level of security to that of a 3072-bit RSA-based key[21]. Moreover, ECC-based cryptosystems demand lower computational power, making them more efficient in various computing environments.[21].

2.3.1 Cryptographic computations based on elliptic curves

The elliptic curve group. E(Fp) is defined over a finite field F_p of prime order p. It can be represented by the equation: $y^2 = x^3 + ax + b \bmod p$, where both a and b are elements of the field F_p. This group comprises of:

- An infinity point *(O).*
- The points *(x, y)* that satisfy the elliptic curve equation.

They both form a cyclic additive group denoted as G with an order q, representing the number of points in the group. Within this group, a base point P is chosen, and it has an order denoted as n. This n is the smallest positive integer for which $nP = O$ and belongs to the set Z_q^*. The selection of such a base point P effectively generates the entire group G. Notably, both p and q are large prime numbers, contributing to the security of the cryptographic system.

ECC Computations:

- Points $R, P, S \in E(Fp)$*:* These represent points located on the elliptic curve and are members of a group denoted as G.
- Point Addition $(R = P + S)$*:* This operation involves adding points P and $S \in E(Fp)$, resulting in a new point R, which is the intersection point of the line connecting P and S with the elliptic curve.
- Scalar Multiplication *(nP):* Scalar multiplication in the group G is defined as repeatedly adding the same point P to itself n times, where n is an integer from the set Z_q^* (integers modulo a prime number q). This operation allows the calculation of a new point by iteratively adding the point to itself.
- Point at Infinity (O): Within ECC, the point at infinity, represented as O, serves as an identity element. When combined with any other point, it leaves the result unchanged.

Special Cases:

- When P is identical to S, the outcome of $P + S$ can be computed.
- When P is equivalent to the negative of S *(-S),* the result of $P + S$ is the point at infinity *(O).*

ECC is a crucial cryptographic technique, employed in various security applications, due to its efficiency and robust security characteristics. It leverages the mathematical properties of elliptic curves to enable secure key exchange and digital signatures.

2.3.2 Problems and assumptions based on elliptic curves

In the field of elliptic curve-based cryptography, there exists complex problems deemed computationally infeasible, where no known algorithm can solve these problems efficiently in polynomial time. These problems are defined as follows: the *Discrete Logarithm Problem based on elliptic curves* and the *Diffie-Hellman Problem based on elliptic curves*. These problems form the foundation of security provided by ECC and are detailed in the following subsections.

2.3.2.1 Discrete Logarithm Problem based on Elliptic Curves

ECDLP [9], [22] : For all points *P* and *Q* belonging to the elliptic curve *E* over the F_p and for any integer *x* from Z_{n-1}^*. If *Q* equals *xP*, then determining the value of *x* is a computationally challenging task.

2.3.2.2 Assumption regarding the Discrete Logarithm Problem based on Elliptic Curves

ECDLA [9], [22]: The assumption here is that there are no known algorithms capable of solving the ECDLP with a non-negligible probability in polynomial time.

2.3.2.3 Diffie-Hellman Problem Based on Elliptic Curves

ECCDHP [9], [22]: In Elliptic Curve Cryptography, for all points *P* and *Q* belonging to the elliptic curve *E* over the finite field F_p, where *x* and *y* are integers from the set Z_{n-1}^*, the problem of calculating the product *xyP* is computationally difficult, when only provided with points *R* and *Q* (*R = xP* and *Q = yP*). This computational difficulty forms the basis of the security of elliptic curve cryptography.

For all points *P* and *Q* belonging to the Elliptic Curve *E* over the field *Fp*, and for any integers *x* and *y* from Z_{n-1}^*. If *R* is defined as *xP* and *Q* is defined as *yP*, then calculating the product *xyP*, given *R* and *Q*, is a computationally challenging task.

2.3.2.4 Hypothesis Regarding Elliptic Curve Diffie-Hellman Problem

Hypothesis regarding the Elliptic Curve Diffie-Hellman Problem (ECCDHA) [9], [22]: We assume that there are no known algorithms capable of solving the ECCDH problem in polynomial time with a non-negligible probability.

2.4 Mapping

2.4.1 Cryptographic computations based on Pairing

A pairing can be defined as a bilinear map on two subgroups of rational points of an elliptic curve *E*. Let *E* be over *GF(p)* with *p* its characteristic, and *k* is the minimum integer for which *q* is a divisor of $p^k - 1$, where *k* is called the embedding degree of *E* over *GF(p)*. Let G_1 be a cyclic subgroup of *E(GF(p))* of order *q*. In this case, there exists a cyclic subgroup of $E(GF(p^k))$ of order *q* that defines G_2. Let G_T be a subgroup of the multiplicative group $(GF(p^k))^*$ with order *q*. Now, we have G_1, G_2 and G_T be cyclic groups of order *q*, where G_1, G_2 are additive groups and G_T a multiplicative group. A pairing is defined as a bilinear map $e: G_1 \times G_2 \longrightarrow G_T$ if it satisfies the following properties [23], [24]:

Bilinearity: $\forall P \in G_1$ *and* $Q, R \in G_2$: $e(P, Q+R) = e(P,Q) \cdot e(P,R)$ *Also,* $\forall P, Q \in G_1$ *and* $R \in G_2$: $e(P+Q, R) = e(P,R) \cdot e(Q,R)$

Non-degeneracy: $\exists P, Q \in G_1, e(P, Q) \neq 1_{G_T}$.

Computability: For any $P, Q \in G_1 \times G_2$, There is an efficient algorithm to compute *e(P, Q).*

2.4.2 Problems and assumptions based on Pairing

In pairing-based cryptography, there is a challenging problem that is considered computationally infeasible, for which no known algorithm can find a solution it in polynomial time. This problem is Diffie-Hellman Problem based on pairing.

Computational Diffie-Hellman problem and assumptions based on Pairing

1) In case $G_1 = G_2$: the pairing *e* is called symmetric, and the setting is called *Type-1*. In addition, this symmetric pairing is based on DDH solver and CDH problem and where *Type-1* curves are supersingular.

Decisional Diffie-Hellman (DDH) assumption. For $P, xP, yP, zP \in G_1^4$, check whether $z = xy$ or not by checking whether $e(xP, yP) = e(P, zP)$.

Computational Diffie–Hellman (CDH) problem. Given $P, xP, yP \in G_1$ and an admissible pairing e: $G_1 \times G_1 \longrightarrow G_2$, compute *xyP* for unknown $x, y \in Z_q$ and *P* is a generator of G_1. The probability of success to solve a CDH problem instance in G_1 in a probabilistic polynomial-time algorithm *A* is defined as follows:

$$Succ_{A;G1}^{CDH} = Pr\,[A\,(P, xP, yP) = xyP \text{ with } x, y \in Z_q^*].$$

Computational Diffie–Hellman (CDH) assumption. a probabilistic polynomial-time algorithm *A* has a negligeable probability to succeed and resolve a CDH problem in G_1, in which $Succ_{A;G1}^{CDH}$ is negligible.

2) In case $G_1 \neq G_2$: the pairing *e* is called asymmetric. Here, we distinguish between *Type-2* and *Type-3:*

- If there is an efficient computable isomorphism $\psi: G_2 \rightarrow G_1$, then the setting is called *Type-2* (e.g., MNT) curves [23], [25]).

- If there exists no efficient isomorphism between G_1 and G_2, the setting is classified as *Type-3* (e.g., BN curves[26] or BLS curves [27]). Those pairings are constructed from either Weil [28] or Tate pairings [29] over suitable elliptic curve groups G_1 and G_2, and their modifications are constructed using the ate pairing[30] and the R-ate pairing [31]. In addition, the computational hardness assumptions can be defined as follows: *Computational Diffie–Hellman problem (CDH)* is intractable in G_1 and G_2. Plus, *DDH assumption* holds in both G_1 and G_2 in *Type-3* bilinear group and called *symmetric external Diffie-Hellman (SXDH) assumption*. Additionally, *DDH* assumption holds in G_1 and does not hold in G_2 in the *Type-2* setting, which is called *external Diffie-Hellman (XDH) assumption* [23]. In practice, it is believed that *XDH* assumption may hold in certain subgroups of MNT curves.

2.5 Comparison between computations based on Elliptic Curves and Pairing in Terms of Execution Time.

In many research work, CPPA protocols utilize various cryptographic computations such as computations based on elliptic curves (ECC), pairing, and/or MTH function. For instance, protocols defined in [16], [21], [32], [33], [34], [35] use computations based on elliptic curves, while [36], [37], [38], [39], [40], [41] employ pairing and mapping the hash function to a point (MTH). Even though pairing and MTH function allow to construct challenging problems regarding signature aggregation and batch verification, these computations are well-known for their complex implementation and also require more processing time compared to computations based on elliptic curves [42].

2.6 Existing PKC approaches related to authentication in VANET

In public key cryptography-based approaches, there are various methods applied to authentication protocols in VANET, each offering different level of security. In our work, we define a level of trust as the security strength of a protocol, where higher TLs correspond to higher strength.

Authentication based on PKC Approaches. According to the classification of TLs provided by Girault [13], PKC-based approaches can be categorized into three groups with varying strength levels:

i. *TL 1* is fulfilled when the TA possesses knowledge of the *sk* associated with a VC or RSU within the network.

ii. *TL 2* is satisfied when the TA does not know the *sk* of a VC or RSU. However, TA can still encrypt and send messages under the identity of a legitimate VC by creating a fake certificate or a false *pk*.

iii. *TL 3* is satisfied when the TA cannot create a valid *sk* or *pk* for a VC. Moreover, it can be proven that an unauthorized TA has created a false *pk* in a dispute between the authority and VC. Therefore, TL3 enables non-repudiation between the authority and the network nodes.

Furthermore, the existing approaches related to PKC can be classified into three groups: 1- Certificate-based PKI, 2- ID-PKC, and 3- CL-PKC. [22].

2.6.1 Certificate-based PKI

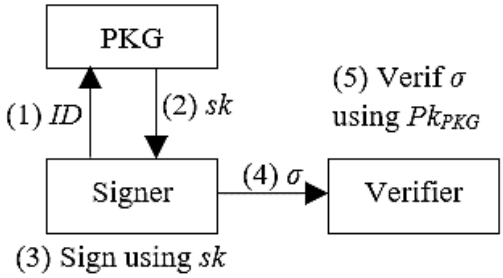
In VANET, a certificate-based protocol requires the inclusion of a digital certificate and a signature in the message during the authentication process [43], where a VC's *pk* is authenticated using the certificate provided by the authority. In this way, the certificate attests that a VC's *pk* is associated with its pseudonym. However, PKI is a protocol that suffers from two disadvantages, that can be defined as follows:

- *Flexibility and Certificate Management* [18]. Addressing flexibility poses the challenge of expanding the PKI to incorporate additional third parties to achieve wider network coverage. Meanwhile, certificate management encompasses the procedures of creating, disseminating, and verifying certificates. Moreover, extending PKI to new regions is challenging because it requires new management of the trust relationship between intermediate certification authorities in the trust hierarchy, as well as the analysis and development of a new trust model, etc.
- *Certificate Authenticity Verification process*. It is known as complex and demands significant calculations: during the process of verifying a signature, a VC requires numerous calculations due to certificate path verification, which can introduce significant delays in the authentication process. Furthermore, the distribution and management of certificates by the TA is a complex and costly process, especially as the number of Certificate Revocation Lists (CRLs) increases linearly with the production of new VCs each year [44], potentially saturating the network bandwidth.

In VANET, several research works [45], [46], [47], [48] use a certificate-based PKI protocol during the authentication process, where a Certification Authority (CA) distributes certificates to VCs and RSUs. Nevertheless, the storage, verification, computation, and revocation of certificates remain a challenge to meet VANET's requirements in terms of bandwidth and time constraints. Even though a PKI-based protocol is a secure solution, it still faces issues related to certificate management. According to the TL classification mentioned above, PKI satisfies TL3 [13].

2.6.2 ID-PKC

In order to tackle the concern related to certificate management, an ID-based protocol was designed by Shamir [49]. Within this protocol, the *pk* is associated with an entity and is generated based on specific attributes of its identity, which could include elements like a network host's IP address, an email address, or a phone number, among others. Once the PKG receives one of these parameters, it can provide the corresponding *sk* for the *pk*, as depicted in Figure 2. However, an ID-based protocol suffers from the issue of trust because the authority knows the *sk* of the user, which offers a limited security. As an illustration, an adversary PKG could exploit this breach to deceive and impersonate a node. Thus, the security requirement related to non-repudiation between the authority and user cannot be satisfied, as we cannot prove that a unauthorized PKG/user has falsified a valid signature using a valid *sk* [50].

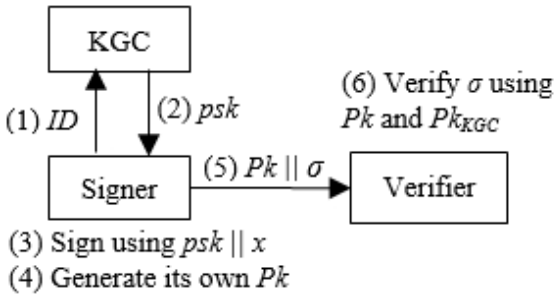


ID: Node's identity
sk : Signer's *sk*.
x : signer's *x*
Pk_{PKG}*:* PKG's *pk*
σ: Signature

Fig 2. ID-based Protocol

2.6.3 CL-PKC

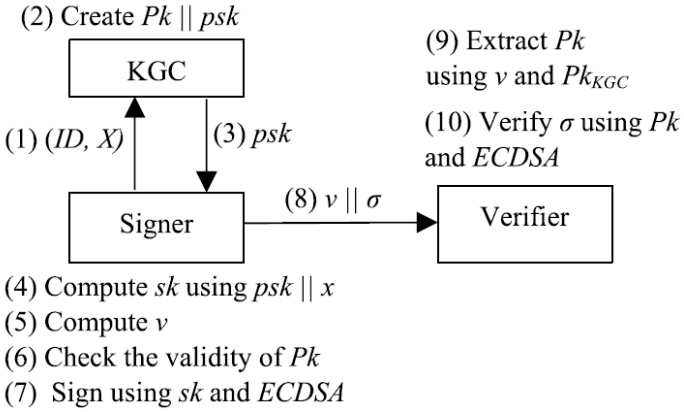
To resolve the key escrow issue in ID-based PKC, Al-Riyami and al. [49] designed a CL-based cryptographic scheme to simultaneously address both certificate management challenges and the key escrow issue [51]. In a normal CL-based protocol, a user's *sk* is the result of a combination of a partial private key (psk_i) provided by an authority known as KGC and a secret key (x_i) generated by the user after receiving psk_i. The combination of psk_i and x_i allows to generate a *sk* that will be used in authentication, as mentioned in Figure 3. However, this type of protocol does not satisfy the non-repudiation requirement between the authority and a user, as the *pk* is not unique. Consequently, this protocol remains vulnerable to the PK-Replacement attack. According to the classification given by Girault [13], a normal CL-based authentication protocol satisfies a TL 2.



ID : Node identity
psk : Signer's *psk*
sk : Signer's *sk*
Pk : Signer's *pk*
Pk_{KGC} *: KGC's pk*
*σ :*Signature

Fig 3. Existing CL-based protocol with a TL 2

To create a CL-based protocol with a TL3, which can attain an equivalent level of trust as a certificate-based PKI protocol, Al-Riyami and al. designed a compelling binding technique, albeit without a formal security proof [46], [51], in which a hash function links the user's identity to its *pk*. Additionally, Yang et al. have emphasized the security benefits of using this binding technique [52]. Hassouna et al. have explained the significance of this technique, which involves having a unique *pk* corresponding to the *sk* during the encryption process [53]. Consequently, this technique ensures the uniqueness of the *pk* and fulfills the requirement related to non-repudiation between the authority and user. In addition, this technique provides resilience against a PK-Replacement attack. By utilizing this technique, a standard CL-based protocol with a TL2 can be transformed into a CL-based protocol with a TL3. In our work, we have applied this principle to our protocols in VANET, as mentioned in Figure 4.



ID : Node identity
psk : Signer's *psk*
sk : Signer's *sk*
Pk : Signer's *pk*
x : Signer's *x*
X : ECC point generated by the signer
v : Cryptographic value calculated by the signer
Pk_{KGC} *:* KGC's *pk*
σ *:* Signature

Fig 4. Our CL-based protocol with a TL 3

2.6.4 SC-PKC

Girault introduced the concept of a SC-PKC [13] which is similar to a CL-PKC protocol [54]. This approach has been adopted in other studies with the aim of substituting certificates with "witnesses," aiming to reduce the communication overhead associated with the inclusion of a certificate [55], [56]. In a SC-PKC protocol, a user autonomously generates its *sk* along with the corresponding *pk*. Then, the user sends *pk* to TA. Subsequently, the TA associates the user's *pk* with his identity. As a result, the TA creates a "witness", which is considered as TA's signature, confirming the linkage between the identity and *pk*. However, it was noted by Saeednia that Girault's SC-PKC protocol does not achieve a TL3 but only meets a TL1 [57].

In the context of VANET, Zhang et al. created a protocol that aims at preserving anonymity. This protocol relies on SC-PKC [58]. The system primarily involves two entities: VCs and TA. Each VC creates its key pair using the TA's master key, denoted as $\alpha \in Z_q^*$. Subsequently, a user requests a "witness" from the TA to sign its safety-related message. As a result, the node incorporates the "witness" into its signed safety-related messages to simplify the verification of the signature. Nevertheless, a disadvantage of this type of protocols is that the master key is a common key among all VCs in the network. As a result, if a single VC is compromised, the entire system becomes vulnerable. This weakness was also observed in the reference [59], herein the system's encryption key *x* is distributed among all VCs in SC-PKC protocols. As a result, SC-PKC protocols typically achieve the same TL as that of an ID-based protocol, which is TL1.

2.7 Existing Protocols Related to Authentication in VANET

2.7.1 Existing Certificate-Based Protocols

In VANET, several anonymity-preserving authentication protocols have been designed and rely on the conventional PKI [11], [12], [60], [61]. However, these protocols necessitate a certificate for each *pk*. Namely, a signer appends a digital certificate to its safety-related message to demonstrate that its *pk* is associated with its pseudonym. Despite its TL3, a certificate-based approach results in a network overhead and substantial delays during the message verification process. Furthermore, the certificate management poses many challenges that encompasses certificate distribution and revocation procedures [18].

2.7.2 Existing ID-based Protocols

To address the issue of certificate overhead, Shamir developed the concept of ID-based cryptography [48]. In this regard, the ID-based protocols [16], [32], [33] have a drawback associated with the key escrow problem because the *sk* is exclusively generated by the authority. In the protocol introduced by He et al. [16], the TA selects a master key and provides the node with an identity, a password to a VC at the initialization phase. After that, it preloads those parameters into the VC's TPD. In [32], the PKG furnishes a user with a *sk*, that the user employs each time to generate signatures for specific safety-related messages. This *sk* is acquired during the registration phase. In the protocol developed by Wu et al. [33], a secret key *ppvi* is preloaded by the authority when the VC's identity is received during the purchase process. The VC then utilizes this x_i for signing its safety-related messages. In [62], the authors propose an ID-based digital signature protocol employing the ECDSA algorithm. Therefore, all the protocols [16], [32], [33] rely on secret cryptographic parameters provided by the authority to create signatures for safety-related messages. Following Girault's classification, all the above-mentioned protocols belong to the category of ID-based protocols. Consequently, they are vulnerable to the key escrow problem and achieve a TL 1 [13].

2.7.3 Existing CL-based Protocols

In VANET, numerous CL-based protocols based on elliptic curves have been developed. The protocols mentioned in [21], [34], [35], [36], [37], [63] are CL-based protocols with a TL2 that use elliptic curves during the encryption process. In the protocol developed by Kuo-Hui Yeh

[21], KGC delivers $D_i = (s_i, R_i)$ as its *psk* to network nodes. Then, the node adds a *x* to the *psk* to generate Pk_i as follows: $Pk_i = x_i \cdot P + R_i$. As a result, the transmitted safety-related message takes the following form $\{ID_i, Pk_i, \sigma_i = (T_i, \tau_i)\}$. If the *psk* is revealed during the transmission over an insecure channel, an adversary can obtain it and calculates a valid *pk* of its choice. In this case, the authority is unable to identify the unauthorized node based on its broadcasted *pk*. In the protocol proposed by Kang Li et al.[34], the node selects $x_{IDi} \in Z_q^*$ and calculates $vpk_{IDi} = x_{IDi} \cdot P$ during the key generation process. Then, V_i broadcasts a safety-related message in the form $\{m_i, PID_i, \sigma_i, t_i, vpk_{IDi}, Q_{IDi}\}$ to its surrounding RSUs and VCs. In the protocol developed by Ming et al. [35], the *pk* is calculated using $x_i \in Z_q^*$. Then, V_i broadcasts the safety-related message in the form $\{M_i, PID_i, t_i, P_i, D_i, R_i, \sigma_i\}$ to its surrounding *RSUs*. In [36], a CL-based batch verification protocol was designed, where VC chooses $x_{IDi} \in Z_q^*$ and calculates vpk_{IDi} as follows: $vpk_{IDi} = x_{IDi} \cdot P$ during the *Vehicle-Key-Generation* phase. Then, V_i broadcasts the safety-related message in the form of $(ID_i, vpk_{IDi}, M_i, R_i, t_i, \sigma_i)$ to its surrounding RSUs. In [37], a CL-based aggregate signature protocol is developed and is resistant against attacks from a unauthorized but passive KGC. During the *UserKeyGen* phase, the entity ID_i picks $x_{IDi} \in Z_q^*$ and calculates upk_{IDi} as follows: $upk_{IDi} = x_{IDi} \cdot P$. After that, the transmitter sends the CL-based signature in the following format (U_i, V_i) to the surrounding nodes. In the protocol mentionned in [63], a CL-based signature is constructed to develop a remote checking protocol, that aims to verify the integrity of shared data. At the *SecretValueGen* and *PublicKeyGen* phases, a user selects $x_i \in Z_q^*$ as its secret value and defines its key value as follows: $S_i = x_i$. In this way, the user deploys its secret key to calculate PK_i as follows: $PK_i = g^{Si} = g^{xi}$. However, in these existing CL-based protocols, a VC selects its *x* and calculates its *pk*. As a result, this method of computation has some security limitations. In other words, the non-repudiation between the authority and node is not garanteed. Plus, the *pk* remains susceptible to a replacement by an adversary as it is not tied to an identity. As a result, the TA cannot use a *pk* to identify a node. In addition, the authority can only identify an unauthorized node using its pseudonym. As a result, the existing protocols can be classified as TL2 protocols that use ECC. As for our research work, we have developed three protocols based on ECC with a higher security strength:

1. The first protocol: a CL-based protocol that meets a TL 3 and fulfills the security requirements known in VANET. During the authentication process, our protocol uses ECDSA and elliptic curves. In addition, it allows to carry out a batch verification:
2. A second protocol: a CL-based protocol which is based on Schnorr algorithm and elliptic curves. This protocol allows to aggregate signatures received from different nodes into a compact single signature as part of the authentication process. Moreover, it enables VCs to perform MultiSig by efficiently verifying signatures for a single message and also batch verification of multiple signatures for different safety-related messages. Our Schnorr-based protocol ensures the uniqueness of the *pk* and non-repudiation between the authority and a node in the network. In addition, we show that the protocol achieves a TL3.
3. A third protocol: a CL-based protocol based on hybrid cryptography and elliptic curves is developed. This protocol combines both symmetric and asymmetric encryption during the authentication process. Furthermore, this protocol ensures the uniqueness of the *pk* and non-repudiation between the authority and a node in the network. In addition, we show that it satisfies aTL3.

CL-based protocol based on pairing. The protocols mentioned in [38], [39], [40], [41], [64], [65], [66], [67], [68], [69], [70] use pairing during the authentication process with a TL2. In case an adversary intercepts the *psk*, it can generate its own *pk* and replace the original one. Additionally, the authority can also impersonate the identity of a legitimate VC and generate a false *pk*, as indicated in Table 1. Therefore, these protocols do not provide strong non-repudiation between the authority and node in the network, given that the *pk* cannot pass the authority check. In [40], Malhi and Batra propose a CLAS protocol where a VC computes its *pk* during the *public-key-generation* phase. In this protocol, the VC selects $x_i \in Z_q^*$ as its secret key, then calculates its P_i as follows: $P_i = x_i \cdot P$, then broadcasts the safety-related message in the form of $(M_i, PS_j, PS1_j, P_i, \sigma_i = (U_i, V_{ijk}))$ to the surrounding RSUs. In [38], Horng et al. propose a CLAS protocol with anonymity preservation, where the VC chooses its own *pk* using the *psk* provided by the authority during the *public-key-generation* phase. In this protocol, VC chooses $x_{IDi} \in Z_q^*$ as its secret key $x_{Idi}=vsk_{IDi}$ during the *key-generation* phase, and calculates vpk_{IDi} by itself as follows: $vpk_{IDi} = x_{IDi} \cdot P$. Then, V_i transmits the safety-related message in the form of $(ID_i; vpk_i; M_i; t_i; \sigma_i)$ to the surrounding RSUs. In [39] , Li et al. provide a cryptanalysis of an existing protocol in the literature and propose a new CLAS protocol, where the VC computes its *pk* after receiving a *psk* from the authority during the *public-key generation* phase. In this protocol, a VC selects $x_{IDi} \in Z_q^*$ and generates vpk_{IDi} as follows: $vpk_{IDi} = x_{IDi} \cdot P$. Subsequently, V_i broadcasts the safety-related message in the form $(ID_i, vpk_{Idi}, M_i, t_i, \sigma_i)$ to its surrounding RSUs. In [71], Kumar and Sharma propose a CLAS protocol that satisfies a TL 2, where a VC generates its *pk* after receiving a *psk* during the *public-key-generation* phase. In [64], Cui et al. propose a CLAS protocol where a VC chooses a *pk* during the *public-key-generation* phase, after receiving a pseudonym and a *psk* during the *pseudonym* and *partial-private-key-generation* phases. In [41], Zhong et al. propose a protocol that ensures authentication, aggregation, and anonymity preservation. During the *public-key-generation* phase, a VC chooses a *pk*. In [65], Kamil and Ogundoyin propose a new CLAS protocol where the VC first selects a x_i using its *psk* and some random numbers after receiving a *psk*, and computes a *pk*. In [66], Zhao et al. propose a CLAS protocol in the context of the Internet of vehicles, where the VC chooses a key pair using its pseudonym and *psk*. In [67], Kamil and Ogundoyin propose a CLAS protocol that guarantees authentication and anonymity preservation, where the VC chooses a *pk* during the *public-key-generation* phase after obtaining a pseudonym and a *psk* from the authority. In[68], Li et al. propose a CLAS protocol where the VC chooses a *pk* during the *public-key-generation* phase. Afterward, the *pk* is obtained by combining the *psk* and x_i to complete the *sk*. In Mei et al. protocol, a CLAS is developed that ensures conditional authentication with anonymity preservation in the context of the Internet of VCs, where a VC chooses a *pk* during the *public-key-generation* phase. In [70], Thumbur et al. propose a CLAS protocol that satisfies authentication in VANET, in which the VC chooses a x_i and then calculates its *pk* using its *x* and *psk* during the *public-key-generation* phase. In [72], Cahyadi et al. propose a CLAS protocol that ensures anonymity and security, in which the VC chooses its *x* and *pk* during the *public-key-generation* phase. In Zhong et al.'s protocol [41], VC selects $x \in Z_q^*$ during *public-key-generation* and calculates the *pk* as $vpk_i = x_i \cdot P$, then broadcasts the safety-related message in the form of $\{PID_i, m_i, vpk_i, t_i, \sigma_i\}$ to its surrounding RSUs.

Table 1. Review of CLAS protocols in the literature

Authors	Algorithms used in each scheme	Generation phase of the pk	TL
Malhi et Batra [40]	*1-Setup, 2-Registration, 3-Partial-Private-Key-Gen, 4- UserKeyGen, 5-Pseudonym-Gen, 6-Sign/aggregate, 7-Verify /Aggregate-Verify.*	During *UserKeyGen*, a VC with the identity Q_{ID_i} selects $x_i \in Z_q^*$ and sets x_i and its P_i as follows: $P_i = x_i \cdot P$	2
Horng et al. [38]	*1-Setup, 2- Pseudo-Identity-Generation, 3-Partial-Private-Key-Extraction, 4- Vehicle-Key-Generation, 5-Sign/ Aggregate, 6-Individual Verify/ Aggregate-Verify.*	During *Vehicle-Key-Generation*, a VC chooses a $x_{ID_i} \in Z_q^*$ and sets its secret key $x_{ID_i}=vsk_{ID_i}$. Then, it computes $vpk_{ID_i}= x_{ID_i} \cdot P$ as its pk.	2
Li et al. [39]	*1-Setup, 2- Pseudo Identity Generation/Partial Private Key Extraction, 3- Vehicle Key Generation, 4-Sign /Aggregate, 5-Individual Verify/ Aggregate-Verify.*	During *Vehicle Key Generation*, a VC randomly selects $x_{ID_i} \in Z_q^*$ and chooses $x_{ID_i}= vsk_{ID_i}$, and calculates *pk* as follows : $vpk_{ID_i} = x_{ID_i} \cdot P$.	2
Kumar et Sharma [71]	*1-Setup, 2-Partial-Private-Key-Gen, 3-UserKeyGen, 4-Pseudonym-Gen, 5-Sign /Aggregate, 6-Verify/ Aggregate -Verify.*	During *UserKeyGen*, a VC with an identity Q_{ID_i} chooses a random number $x_i \in Z_q^*$ and computes its P_i as follows: $P_i = x_i \cdot P$.	2
Cui et al. [64]	*1-Setup, 2-Pseudo-Identity-Generation/ Partial-Private-Key-Extraction, 3-Vehicle-Key-Generation, 4-Sign /Aggregate, 5-Individual Verify / Aggregate-Verify.*	During *Vehicle-Key-Generation*, a VC chooses $x_{ID_i} \in Z_q^*$ as its secret key vsk_{ID_i}, and outputs its $vpk_{ID_i}= x_{ID_i} \cdot P$.	2
Zhong et al. [41]	*1-Setup, 2- Pseudonym generation, 3- Partial key generation, 4- Vehicle key generation, 5-Sign / Aggregate, 6-Verify /Aggregate-Verify.*	During *Vehicle key generation*, a VC chooses $x_i \in Z_q^*$ and sets x_i and sets its vpk_i as follows: $vpk_i = x_i \cdot P$	2
Kamil et Ogundoyin [65]	*1-Setup, 2-UserRegistration, 3-PartialSecretKeyGeneration, 4-Pseudonym-Gen, 5-UserKeyGeneration, 6-Sign /Aggregate, 7-Verify/ Aggregate-Verify.*	During *UserKeyGeneration*, a VC with an identity ID_k generates its private key SK_k as follows: • It randomly chooses $a_k, r_k^1, r_k^2 \in Z_q^*$ and calculates $SK_k^1 = h_3 (r_k^1 \| A_k \| PID_{y,k})$, $SK_k^2 = h_3(r_k^2 \| A_k \| PID_{y,k})$ and (A_k, x_k) is the psk. • It outputs $SK_k= a_k (SK_k^1+ SK_k^2)$ and $PK_k = SK_k \cdot P$ as its private and *pk*s respectively.	2
Zhao et al. [66]	*1- Setup, 2- RegistVehicle, 3-GeneratePseudonym, 4-GeneratePartialPrivateKey, 5-GenerateVehicleKey, 6-Sign /Aggregate, 7-Verify/ Aggregate-Verify.*	During *GenerateVehicleKey*, a VC with a pseudo identity $PID_{y,\lambda}$ generates its private key SK_λ and PK_λ as follows: • Randomly picks $a_\lambda, r_\lambda^1, r_\lambda^2 \in Z_q^*$ • Calculates $SK_k^1 = h_3 (r_\lambda^1 \| A_k \| PID_{y,\lambda})$ • $SK_k^2 = h_3 (r_\lambda^2 \| k_\lambda \| PID_{y,\lambda})$. • Outputs $SK_k= a_k (SK_k^1+ SK_k^2)$ and $PK_\lambda= SK_\lambda \cdot P$ and (A_k, k_λ) is the psk.	2
Kamil et Ogundoyin [67]	*1-Setup, 2- Pseudonym generation, 3- Partial-key generation, 4- Vehicle key generation, 5-Sign / Aggregate, 6-Verify/Aggregate-Verify.*	During *Vehicle key generation*, V_i selects $x_i \in Z_q^*$, sets x_i as its vsk_i, and generates its vpk_i as: $vpk_i = x_i \cdot P$.	2
Li et al. [68]	*1-Setup, 2- Pseudonym generation, 3- Partial key generation, 4- User key generation, 5-Sign /Aggregate, 6-Verify /Aggregate-Verify.*	During *User key generation*, a VC chooses $x_i \in Z_q^*$ and calculates the corresponding *pk* as follows: $P_i = x_i \cdot P$. The vehicle's private key is $S_i = (d_i, x_i)$ and d_i is the *psk*.	2

Mei et al. [69]	*1- Setup, 2-Pseudonym-Gen, 3-Partial-Private-Key-Gen, 4-Vehicle-Key-Gen, 5-Sign /Aggregate, 6-Verify/Aggregate-Verify.*	During *Vehicle-Key-Gen*, a VC selects $x_i \in Z_p^*$ and sets $x_i = vesk_i$. Then, V_i outputs $vepk_i = x_i \cdot P$ as its pk.	2
Thumbur et al. [70]	*1-Setup, 2- Pseudo Identity Generation, 3-Partial Key Generation, 4- Set Secret Value, 5-Vehicle Key Generation, 6-Sign/ Aggregate, 7-Verify/ Aggregate-Verify.*	During *Set Secret Value*, a VC chooses $vsk_{PID_i} \in Z_q^*$ as its secret value and computes $X_i = vsk_{PID_i} \cdot P$ During *Vehicle Key Generation*, a V_i creates its own pk as follows: V_i computes $h_{2i} = H_2(PID_i, X_i)$, $Q_i = R_i + h_{2i} \cdot X_i$, and X_i is its secret which is randomly generated. Then, V_i computes its *pk* as $vpk_{PID_i} = (Q_i, R_i)$. Then, the sk as follows: $VSK_{PID_i} = (vpk_{PID_i}, vsk_{PID_i})$, and R_i is part of the psk.	2
Cahyadi et al. [72]	*1-Setup, 2-Pseudonym-Gen, 3-Partial-Private-Key-Gen, 4-Vehicle-Key-Gen, 5-Sign /Aggregate, 6-Verify /Aggregate-Verify.*	During *Vehicle-Key-Gen*, a VC selects $x_i \in Z_q^*$ as its private key vsk_{ID_i} in which $vsk_{ID_i} = x_i$. Then it uses a generator point *P* to compute its $vpk_{ID_i} = x_i \cdot P$.	2
Our CLAS scheme	*1-Setup, 2- Vehicle-Key-Gen, 3- Pseudonym-Gen, 4-Partial-Private-Key-Gen, 5-Sign/ Aggregate, 6-Verify/ Aggregate-Verify.*	The phase *Vehicle-Key-Gen* is carried out at an earlier stage after *setup*. In addition, the TA binds the *pk* to a pseudonym using a hash function during *Partial-Private-Key-Gen* phase.	3

2.7.4 Conclusion

According to the bibliographic research conducted in the context of VANET, it turns out that the existing CL-based protocols which use pairing satisfy TL2. To achieve a protocol with TL3, we propose new CL-based protocols that use ECC and pairing, meet TL3 and also fulfill security requirements in VANET. Furthermore, our protocols link the *pk* to the VC's identity, where the *Vehicle-Key-Gen phase* is performed before the phases related to *pseudonym* and *partial-private-key generation*. As a result, our new order of algorithms ensures the uniqueness of the *pk*.

On the one hand, a VC cannot replace one *pk* by another since it is unique. Plus, only a certified *pk* by the authority can pass the authentication process. On the other hand, the authority will be held responsible if it replaces an original *pk* because the presence of two correct *pk*s for the same pseudonym implies that the authority has generated two *psk*s. As a result, the authority will only be designated as responsible during a dispute. Consequently, our protocols guarantee non-repudiation between the authority and a network node and also satisfy TL3.

CHAPTER 3: CL-based Protocol with ECDSA AUTHENTICATION

In this Chapter, we introduce our ECDSA-based scheme, as well as an improved version which is ECDSA*-based scheme [9]. We begin by giving an introduction of the context of application of our schemes. In the section 2, we explain the challenge faced in VANET and how our schemes can counter the present vulnerabilities. In the section 3, we outline the security objectives of the main scheme ECDSA-based. The section 4 explains how nodes can use those protocols in the network, while the section 5 describes the system model, and the section 6 details the algorithm used. We provide an overview and cryptographic operations details of our schemes in the sections 7 and 8, respectively. Additionally, we present a security proof and analysis in Sections 9 and 10. A simulation was performed, as described in the section 11, and we conclude the chapter in Section 12.

3.1 Introduction

A CCPPA comprehensively addresses all security requirements and satisfy both security and anonymity concerns within VANET. Nevertheless, many CL-based protocols typically attain a TL 2, as outlined in Girault's hierarchy. In the event of *psk* exposure, an adversary can potentially change the *pk*, as it is not inherently associated with a specific identity. Consequently, the TA's capability is limited to identify unauthorized nodes by their pseudonyms. In our protocol, we use a binding hash technique that aims at elevating the TL of a CL-based protocol from TL 2 to TL 3.

Context related to ECDSA-based and ECDSA-based schemes.* Our research introduces novel CCPPA protocols that incorporate this binding hash technique, thereby achieving a stronger security TL 3. In the event of a *psk* leakage, only a legitimate node can utilize the corresponding *pk*, delivered by the TA. Consequently, our CCPPA protocols turn out resilient against PK-Replacement vulnerability, which is present in several protocols. Furthermore, our protocols introduce an innovative tracing mechanism that empowers the TA to identify an unauthorized node using both its pseudonym and *pk*. Leveraging ECC, our protocols sidestep the need for a MTH function and pairing computations, thereby enhancing computational efficiency. Moreover, we have developed two protocols that are respectively based on ECDSA and ECDSA* algorithms during authentication. A comprehensive security analysis validates the robustness of our two protocols and show that they are EUF-CMA. Notably, ECDSA*-based protocol gains an advantage over ECDSA-based protocol by facilitating verification of signatures by batch. This feature allows RSUs to collect signatures from multiple VCs and carry out a batch verification, significantly reducing verification time in traffic-dense areas. Additionally, we conducted a thorough performance analysis, comparing ECDSA*-based scheme with several known protocols. Simulation results demonstrate that ECDSA*-based scheme outperforms many protocols in terms of both message signing and verification processes. Moreover, ECDSA*-based scheme exhibits lower communication costs when transmitting safety-related messages in comparison to the studied CL-based protocols.

3.2 Problem Statement regarding ECDSA-based scheme

Numerous anonymity-preserving conditional authentication protocols have been developed to address the trade-off between authentication and anonymity using PKC[73], [74]. These CPPA protocols fall into three primary categories: PKI, ID-PKC, CL-PKC [21]. In addition, Girault's classification[13] categorizes PKC approaches based on their security strength as follows:

1. *TL 1:* At this level, the TA possesses knowledge of the *sk* of the nodes.
2. *TL 2:* is attained when the TA cannot generate a node's *sk*. However, the authority has the capability to use the identity of a legitimate node by creating a fraudulent certificate or a false *pk*.
3. *TL 3:* is satisfied when the TA cannot generate a *sk* for a node. In a such case, we can prove that the TA generated a false *pk* if it had performed such an action.

According to our literature review, current anonymity-preserving conditional authentication protocols satisfy TLs 1 and 2. Consequently, they provide lower security compared to a protocol classified under TL 3. Therefore, our research aims to develop protocols called ECDSA-based and ECDSA*-based scheme that meet TL 3.

3.3 Objectives of ECDSA-based scheme

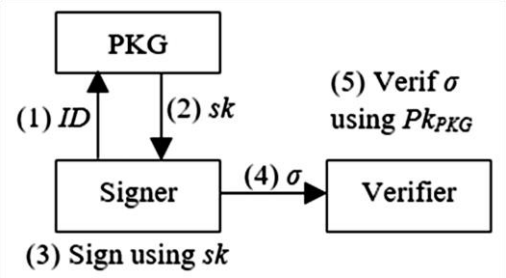
The main idea of this work is to construct protocols that ensure non-repudiation between the authority and a network node. Furthermore, these protocols should be resilient to the PK-Replacement cyberattack by linking a VC's identity to its *pk* via a hash function. In addition, the efficiency of our protocols should be based on the difficulty of solving the ECDL and ECCDH problems. In terms of cryptographic computations, the protocols are based on CL-based and ECC. Plus, they can be classified as PKC with a TL 3. Our protocols aim to achieve the following security points:

1. *TL 3*: Our CL-based protocols should satisfy TL 3. Plus, it provides a resilient strategy against the PK-Replacement attack. In case *psk* is transmitted by KGC and revealed during its transmission, an adversary can carry out this cyberattack, which is likely to happen in a normal CL-based protocol with a TL 2.
2. *Strong Non-Repudiation:* In our protocols, a KGC will be held accountable if it has replaced the original *pk* of an entity with a new *pk*. This is because the presence of two valid *pk*s for the same identity implies the presence of two *psk*s linking the identity to two *pk*s. As a result, we can show that KGC is accountable, as it has generated two correct *psk*s.
3. *Identification of an unauthorized node using two components: pk and Pseudonym:* In many CL-based protocols [34], [35], [38], [39], [40], [41], the identification of an unauthorized node is solely reliant on the pseudonym generated by the TA in VANET. But our protocols introduce an advanced method to identify nodes, not only through their pseudonyms but also via their *pk*s. This enhanced identification strategy ensures a higher level of network security by providing increased confidence in network monitoring.

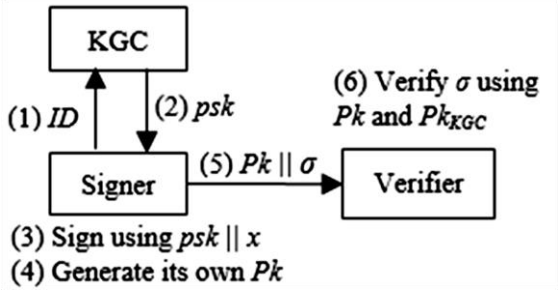
4. *Standardized Algorithm and Batch Verification*: Our protocols facilitate the use of the traditional ECDSA algorithm by network nodes. Furthermore, they exclusively utilize ECC computations, eliminating the need for pairing computations and the MTH function, which often require extended execution times. The ECDSA-based and ECDSA*-based schemes empower a node to generate key pairs in a safe way and sign its safety-related messages by respectively using either ECDSA or ECDSA*. Additionally, these protocols offer the capability to carry out a batch verification for signatures when employing the ECDSA* algorithm.
5. *Security Analysis:* A comprehensive security analysis has been conducted to demonstrate the robustness of our protocols in the random oracle model, considering the complexity of the ECDLP and ECCDHP. Furthermore, our protocols achieve security and anonymity within VANET.
6. *Evaluation analysis:* Our simulation consists of analyzing both execution time and communication cost. In this regard, our simulation results indicate that the ECDSA*-based scheme protocol demonstrates the most efficient execution time for authenticating a single message. Moreover, our protocol incurs lower communication costs compared to the existing CL-based protocols. [34], [35], [38], [39], [40], [41].

3.4 Introduction of our new ECDSA-based and ECDSA*-based schemes

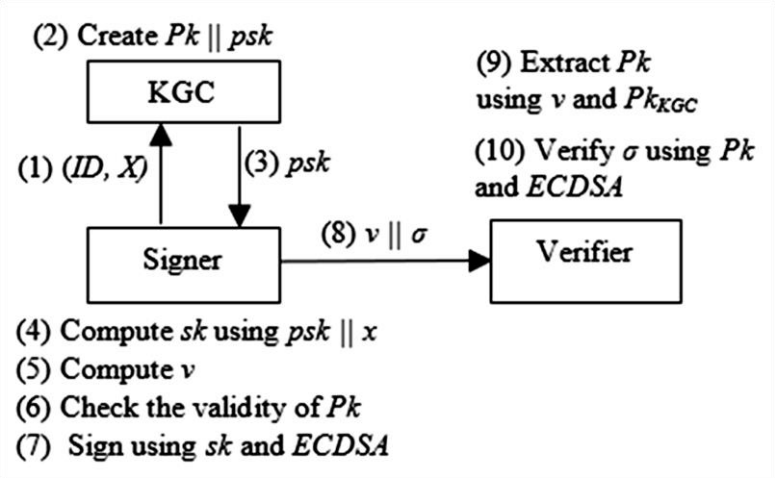
Our two CL-based protocols eliminate the need for certificates and resolve the key escrow problem as illustrated in the figure 5-a. Furthermore, they are resilient against the PK-Replacement attack mentioned in [75], [76], [77] as illustrated in the figure 5-b. Both of our protocols only allow a node to use *pk*s that are confirmed by the TA as illustrated in Figure 5-c and Figure 5-d. Plus, ECDSA*-based scheme allows to carry out a batch verification. [78]. The steps used in both protocols, as depicted in Figure 5-c and Figure 5-d.



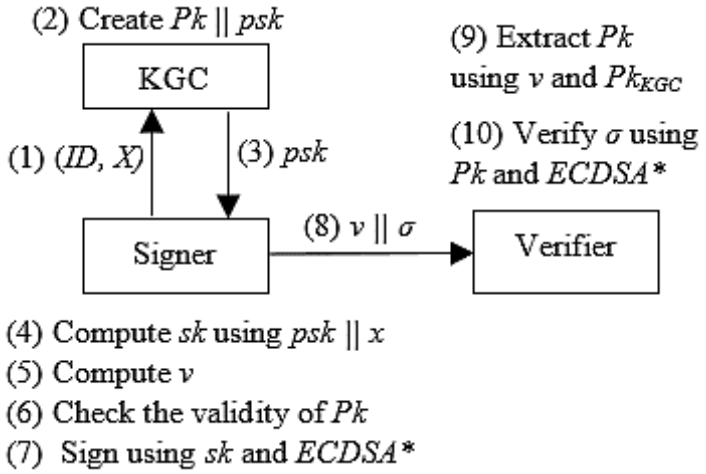
(a) ID-based protocol



(b) Existing CL-based Protocol with TL 2



(c) Our ECDSA-based Protocol with TL 3



(d) Our ECDSA*-based Protocol with TL 3

ID: Node's identity
psk: Signer's *psk*
sk: Signer's *sk*
Pk: Signer's *pk*
x: Signer's *x*
X: ECC point generated by the signer
v: Cryptographic value calculated by the signer
Pk_{PKG}, Pk_{KGC}: *pk*s of PKG and KGC, respectively
σ: Signature

Fig 5. Architecture of ID-based protocol and CL-based protocol

3.5 VANET Model of our ECDSA*-based scheme

Our VANET model can be divided into two layers: an upper layer consisting of three entities: *KGC*, *TRA*, *AS*, and a lower layer comprising VCs that are equipped with an OBU and RSUs. Plus, nodes are assumed to communicate using DSRC protocol, which is identified as IEEE 802.11p.

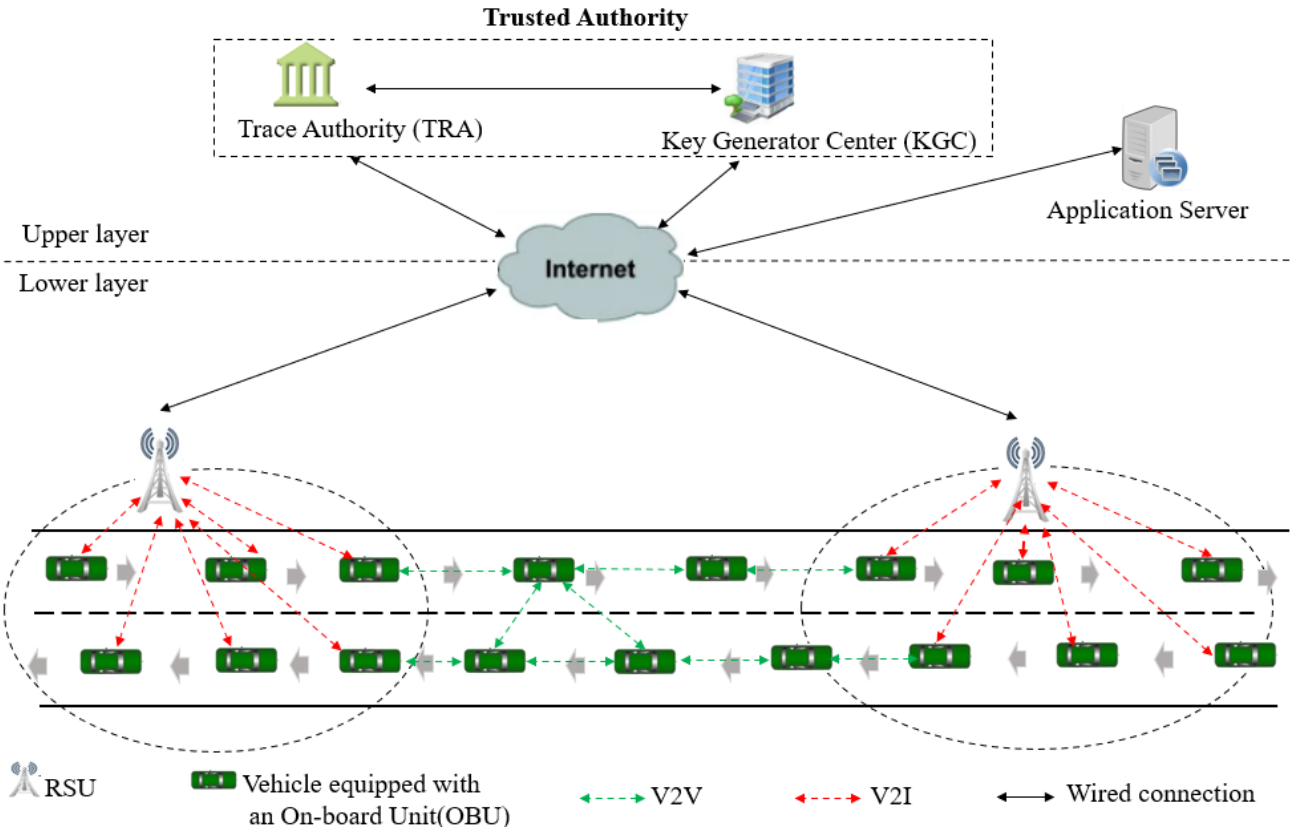


Fig 6. VANET model of our ECDSA*-based protocol

3.6 ECDSA Algorithm

	ECDSA	ECDSA* [47]
Setup	Domain parameters = $\{E(F_p), a, b, n, P, h\}$	Domain parameters = $\{E(F_p), a, b, n, P, h\}$
KeyGen	$s_k \in Z^*_{n-1}$	$s_k \in Z^*_{n-1}$
	$Pk = s_k \cdot P$	$Pk = s_k \cdot P$
Sign	$k \in Z^*_{n-1}$	$k \in Z^*_{n-1}$
	$R = kP$	$R = kP$
	$r = x(R) \ (mod\ n)$ where $x(R)$ is x-coordinate of R.	$r = x(R) \ (mod\ n)$ where $x(R)$ is x-coordinate of R.
	$s=k^{-1}\,(h(m) + s_k \cdot r) \ (mod\ n)$	$s=k^{-1}\,(h(m) + s_k \cdot r) \ (mod\ n)$
	$\sigma=(r,s)$	$\sigma=(R,s)$
Verify	$w = s^{-1} \ (mod\ n)$	$w = s^{-1} \ (mod\ n)$
	$u= h(m)w \ (mod\ n)$	$u= h(m)w \ (mod\ n)$

$u' = rw \pmod n$	$u' = rw \pmod n$
$R \cdot y \leftarrow$ square root method from the received r value	
$R = u \cdot P + u' \cdot Pk \quad \in E(F_p)$	$R = u \cdot P + u' \cdot Pk \quad \in E(F_p)$
$true \leftarrow Verify\ (Pk, \sigma, m)$ if $x(R) = r \pmod n$	$true \leftarrow Verify\ (Pk, \sigma, m)$ if $x(R) = r \pmod n$

Fig 7. Description of the ECDSA and ECDSA* algorithms

3.7 Overview ECDSA-based scheme

Our ECDSA-based scheme can be classified as a CL-based protocol according to the definition given in [79]. In this section, we give the notations and descriptions of each phase used in our ECDSA-based scheme, as mentioned in Table 2 and Figure 8.

Table 2. Notations used in ECDSA-based and ECDSA*-based protocol

Symbol	Description
(s_{TRA}, Pk_{TRA})	TRA's key pair
(s_{KGC}, Pk_{KGC})	KGC's key pair
psk_i	Node i's psk
sk_i	Node i's sk
Pk_i	Node i's pk
X_i	Point in ECC
v_i	Value created by a node
RID_i	True identity of a node
$PID_{i,1}$	Pseudonym created by a node
$PID_{i,2}$	Pseudonym created by TRA for a node
ID_i	Identity of a node created by itself
PID_i	Pseudonym created by TRA for a node
H_1	$\{0, 1\}^* \times G \longrightarrow Z_q^*$: Hash function
H_2	$\{0, 1\}^* \longrightarrow Z_q^*$: Hash function
T_i	Lifetime of a pseudonym
t_i	Timestamp

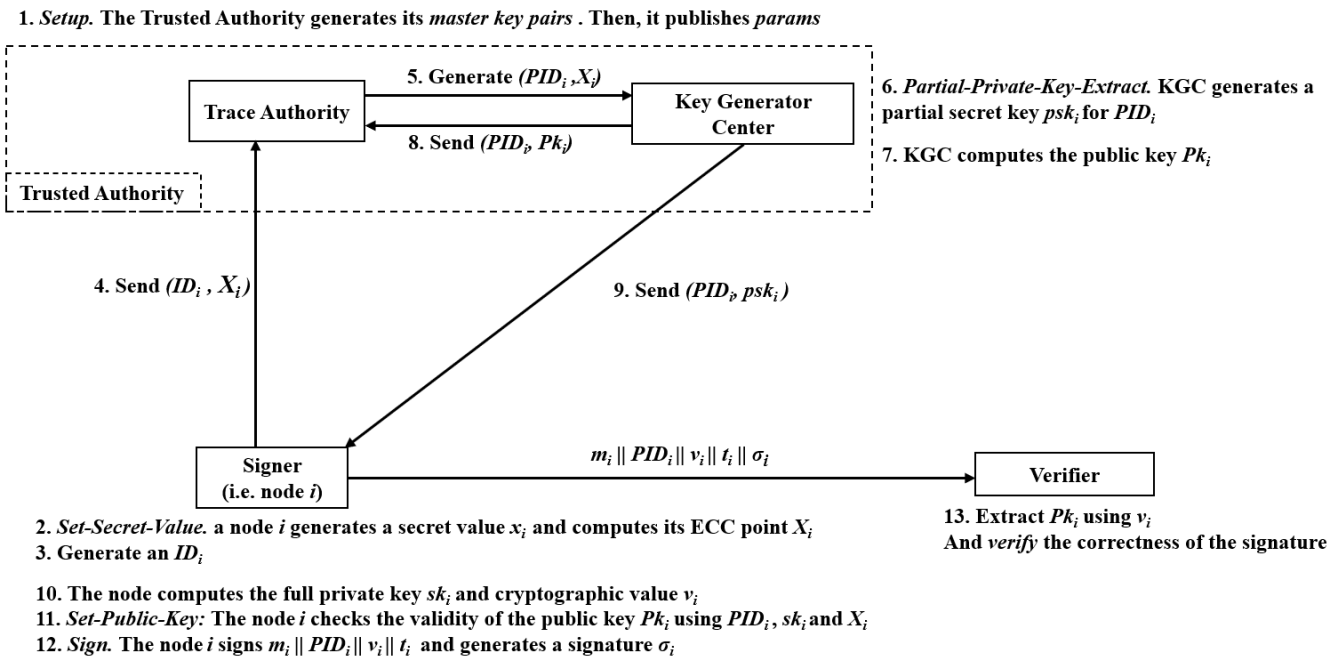


Fig 8. Overview of our ECDSA-based Protocol

3.8 Cryptographic computations in ECDSA-based and ECDSA*-based schemes

3.8.1 Cryptographic computations in ECDSA-based scheme

In this section, the computation of each step used in ECDSA-based scheme is described in Figure 9.

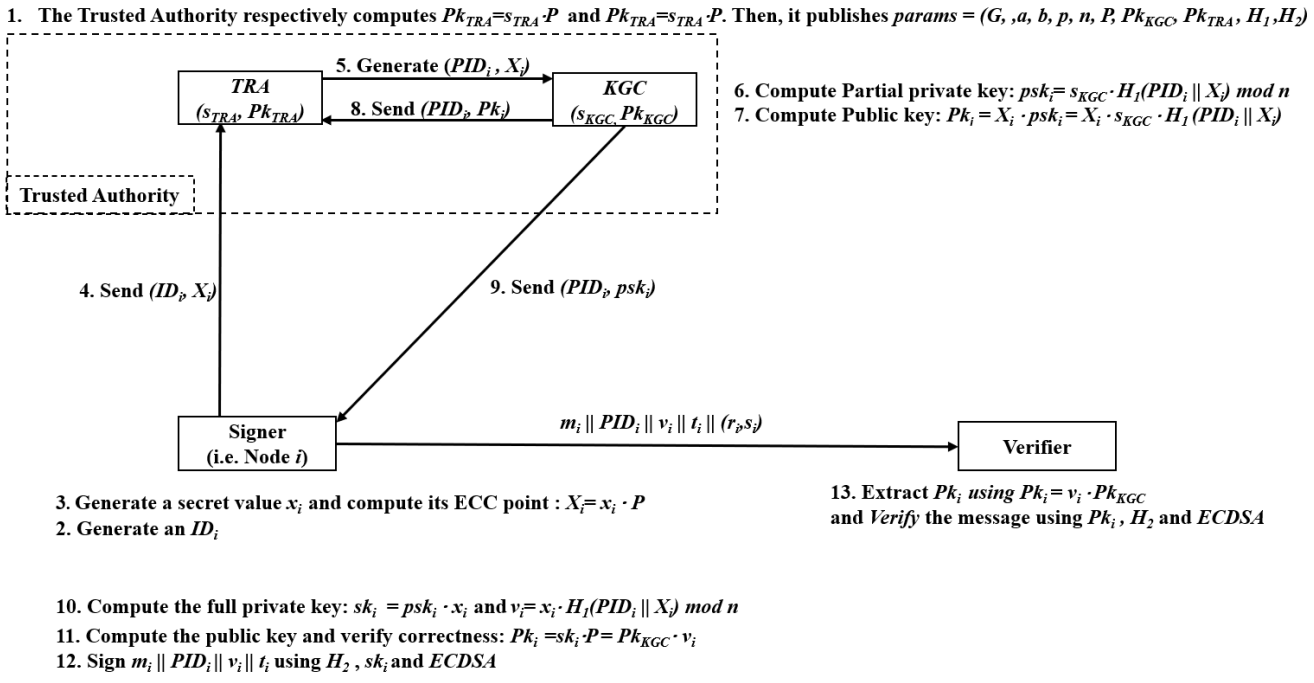


Fig 9. Cryptographic Computations in ECDSA-based scheme

Batch Signature Verification:

Two protocols have been developed: ECDSA-based and ECDSA*-based schemes. The first protocol can be used to generate an ECDSA signature, while the second protocol, which is ECDSA*-based scheme, can be used to sign a safety-related message via ECDSA*, which is a modified version of ECDSA as described in Figure 7. In our research work, ECDSA*-based scheme enables batch verification, that results in reducing verification time. In our work, a RSU can employ a batch verification due to its higher performance and coverage radius compared to an OBU [78].

3.8.2 Cryptographic computations of ECDSA-based scheme*

The ECDSA-based scheme allows the generation of a signature (r_i,s_i) by using a key pair via an ECDSA algorithm. In the same way, this key pair can also be used to generate a signature (R_i,s_i) via ECDSA* algorithm. The protocol ECDSA*-based scheme allows to carry out a batch verification, where two cases are analyzed during the reception of messages:

1. A receiver verifies multiple safety-related messages from a "single signer".
2. A receiver verifies multiple safety-related messages from "multiple signers" by using a batch verification to verify *t* signatures as mentioned in [80]. For instance, to verify safety-related messages in the form $\{m_i, PID_i, vi, (R_i,s_i)\}$, it selects pairwise primes $b_1, b_2 \cdots b_i < n$, where $b_i \in F_p$. Then, the verifier can use the equations below:

Verification of t safety-related messages received from the same signer:

$$\sum_{i=1}^{t} R_i\, b_i = \left(\sum_{i=1}^{t}(b_i\, u_i)(mod\ n)\right)\cdot P + \left(\sum_{i=1}^{t}(b_i\, u_i')(mod\ n)\right)\cdot Pk$$

$$= \left(\sum_{i=1}^{t}(b_i\, u_i)(mod\ n)\right)\cdot P + \left[\sum_{i=1}^{t}(b_i\, u_i')\ (mod\ n)\right]\cdot v\cdot Pk_{KGC}$$

Where *Pk:* public key.

PID: pseudonym.

v : cryptographic value.

Verification of t safety-related messages received from different signers:

$$\sum_{i=1}^{t} R_i\, b_i = \left(\sum_{i=1}^{t}(b_i\, u_i)(mod\ n)\right)\cdot P + \left(\sum_{i=1}^{t}(b_i\, u_i'))(mod\ n)\right)\cdot Pk_i$$

$$= \left(\sum_{i=1}^{t}(b_i\, u_i)(mod\ n)\right)\cdot P + \left[\sum_{i=1}^{t}(b_i\, u_i')(mod\ n)\cdot v_i\right]\cdot Pk_{KGC}$$

Where Pk_i: public keys.

PID_i: pseudonyms.

v_i : cryptographic values.

3.9 Security proof of ECDSA-based and ECDSA*-based schemes

The concept of security in digital signature schemes was initially introduced by Goldwasser, Micali, and Rivest. While various security objectives exist for such schemes, the most widely adopted model aims to prevent existential forgery (EUF).

In this context, EUF-CMA (Existential Unforgeability under Chosen Message Attack) is a security standard that ensures signatures cannot be forged for new messages, even when an attacker has access to a set of previously signed messages and their corresponding signatures.

The security of the ECDSA-CCPPA scheme is demonstrated through proofs of its existential unforgeability against two types of advanced adversaries, referred to as super types I and II [9]. These proofs rely on the assumed computational difficulty of solving two fundamental problems in elliptic curve cryptography: the Elliptic Curve Computational Diffie-Hellman Problem (ECCDH) and the Elliptic Curve Discrete Logarithm Problem (ECDL).

This approach to proving security ties the scheme's resistance to forgery to well-established hard problems in cryptography, providing a strong theoretical foundation for its security claims.

3.10 Security analysis of ECDSA-based and ECDSA*-based schemes

The results related to the security requirements in our ECDSA*-based scheme can be defined in Table 3.

Table 3. Security requirements of ECDSA*-based scheme in VANET

	ID-Based schemes			CL schemes						
	[16]	[32]	[33]	[38]	[39]	[40]	[41]	[34]	[35]	ECDSA*-based scheme
TL	1	1	1	2	2	2	2	2	2	3
Security of TRA's database	×	×	×	×	×	×	×	×	×	●
Authentication of the transmitted message	●	●	●	×	●	●	●	●	●	●
Protection of identity	●	●	●	●	●	●	●	●	●	●
Traceability of a unauthorized node	●	●	●	●	●	●	●	●	●	●
Unlinkability of pseudonyms	●	●	●	●	●	●	●	●	●	●
Independency between authorities	×	●	×	●	●	×	●	●	●	●
Resiliance against the escrow problem	×	×	×	●	●	●	●	●	●	●
Resiliance against replay attack	●	●	●	●	●	×	●	●	●	●
Resiliance against modification of message attack	●	●	●	×	●	×	●	●	●	●
Resiliance against impersonation attack	●	●	●	×	●	×	●	●	●	●
Resiliance against MITM attack	●	●	●	×	●	×	●	●	●	●
True non-repudiation	×	×	×	×	×	×	×	×	×	●
Resiliance against PK-Replacement attack	●	●	●	×	×	×	×	×	×	●

3.11 Simulation and performance evaluation of ECDSA*-based scheme

a. Calculation Time

The simulation is performed using MIRACL cryptographic library. It is assumed that a nonce is computed via CSPRNG [81], and its computation can be ignored according to the computation mentioned in [82]. The evaluation includes ECDSA*-based scheme and the protocols [16], [32], [33], [34], [35], [38], [39], [40], [41]. The Table 4 gives time related to a single signature and verification. Plus, the Table 5 gives time to perform *t* signatures and verifications, along with a batch verification. The execution times T_{m-ecc}, T_{bp-m}, $T_{MTP,}$ T_H respectively stand for a scalar multiplication in ECC operation, scalar multiplication in BP operation, a map-to-point hash function operation, and ordinary hash function.

Table 4. Execution time of an authentication for a single message in ECDSA*-based scheme

Protocol	Signing process	Verification process	Authentication cost (ms)
[16]	$3T_{m-ecc}$	$3T_{m-ecc}$	2.652
[32]	$2T_{m-ecc}$	$3T_{m-ecc}$	2.21
[33]	$2T_{m-ecc}$	$4T_{m-ecc}$	2.652
[38]	$2T_{bp-m}$	$3T_{bp} + T_{bp-m} + T_{MTP}$	22.166
[39]	$2T_{bp-m} + T_{MTP}$	$3T_{bp} + T_{bp-m} + 2T_{MTP}$	30.978
[40]	$4T_{bp-m}$	$3T_{bp} + 3T_{bp-m}$	24.596
[41]	$3T_{bp-m}$	$3T_{bp} + T_{MTP} + 2T_{bp-m}$	25.584
[34]	$1T_{ecc-m}$	$3T_{ecc-m}$	1.7734
[35]	$3T_{m-ecc}$	$4T_{m-ecc}$	3.094
Our protocol	$1T_{m-ecc}$	$3T_{m-ecc}$	1.768

Table 5. Execution time of an authentication for *t* messages in ECDSA*-based scheme

Scheme	Signing phase (ms)	Verification phase (ms)
[16]	$3t \cdot T_{m-ecc}$	$t \cdot T_{m-ecc} + 2 \cdot T_{m-ecc}$
[32]	$2t \cdot T_{m-ecc}$	$t \cdot T_{m-ecc} + 2 \cdot T_{m-ecc}$
[33]	$2t \cdot T_{m-ecc}$	$(2t + 2) \cdot T_{m-ecc}$
[38]	$2t \cdot T_{bp-m}$	$3 \cdot T_{bp} + t \cdot T_{bp-m} + t \cdot T_H$
[39]	$t \cdot (2T_{bp-m} + T_{MTP})$	$3 \cdot T_{bp} + t \cdot T_{bp-m} + (t + 1) \cdot T_{MTP}$
[40]	$4t \cdot T_{bp-m}$	$3 \cdot T_{bp} + 3t \cdot T_{bp-m}$
[41]	$3t \cdot T_{bp-m}$	$3T_{bp} + t \cdot T_{MTP} + 2t \cdot T_{bp-m}$
[34]	$t \cdot T_{ecc-m}$	$(t + 2) \cdot T_{ecc-m}$
[35]	$3t \cdot T_{m-ecc}$	$(2t + 2) \cdot T_{m-ecc}$
ECDSA*-based scheme	$t \cdot T_{m-ecc}$	$(t + 2) \cdot T_{m-ecc}$

In addition, the Figures 10 reflects the results from Table 4.

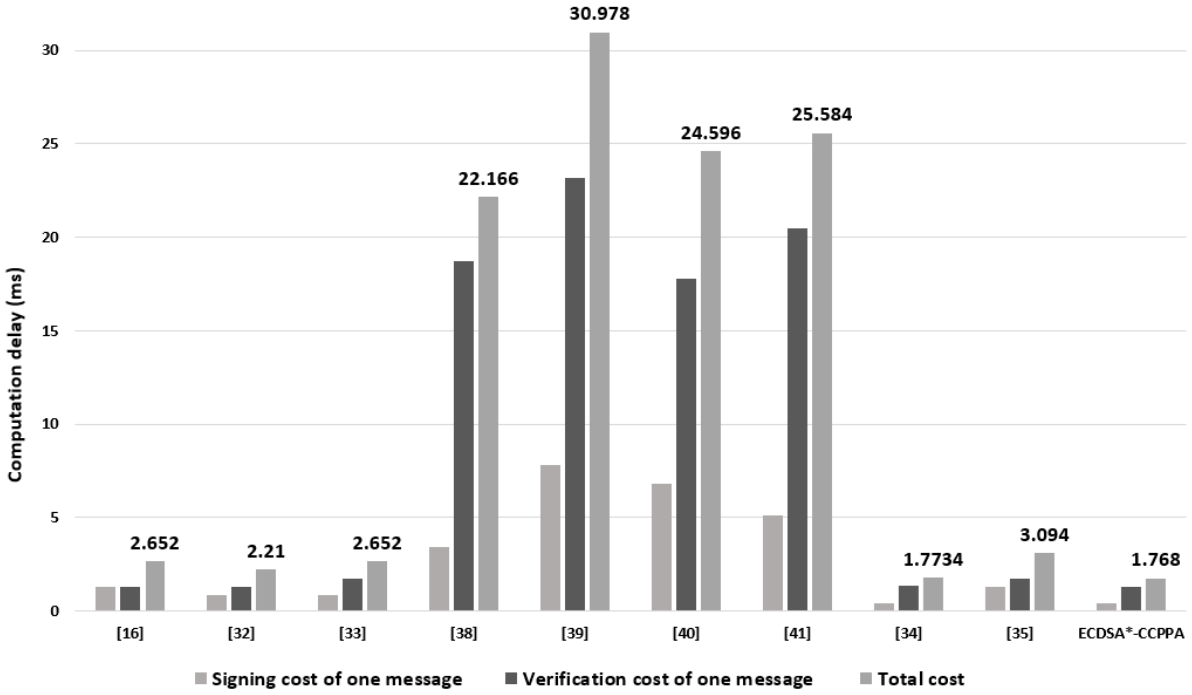


Fig 10. Cost of authentication for a single message in ECDSA*-based scheme

Besides, the Figures 11 and 12 reflect the results from Table 5.

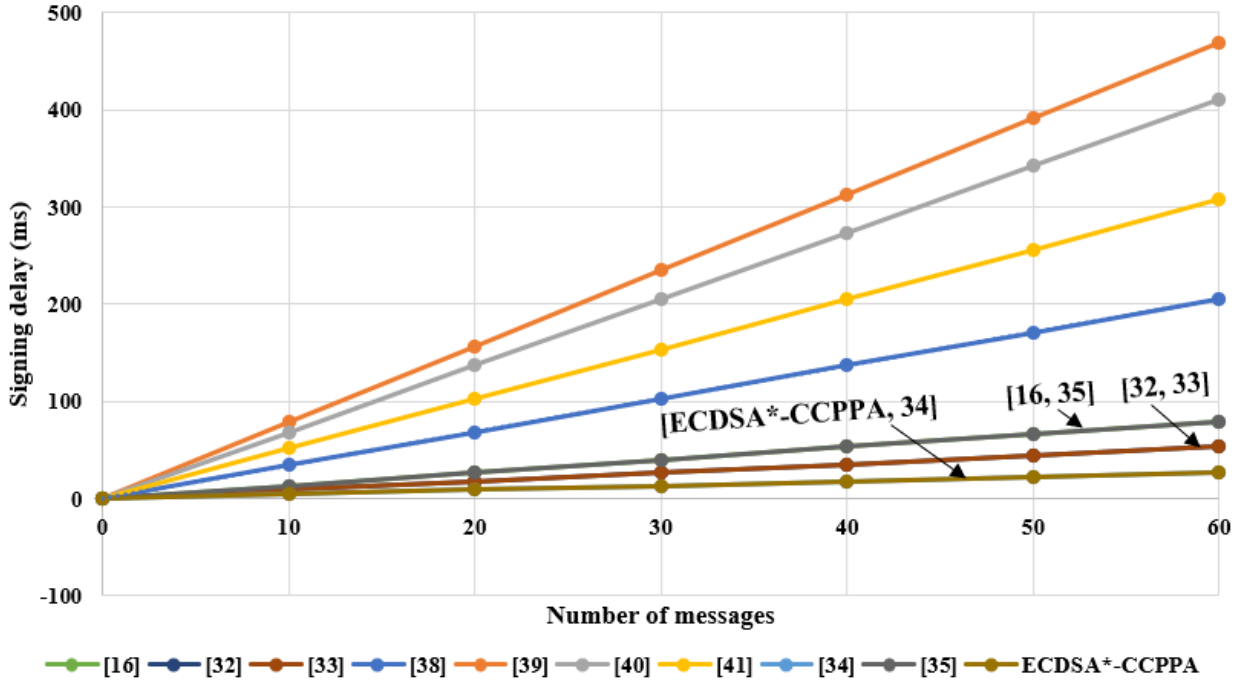


Fig 11. Signing cost for multiple messages in ECDSA*-based scheme

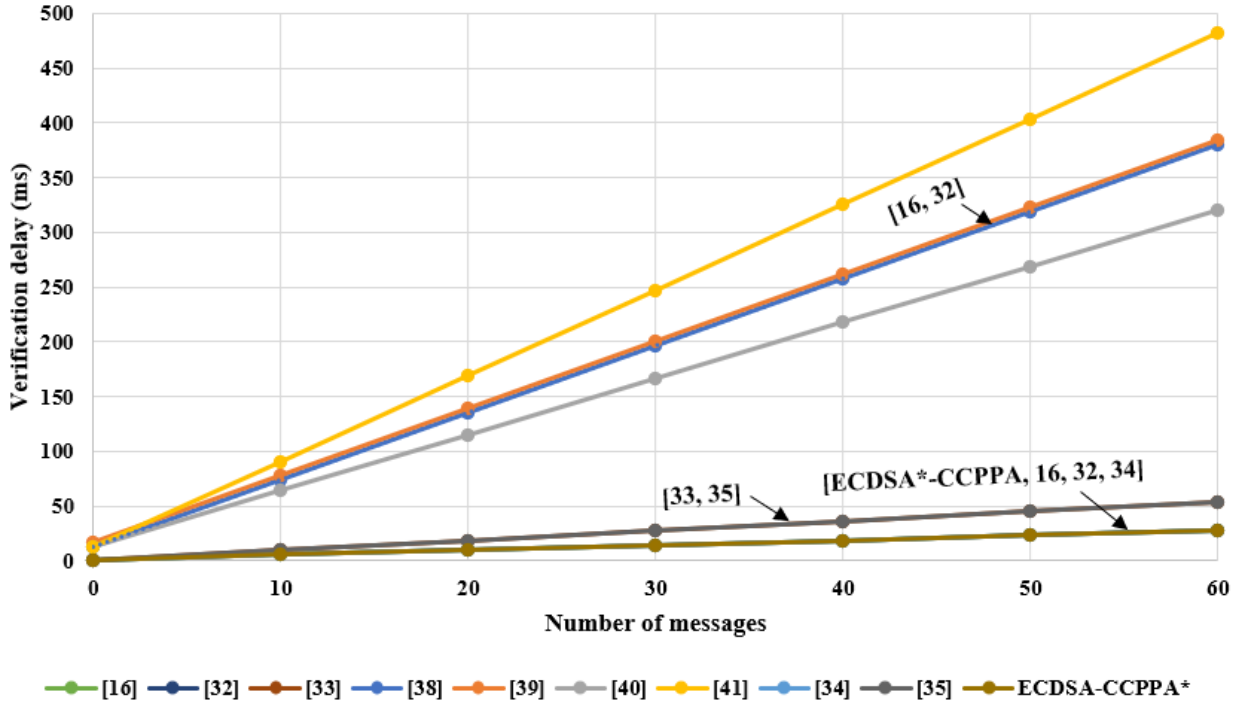


Fig 12. Verification cost for multiple messages in ECDSA*-based scheme

b. Communication Cost

The figure 13 shows that ECDSA*-based scheme outperforms the protocols mentioned in [32], [33], [34], [35], [38], [39], [40], [41], and also requires more communication cost than [16]. However, [16] is classified as an ID-based PKC protocol with a limited security, ECDSA*-based scheme outperforms the CL protocols [34], [35], [38], [39], [40], [41] in terms of better performance.

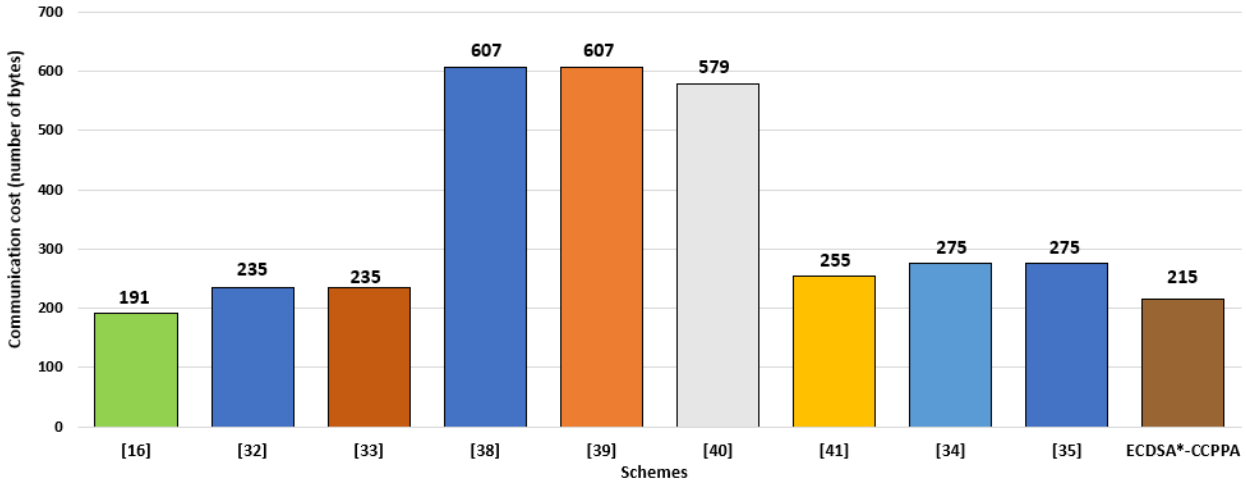


Fig 13. Communication cost in our ECDSA*-based scheme

Even though the protocol [16] incurs lower communication cost compared to the other protocols, it is considered an ID-based protocol and does not achieve a high security strength. Within the group of CL-based protocols [34], [35], [38], [39], [40], [41], our ECDSA*-based scheme

achieves the best communication overhead. For 30,000 messages, ECDSA*-CCPA achieves 15.6% better results compared to the protocol [41], which has the best performance among the [34], [35], [38], [39], [40], [41] group. This translates to a bandwidth saving of 1.2 megabytes, as shown in the Table 6.

Table 6. Communication cost in ECDSA*-based scheme

Protocol	Signing process (ms)	Verification process (ms)
[16]	191 bytes	191 n bytes
[32]	235 bytes	235 n bytes
[33]	235 bytes	235 n bytes
[38]	607 bytes	607 n bytes
[39]	607 bytes	607 n bytes
[40]	579 bytes	579 n bytes
[41]	255 bytes	255 n bytes
[34]	275 bytes	275 n bytes
[35]	275 bytes	275 n bytes
ECDSA*-based scheme	215 bytes	215 n bytes

Additionally, Figure 14 illustrates the communication cost related to the number of messages.

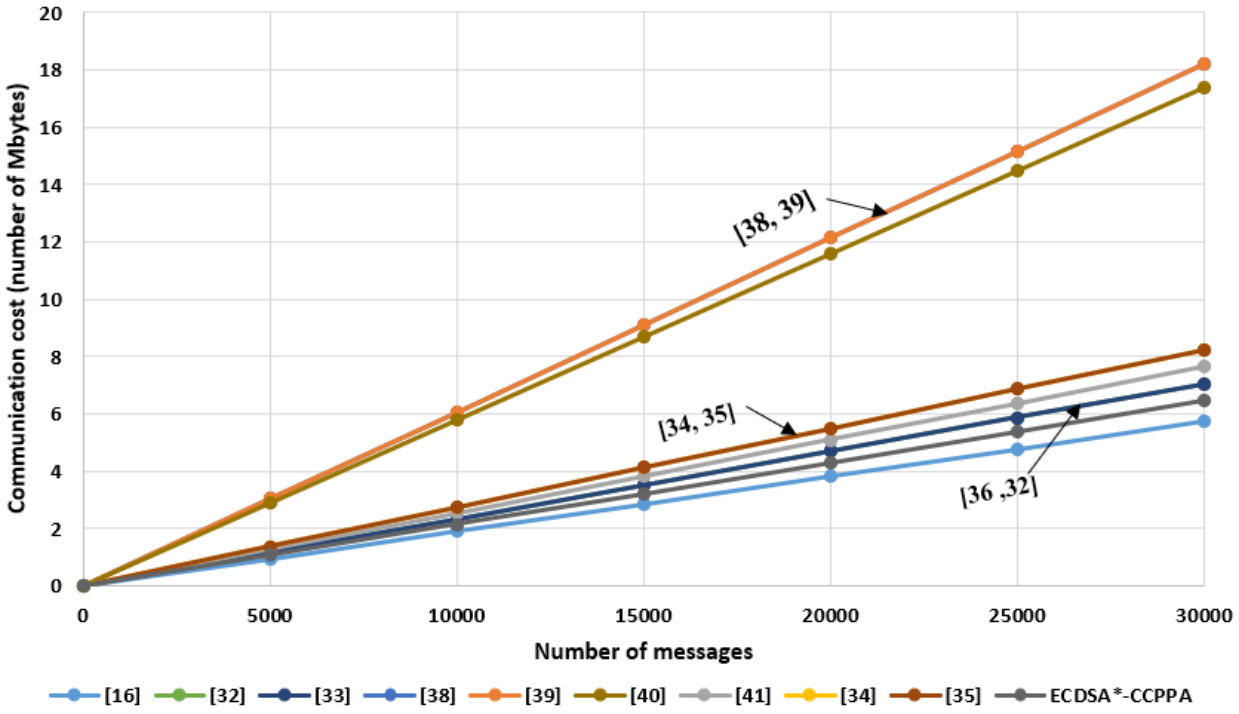


Fig 14. Cost communication in ECDSA*-based scheme vs number of messages

3.12 Conclusion regarding ECDSA*-based scheme

Our research work introduces two conditional anonymity-preserving authentication protocols, ECDSA-based and ECDSA*-based schemes, which do not need the insertion of certificates. Both protocols leverage ECC and avoid the need for MTH functions and pairings. These protocols achieve a TL 3 and satisfy all security requirements in VANET. Moreover, they enable

nodes to use ECDSA and ECDSA* algorithms to authenticate communications. Notably, ECDSA*-based scheme offers an advantage over ECDSA-based scheme by allowing RSUs to carry out a batch verification and assist VCs in verifying their safety-related messages by disclosing the y-coordinate of the point *R*. A security proof in the random oracle model demonstrates that both protocols are secure, considering the difficulty of solving ECC Diffie-Hellman ECCDH and ECDLP. Our results indicate that ECDSA*-based scheme outperforms several protocols, particularly in the signing and verification of single messages, and incurs lower communication cost compared to the studied CL-based protocols. The evaluation was conducted at a SL of 80 bits. In VANET, our ECDSA-based and ECDSA*-based schemes can be employed with a SL of 128 bits, as standardized in[11], [12]. Finally, these protocols can also find applications in authentication connected objects in other IoT.

CHAPTER 4: CL-BASED PROTOCOL WITH SCHNORR AUTHENTICATION

In this Chapter, we introduce our Schnorr-based scheme [83]. We begin by giving an introduction of the context of application of our scheme. In the section 2, we explain the challenge faced in VANET and how our scheme can counter the present vulnerabilities. In the section 3, we outline the security objectives of our Schnorr-based scheme. The section 4 explains how nodes can use those protocols in the network, while the section 5 describes the system model, and the section 6 details the algorithm used. We provide an overview and cryptographic operations details of our scheme in the sections 7 and 8, respectively. Additionally, we present a security proof and analysis in the sections 9. A simulation was performed, as described in the section 10, 11, 12 and 13. Then, we conclude the chapter in the section 12.

4.1 Introduction

Context Related to Schnorr-based scheme. Authentication is considered as an important component in safeguarding VANET against potential cyber threats. Our second research work introduces an efficient CPPA protocol that dispenses with the need for certificates and adeptly addresses privacy concerns throughout the authentication process. Our second new protocol is called Schnorr-based scheme and uses ECC, sidesteps the use of MTH, and obviates the necessity for pairings. It enables to implement Schnorr algorithm by optimizing the length of signatures into compact forms during VANET authentication. When multiple VCs dispatch identical safety-related messages to a RSU, this protocol empowers RSU to validate and aggregate individual signatures into a single aggregated signature via a MultiSig function. The aggregated signature can then be relayed to a TCC for network administration. Furthermore, the RSU can authenticate diverse safety-related messages from VCs through batch verification, aggregating the signatures into a singular form for transmission to a TCC. In both MultiSig verification and batch verification processes, Schnorr-based scheme exhibits robust resistance to rogue attacks, as the RSU necessitates validation of proof of possession (POP) as confirmation of each signer's possession of the *sk*. The protocol is fortified by a security proof, affirming its EUF-CMA. Our simulation results demonstrate that our Schnorr-based scheme's performance is better over numerous other research endeavors in terms of execution speed for both single and multiple safety-related messages.

4.2 Problem statement regarding Schnorr-based scheme

Regarding traditional digital signature algorithms, the authors [84] introduced a protocol featuring lightweight computations that utilize Schnorr signatures and tokens for authentication. In addition, their system can be classified as ID-PKC, as the TA provides a complete *sk* to VCs when vehicles are registered. Moreover, their protocol is restricted to V2I communications and lacks support for MultiSig or batch verification. Additionally, A. Imghoure et al.[9] presented a CL-based protocol based on the ECDSA algorithm, wherein ECDSA serves as the digital signature mechanism for network authentication. Nevertheless, Schnorr still outperforms ECDSA by enabling MultiSig strategy. In this regard, MultiSig outperforms a single signature because it involves multiple entities to validate a safety-related message, thereby enhancing security and trust. Additionally, our shnorr-based scheme is considered as a CL-based protocol

and addresses the limitations associated with certificate-based and escrow problem-based protocols. Furthermore, our protocol implements Schnorr algorithm [85] and turns out that it provides a robust scheme for ensuring authentication, non-repudiation, and data integrity.

4.3 Objectives of Schnorr-based scheme

Our contribution lies in combining the following two advantages:

1. Utilizing a CL-based protocol that eliminates the need for certificates and overcomes the escrow issue. Our contribution is to combine the following two advantages: the first one is to use a CL protocol that operates without inserting certificates and addresses the key escrow problem [86].
2. Integrating a classical algorithm, which is Schnorr algorithm [85]. It allows a MultiSig and aggregation processes when authenticating the same safety-related message from different senders. In addition, our protocol enables the use of the Schnorr algorithm in a way that batch verification and aggregation are possible when receiving different safety-related messages from signers.

4.4 Introduction of our Schnorr-based scheme

In our protocol, a RSU has the capability to verify identical safety-related messages originating from multiple senders and carry out also a batch verification when the safety-related messages are different and received from different signers. In both cases, a RSU possesses the ability to aggregate individual signatures into one signature, which can then be transmitted to a central TCC. Regarding MultiSig strategy, our protocol turns out resistant to a rogue attack. According to literature, this kind of attack turns out a concern when adversaries can arbitrarily select their *pk*s [86]. To mitigate this threat, we adopt the first method outlined by Boneh, where the user is required to provide a proof of knowledge or possession of a *sk* [87], [88], [89]. In this approach, users are required to furnish proof of knowledge or possession of a *sk* [90], [91]. The second method involves aggregation using different safety-related messages [89]–[91]. In our Schnorr-based scheme, we opt for the first option, which introduces a Proof of Possession (PoP). This choice aligns with our protocol's goal of addressing both scenarios: verifying the same safety-related message and verifying multiple safety-related messages through MultiSig and batch verification techniques. We have conducted simulations to compare the performance of our protocol with many other protocols. Our evaluation consists of analyzing execution time for signing and verifying both single and multiple safety-related messages, as well as the communication cost. As a result, our scheme turns out that it allows a driver to reduce the braking distance when verifying signatures, considering VC’s speed and density of road traffic.

4.5 VANET Model of our Schnorr -based scheme

Our Schnorr-based protocol comprises several key components to protect privacy and secure communications:

i. TRA: This component generates pseudonyms. When broadcasting safety-related messages, both VCs and RSUs employ pseudonyms instead of revealing their true identities. This measure is taken to safeguard against tracking attacks. Before entering to

the network, the TRA stores ture identities of VCs and RSUs in a database. In case when a VC or RSU requests a pseudonym, TRA initially verifies the authenticity of the node. If the true identity exists and is valid, the TRA then proceeds to create a pseudonym. Conversely, if the true identity is not found or the node is found illegitimate, the request is disregarded. Furthermore, our protocol allows the TRA to establish a link between a pseudonym and the actual identity of a VC or RSU during a dispute or in unauthorized activities. As a result, the TRA can identify and potentially revoke access to an unauthorized node when it makes subsequent requests for pseudonyms. This mechanism contributes to the overall network surveillance and security.

ii. *KGC:* This component is entrusted with the task of delivering a *psk* to a node. Here's how this the authority proceeds: When the TRA generates a pseudonym for a VC or an RSU, TRA forwards this pseudonym to the KGC. Subsequently, the KGC generates a *psk* and a corresponding *pk* based on the provided pseudonym. Plus, it sends both the pseudonym and the computed *pk* back to the TRA. With this information, the TRA stores the *pk* and actively monitor the communications via the pseudonym and *pk*. Importantly, the KGC delivers only a *psk* to VCs and RSUs and not the complete *sk*. This strategy aligns with the principles of a CL-based protocol. In our Schnorr-based scheme, the authority and node collaborate to build a valid *psk* and *pk*, but only the node has the capability to compute the corresponding full *sk* [53], [86].

iii. *RSU:* When RSU receives safety-related messages from VCs, it can employ a MultiSig to verify and aggregate signatures as indicated in Figure 15, or carry out a batch verification for different safety-related messages. In our protocol, MultiSig and batch verification are used to efficiently reduce computation time and create short signatures [38], [39], [40], [41], [92], [72], [93], [94].

iv. *VC:* This component informs and forwards safety-related messages to its nearby VCs and RSUs within its communication range using an OBU.

v. *TCC:* This component plays a central role in our protocol, as it is responsible for several functions within the network: collection of safety-related message, network management and traffic management. In Figure 15, Application Server represents TCC.

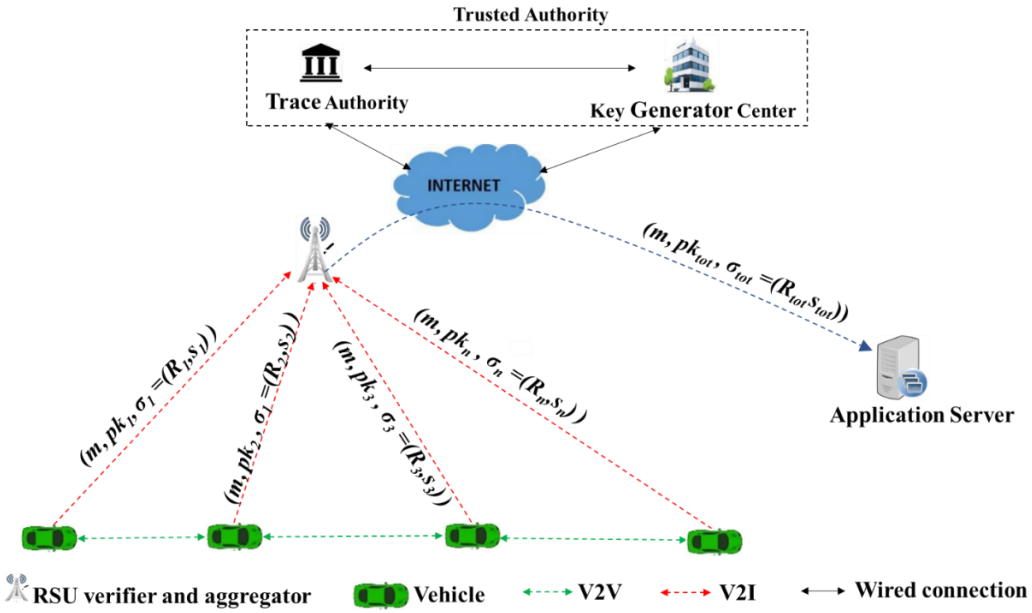


Fig 15. Architecture of Schnorr-based scheme during MultiSig process

4.6 Schnorr algorithm

Schnorr-based scheme uses ECC, and the complexity related to ECDLP and ECCDHP. Furthermore, our protocol allows the integration of Schnorr digital signature algorithm. Below, we provide the notations used in our scheme and definitions in ECC in table 7.

Table 7. Notations used in the Schnorr-based protocol.

Symbol	Description
$E(F_p)$	an elliptic curve defined within the finite field Fp.
n	an integer from Z_q^*
P	Base point
p, q	two primes
p	order of the field F_p
G	group of order q
V_i/ RSU_i	node
m_i	safety-related message
$\oplus$	exclusive *OR*
(β, T_{pub})	TRA's key pair
(α, P_{pub})	KGC's key pair
(s_k, pk)	Node's key pair used in Schnorr algorithm
psk_i	*psk* delivered to a node
sk_i	full *sk* of a node
pk_i	*pk* of a node
pk_{tot}	Aggregate pk generated by RSU_i
$\sigma_i = (R_i, s_i)$	Schnorr signature generated by a node
$\sigma_{tot} = (R_{tot}, s_{tot})$	Aggregate signature

X_i	Point in ECC
PoP_i	possession of proof
RID_i	True identity of a node
$PID_{i,1}$	pseudonym created by a node *i*
$PID_{i,2}$	pseudonym transmitted from TRA to a node *i*
ID_i	identity created by a node *i*
PID_i	pseudo identity of a node *i* generated by TRA
H	$\{0, 1\}^* \times G \longrightarrow Z_q^*$: hash function
h_1	$\{0, 1\}^* \longrightarrow Z_q^*$: hash function
h_2	$G \times G \times \{0, 1\}^* \longrightarrow Z_q^*$: hash function
T_i	Timespan of PID_i
t_i	timestamp

Schnorr signature scheme. is a method for generating digital signatures using the Schnorr signature algorithm. The scheme was initially developed by Schnorr [85] and its patent expired in 2008 [94]. The Schnorr signature scheme is planned to be incorporated into Bitcoin transactions [95], enabling entities to collaborate and create a *pk* aggregation using a MultiSig strategy. Consequently, this computational feature allows authentication, privacy, and the creation of a single *pk* that represents the sum of all the parties' *pk*s [96], [97]. In terms of security, Pointcheval and Stern demonstrated that the Schnorr signature scheme is CMA-EUF secure, assuming the hardness of the ECDLA [98], [99].

Schnorr signature. consists of the following algorithms:

Setup (1^λ)	*KeyGen (par)*
$(q, G, P) \leftarrow Gen\ (1^\lambda)$ Select $h: \{0,1\}^* \longrightarrow Z_q^*$ return $par: = (q, G, P, h)$	$(q, G, P, h) := par\ ;\ y \in Z_q^*;\ Y := y \cdot P$ $sk := (par, y)\ ;\ pk := (par, Y)$ return (sk, pk)
Sign (sk, m)	*Verify (pk, m, σ)*
$(n, G, P, h, y) := sk$ $r \leftarrow Z_q^*\ ;\ R := r \cdot P$ $c := h\ (R, m)\ ;\ s := r + c \cdot y \bmod q$ return $\sigma := (R, s)$	$(q, G, P, h, Y) := pk$ $(R, s) := \sigma$ $c := h\ (R, m)$ return $(s \cdot P = R + c \cdot Y)$

Description of Schnorr signature

4.7 Overview of Schnorr-based scheme

Our Schnorr-based scheme aligns with the concept of a CL-based scheme and involves five phases as explained in [79]. These phases and their respective algorithms are detailed in Figure 16, 17 and 18. In addition, a detailed description of each step is provided on how the *psk*, *pk*, and pseudo identity are created.

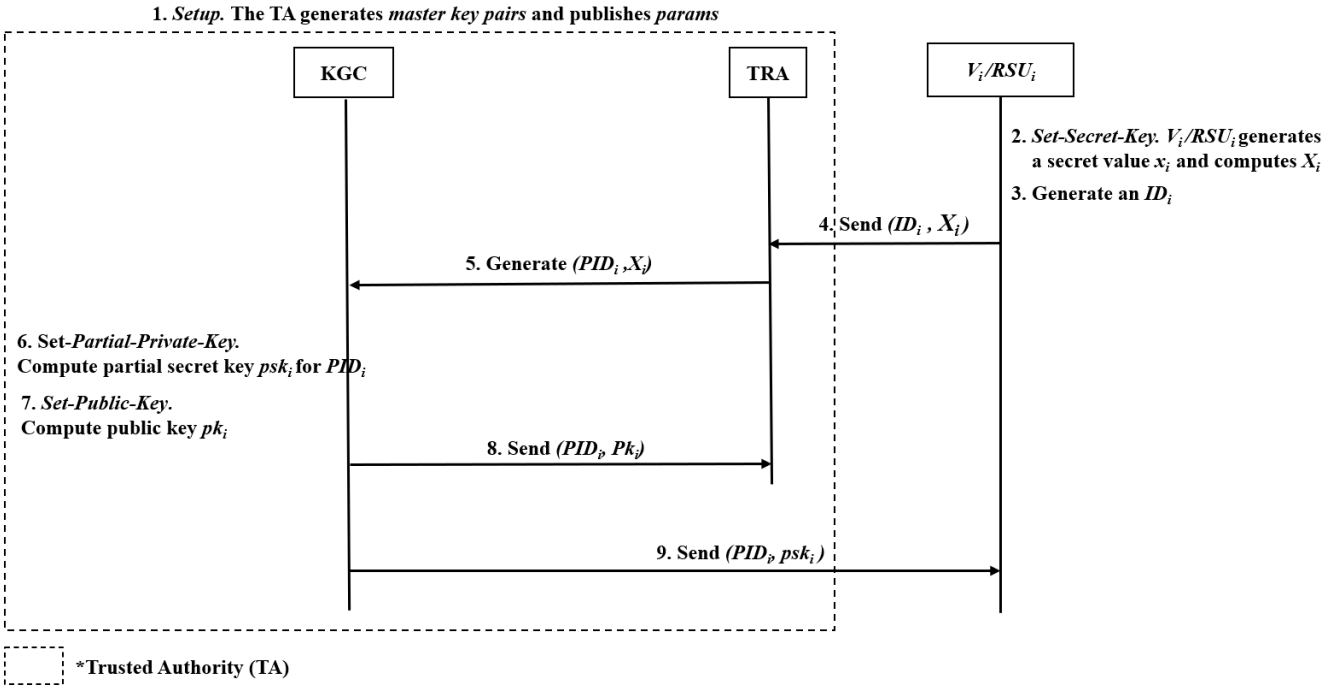


Fig 16. Overview of communication exchange between TA and a node

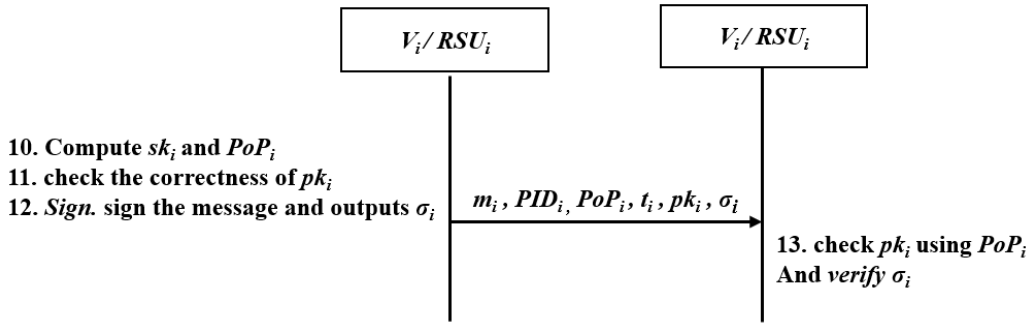


Fig 17. Overview of communication exchange between two nodes

4.8 Cryptographic Computations in Schnorr-based scheme

In this subsection, the cryptographic computations are provided for different parameters at each step in our Schnorr-based scheme. The Figures 18 and 19 show how the *psk* / *pk* / pseudonym are created.

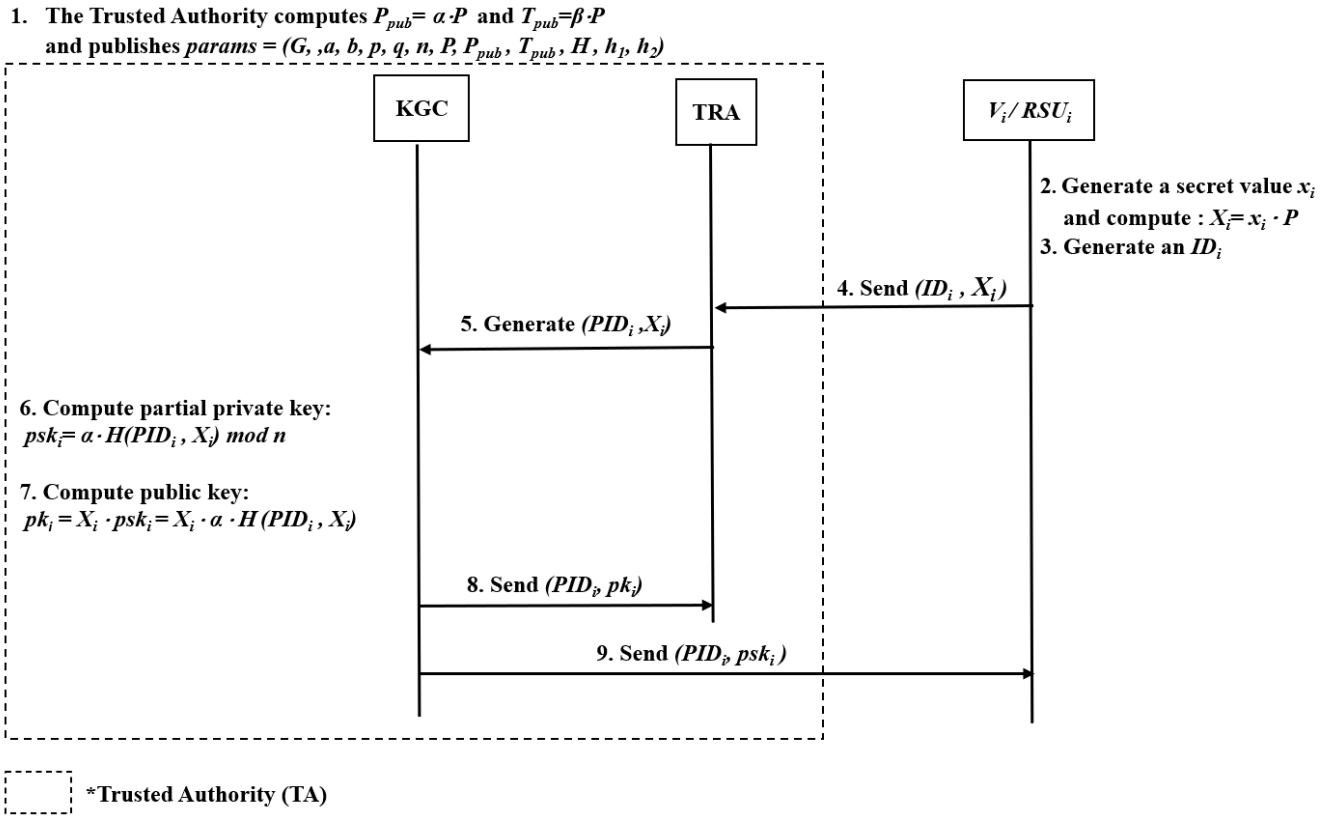


Fig 18. Cryptographic computations between TA and a node

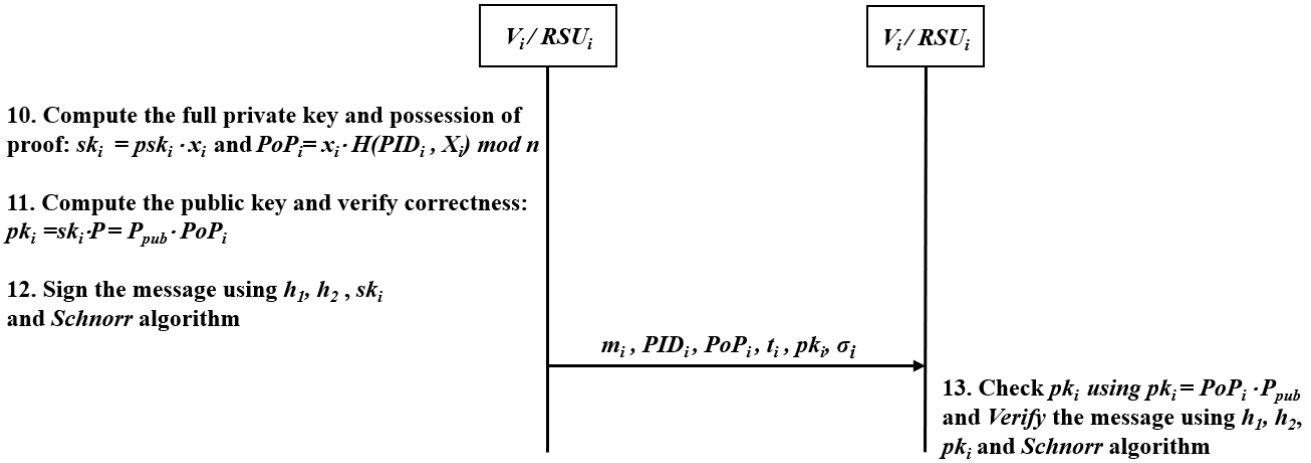
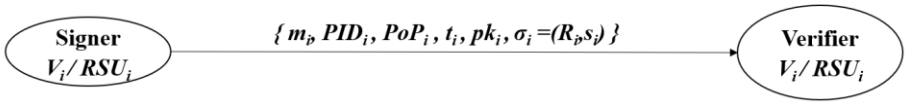
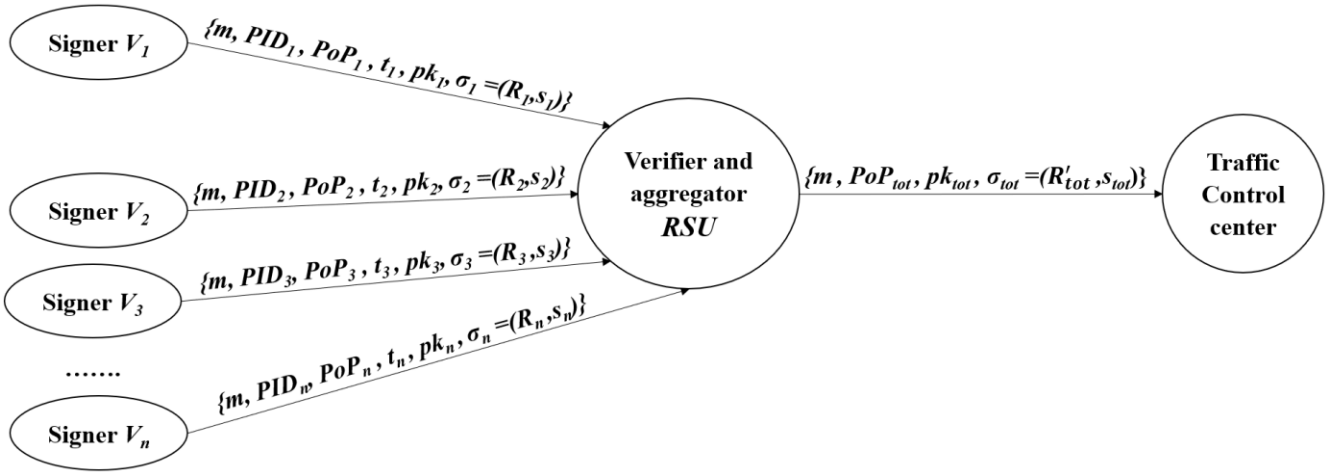


Fig 19. Cryptographic computations during communications between two nodes

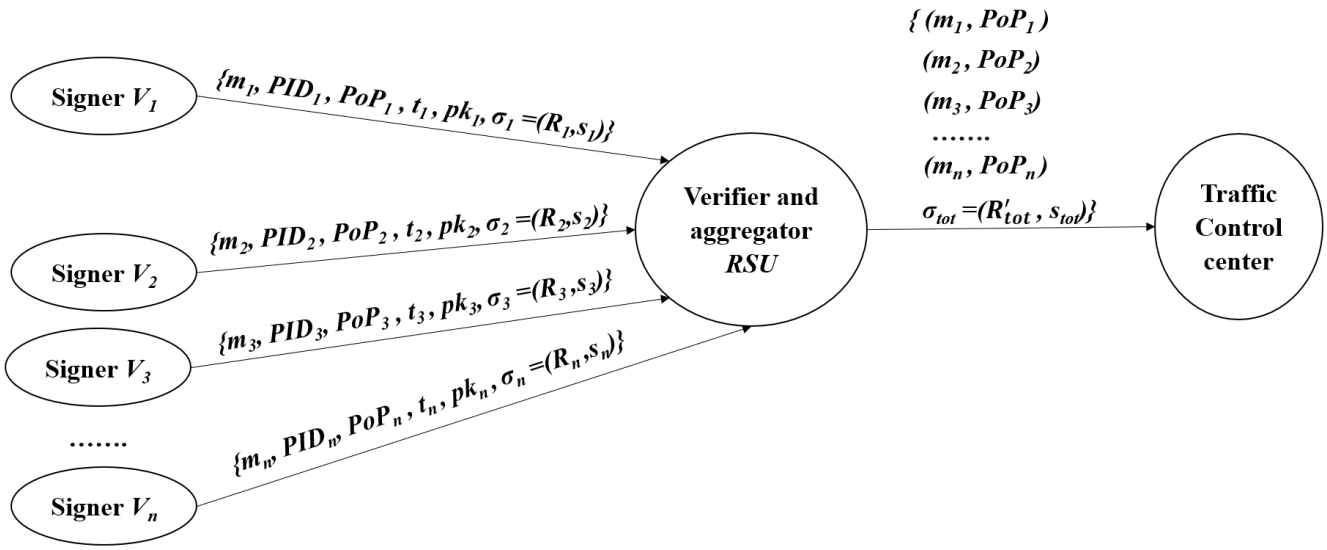
In our protocol, we distinguish between three scenarios to address different message signing situations. The first scenario deals with the format of a message originating from a single signer. The second scenario addresses the case where multiple signers transmit the same message. Finally, the third scenario covers the situation where multiple signers transmit distinct messages. This comprehensive approach allows our protocol to handle various signing configurations efficiently and securely.



Scenario 1: *Verification of a Single Message*



Scenario 2: *MultiSig and Signature Aggregation for the Same safety-related* m*essage*



Scenario 3: *Batch Verification and Aggregation of Signatures for different* safety-related m*essages*

Fig. 20 Authentication scenarios in Schnorr-based scheme

4.9 Security proof of Schnorr-based scheme

The security of the Schnorr-CPPA scheme is demonstrated through its resistance to existential forgery under chosen message attacks (EUF-CMA) [83]. This property is crucial for ensuring the integrity of digital signature schemes. The proof addresses two types of adversaries: Type I (TA_I) and Type II (TA_{II}).

Theorem 1 focuses on TA_I:

The scheme's security against TA_I relies on the computational difficulty of the Elliptic Curve Computational Diffie-Hellman Problem (ECCDHP). If an adversary of *type I* can successfully forge a signature after interacting with the provided oracles, then an algorithm exists that can solve the ECCDH problem in polynomial time. The success probability of this algorithm is expressed as a function of the adversary's success rate and the number of oracle queries.

Theorem 2 addresses TA_{II}:

For type II adversaries, the security is based on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP). Similar to the first theorem, if a *type II* adversary can produce a valid forgery, an algorithm can be constructed to solve the ECDL problem in polynomial time. The success probability of this algorithm is also related to the adversary's success rate and the number of oracle queries.

4.10 Simulation and performance evaluation of Schnorr-based scheme

During the simulation, we considered the cryptographic operations mentioned in the table 8.

Table 8. Execution time of cryptographic operations

Notation	Cryptographic operation
Δ_{bp}	$\overline{e}$ (A, B) in BP, where $\overline{A}$, $\overline{B} \in G_1$
Δ_{bp-m}	$x \cdot \overline{A}$ in BP where $\overline{A} \in G_1$, $x \in Z_q^*$
Δ_{bp-a}	$\overline{A} + \overline{B}$ in BP, where $\overline{P}$, $\overline{Q} \in G_1$
$\Delta_{m\text{-}ecc}$	$x \cdot A$ in ECC, where $A \in G$ and $x \in Z_q^*$
$\Delta_{a\text{-}ecc}$	$A + B$ in ECC, where $A, B \in G$
Δ_{MTP}	MTH function computation
Δ_h	hash function

The Table 9 and Figure 21 represent the execution time of signing and verifying for a single safety-related message for each protocol.

Table 9. Execution time during safety-related message authentication per protocol

Protocol	Signing of one safety-related message (ms)	Verification of one safety-related message (ms)	Total (ms)
[38]	$2\Delta_{bp-m}$	$3\Delta_{bp} + \Delta_{bp-m} + \Delta_{MTP}$	22.166
[39]	$2\Delta_{bp-m} + T_{MTP}$	$3\Delta_{bp} + \Delta_{bp-m} + 2\Delta_{MTP}$	30.978
[40]	$4\Delta_{bp-m}$	$3\Delta_{bp} + 3\Delta_{bp-m}$	24.596
[41]	$3\Delta_{bp-m}$	$3\Delta_{bp} + \Delta_{MTP} + 2\Delta_{bp-m}$	25.584
[92]	$4T_{bp\cdot m} + 2\ T_{MTP}$	$4T_{bp} + 2T_{bp\cdot m}$	35.910
[72]	$3T_{bp\cdot m}$	$3T_{bp} + 2T_{bp\cdot m}$	21.178
Schnorr-based scheme	$1\Delta_{m\text{-}ecc}$	$4\Delta_{m-ecc}$	2.210

Furthermore, the Table 10 and Figures 22 and 23 give the execution time of signing multiple safety-related messages, as well as a batch verification of multiple safety-related messages.

Table 10. Execution time of *n* safety-related messages per protocol

Protocol	Signing *n* saferty-related messages (ms)	Verification of *n* saferty-related messages (ms)
[38]	$2n\cdot\Delta_{bp-m}$	$3\cdot\Delta_{bp} + n\cdot\Delta_{bp-m} + n\cdot\Delta_{H}$
[39]	$n\cdot(2\Delta_{bp-m} + \Delta_{MTP})$	$3\cdot\Delta_{bp} + n\cdot\Delta_{bp-m} + (n+1)\cdot\Delta_{MTP}$
[40]	$4n\cdot\Delta_{bp-m}$	$3\cdot\Delta_{bp} + 3n\cdot\Delta_{bp-m}$
[41]	$3n\cdot\Delta_{bp-m}$	$3\cdot\Delta_{bp} + n\cdot(\Delta_{MTP} + 2\cdot\Delta_{bp-m})$
[92]	$4n\cdot\Delta_{bp\cdot m} + 2\ T_{MTP}$	$4\cdot\Delta_{bp} + 2n\cdot\Delta_{bp\cdot m}$
[72]	$3n\cdot\Delta_{bp\cdot m}$	$3\cdot\Delta_{bp} + 2n\cdot\Delta_{bp\cdot m}$
Schnorr-based scheme	$n\cdot\Delta_{m\text{-}ecc}$	$(2n+2)\cdot\Delta_{m\text{-}ecc}$

The figure 21, 22 and 23 depict the results from the Tables 9 and 10.

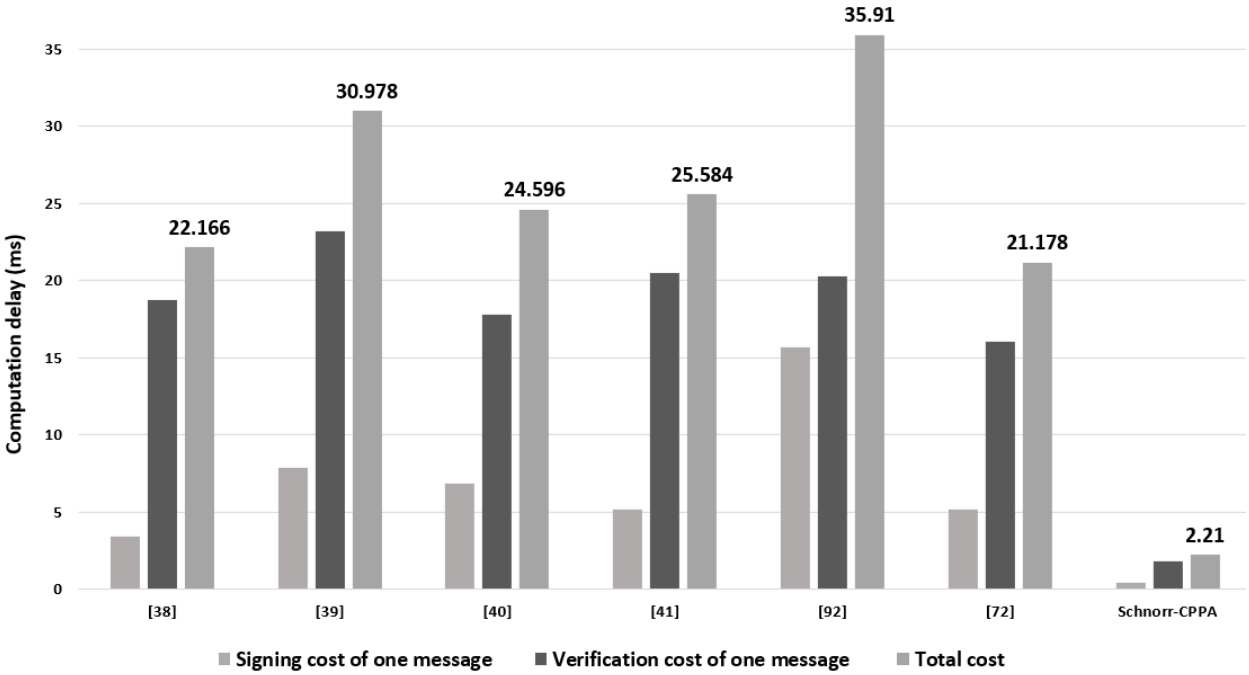


Fig 21. Execution time for a single safety-related message in Schnorr-based scheme

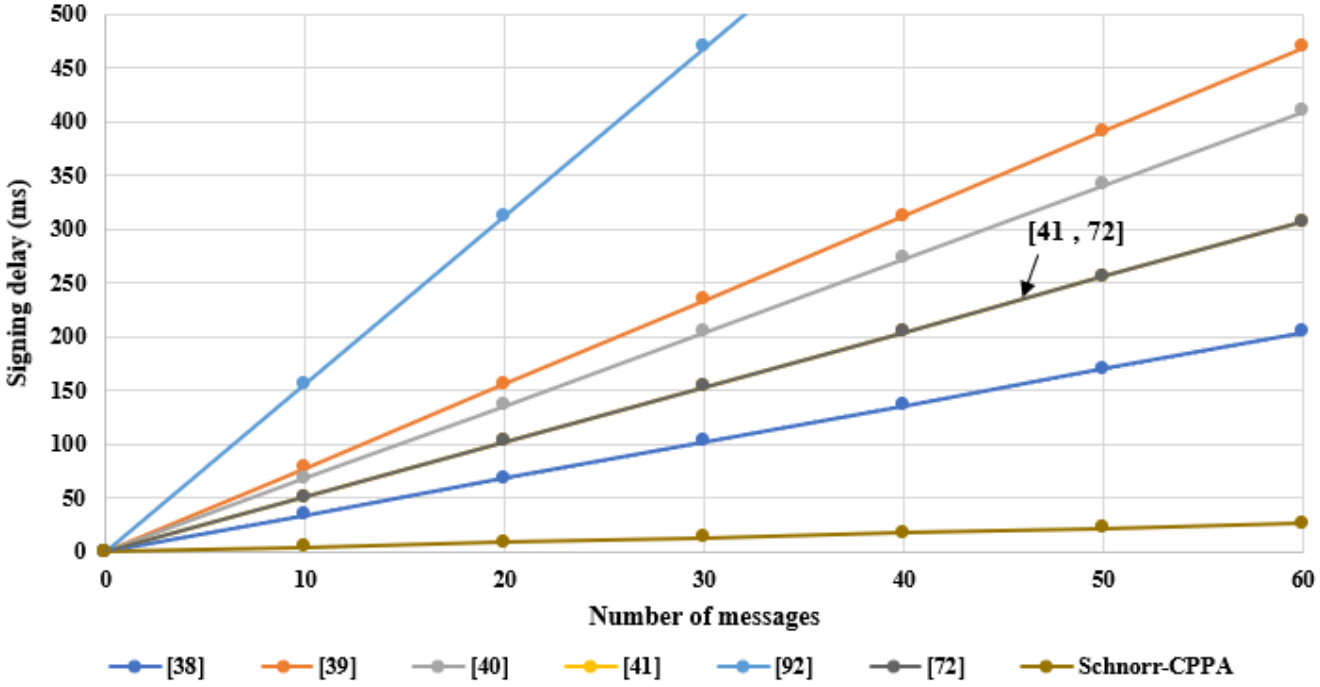


Fig 22. Execution time during signing for multiple safety-related messages in Schnorr-based scheme

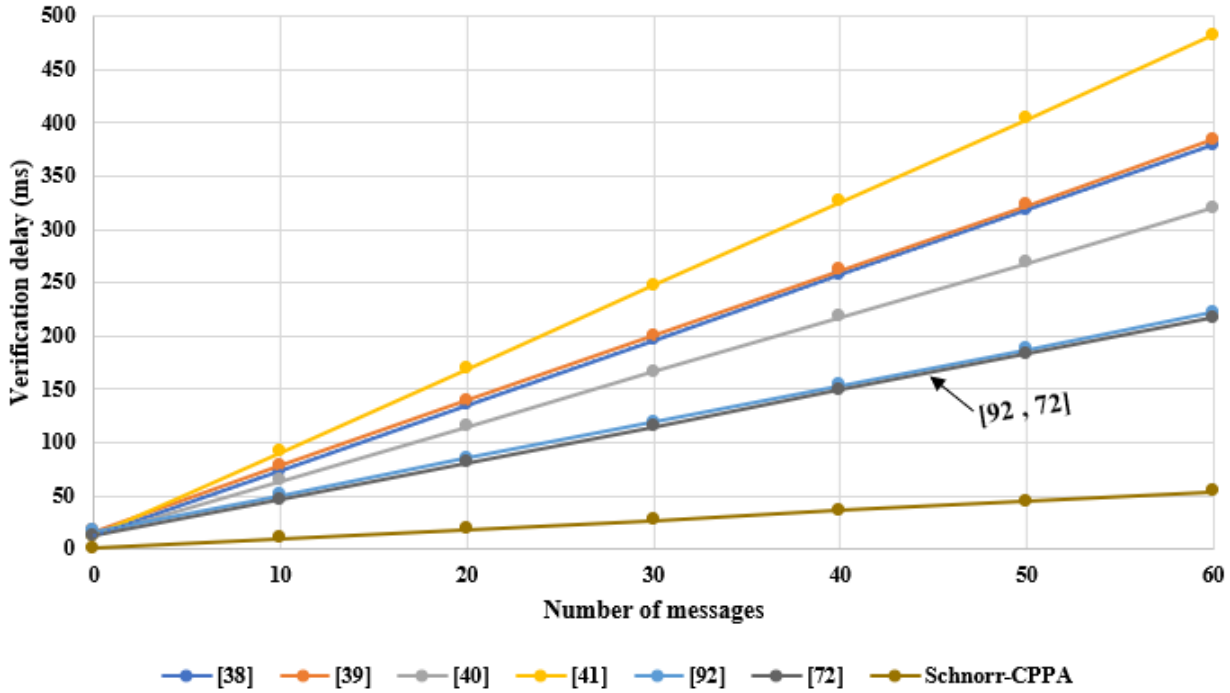


Fig 23. Execution time during verification for multiple safety-related messages in Schnorr-based scheme

4.11 Communication cost

In our setup, we have considered a SL of 80 bits, with $\bar{q}$ and q having lengths of 160 bits, and $|\bar{p}|$ = 64 bytes, $|p|$ = 20 bytes, $|G_1|$ = 128 bytes, $|G_1|$ = 40 bytes, as described in [16]. $|M|$ = 67 bytes, $|hash\ function|$ = 20 bytes, $|Timestamp|$ = 4 bytes. As shown in the Figure 24 and Table 11, both Schnorr-based scheme and [41] seem have lower communication cost than the schemes [38], [39], [40], [72], [92]. In addition, our Schnorr-based scheme and [41] are capable of saving 56% of communication overhead than [40].

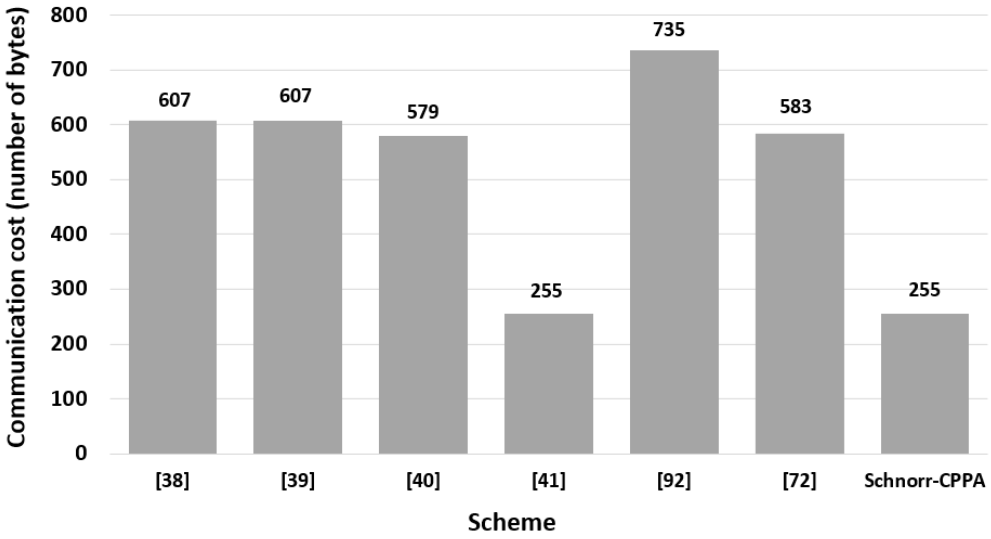


Fig 24. communication overhead in Schnorr-based scheme

Table 11. Communication Cost per Protocol

Scheme	One safety-related message	*n* safety-related messages
[38]	607 bytes	607 n bytes
[39]	607 bytes	607 n bytes
[40]	579 bytes	579 n bytes
[41]	255 bytes	255 n bytes
[92]	735 bytes	735 n bytes
[72]	583 bytes	583 n bytes
Schnorr-based scheme	255 bytes	255 n bytes

The Figure 25 gives the curves regarding the communication overhead vs number of broadcasted safety-related messages. When the number of safety-related messages go up to 25,000, both Schnorr-based and [41] schemes can save 8.1 megabytes of bandwidth in comparison with [40]. As a result, Schnorr-based and [41] schemes allow to reduce the communication overhead when the transmitted safety-related messages increase.

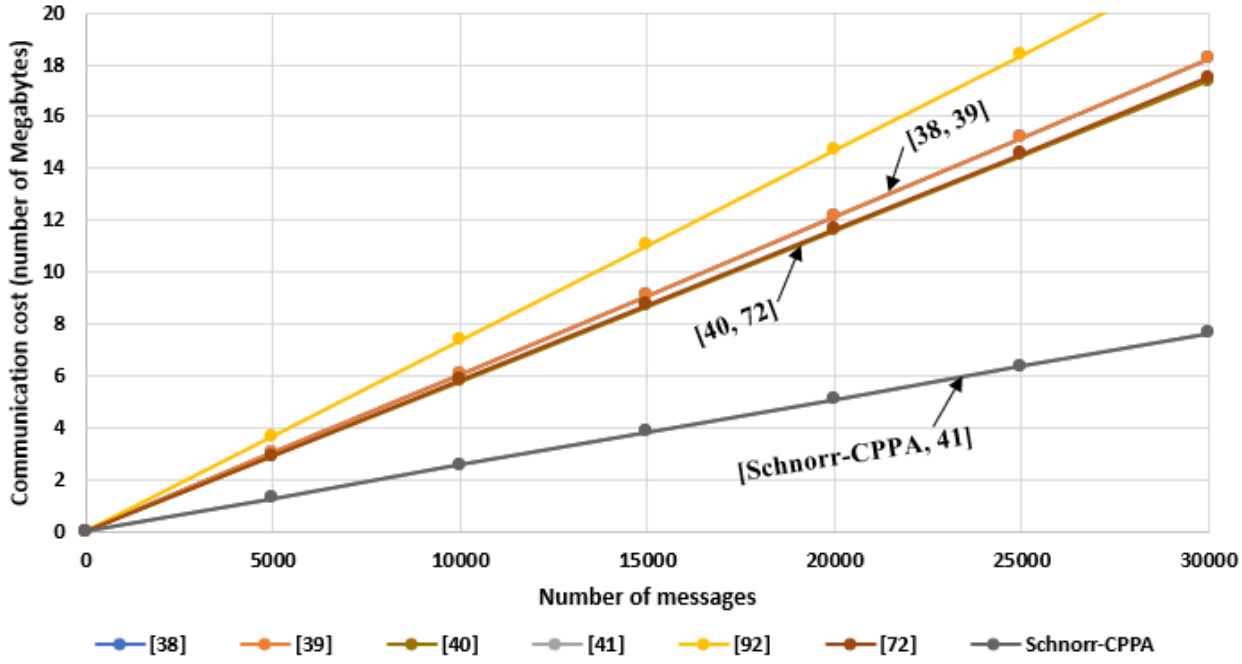


Fig 25. Communication cost in Schnorr-based scheme

4.12 Distance vs velocity

The distance to stop a VC plays a crucial role in ensuring safety of drivers. The faster speed is, the greater the distance will be required to bring the VC to a stop. In our analysis, we assume that VCs possess DSRC and have a broadcasting range of 300 meters. They have also the capability to carry out a batch verification. Assuming that VCs are uniformly distributed on a six-lane highway when traveling at a speed of 120 km/h. According to [100], a safe driving distance between VCs can be calculated as follows: $D_{inter\text{-}VCs} = 0.56 \times Speed$. Additionally, the approximate number of VCs within the transmission range can be determined as: $NTX = 2 \times NL \times \gamma \times \rho \times R$ which approximately equals 53 VCs. Here, $\gamma \times \rho$ represents the density of VCs (measured in VCs per kilometres per lane), and where γ is assumed to be 1, NL represents the number of lanes, and R is VC's coverage range. Using Schnorr-based protocol, if a VC receives 53 safety-related messages from VCs within its transmission range while traveling at a speed of 120 km/h, the distance D required to verify these 53 safety-related messages can be calculated as follows: $D = 1/3600 \times Speed \times T = 1.6$ m. Consequently, our Schnorr-based scheme allows to reduce the required distance by 71% when compared to [72], [92] at a speed of 120 km/h, as illustrated ed in Figure 26. This indicates that our Schnorr-based scheme is more efficient in term of distance compared to the other protocols under the same conditions.

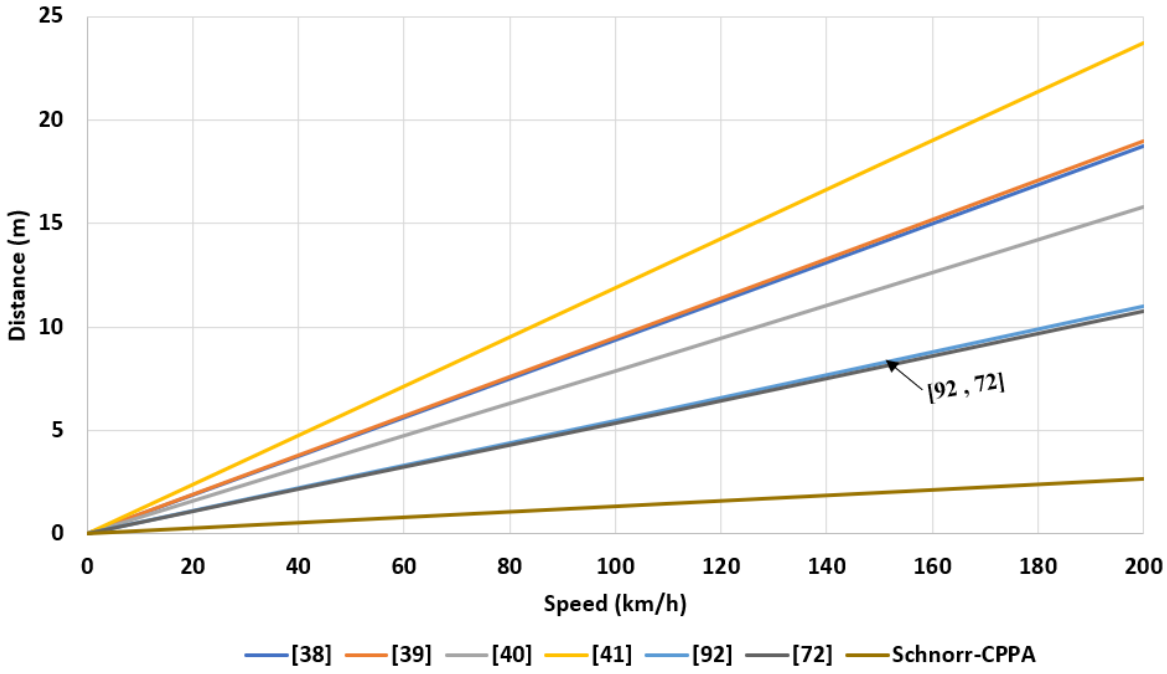


Fig 26. Distance traveled vs velocity

4.13 Distance vs traffic density analysis

The distance to stop a VC is a measure that encompasses several factors. It consists of the distance required to detect a safety-related event (D_v), the distance associated with the driver's reaction time when they initiate braking (D_r), and the distance needed for the VC to come to a complete stop after the brakes are applied (D_p). To calculate the total braking distance, you sum these three components: Braking distance = $D_v + D_r + D_p$. Additionally, Figure 27 illustrates these three components over time as they contribute to the overall braking distance calculation.

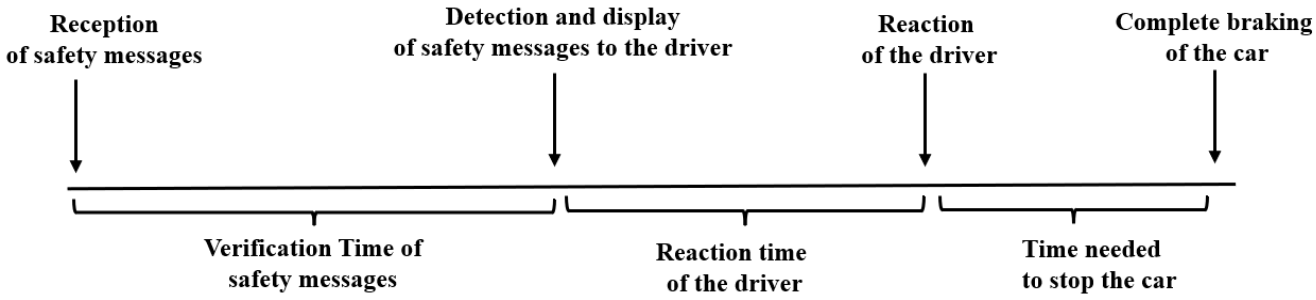


Fig 27. Timespan required for a VC to have a complete stop

Let's consider a situation where a car is traveling at a speed of 120 km/h and receives 53 safety-related messages. The calculation of the braking distance can be broken down into three components: *Dv* (the distance related to the verification time of safety-related messages), *Dr* (the distance associated with the driver's reaction time, which is 1.5 seconds), and *Dp* (the distance needed to stop the vehicle after applying the brakes). D_p is calculated

using the formula: $D_p = \frac{(\frac{5}{18} Speed)^2}{2a}$ = 82 m, where the car decelerates at a rate of 6.8m/s^2 after noticing the safety-related messages. Given these values, we calculate the total braking distance as follows: $D_v + D_r + D_p$ [100]. As a result, our Schnorr-based scheme allows to increase the safety distance by 4.5 m when compared to [72], [92], as illustrated in Figure 28.

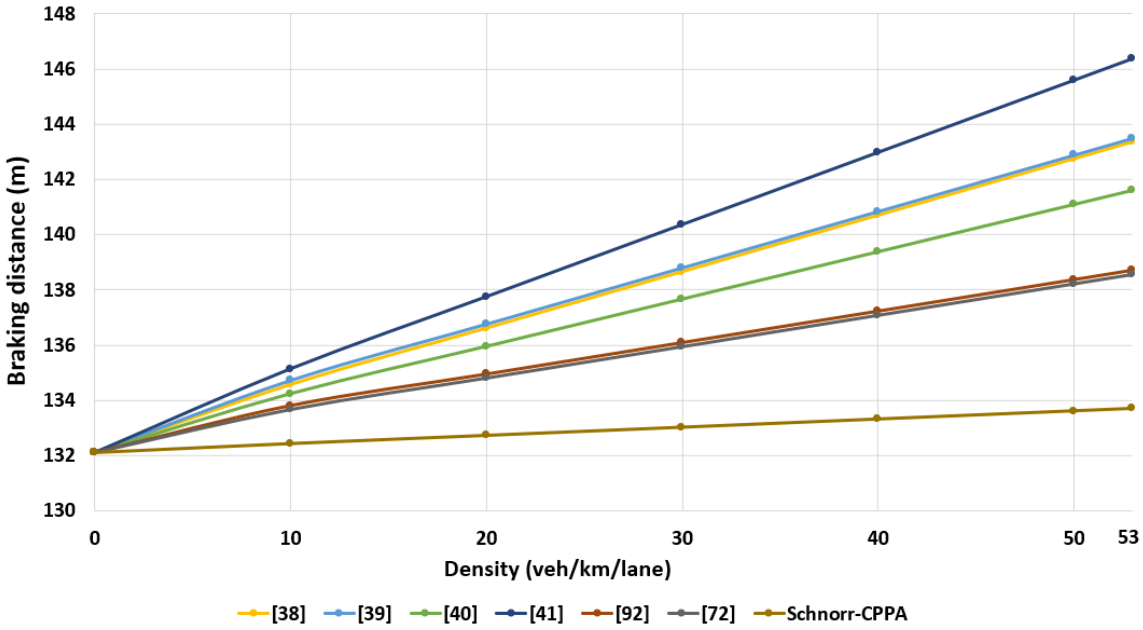


Fig 28 . Braking distance as a function of density at 120 km/h

4.14 Conclusion regarding Schnorr-based scheme

The Schnorr-based protocol has been developed as an efficient and lightweight authentication and implementation in VANET. It enables both VCs and RSUs to perform authentication using the Schnorr algorithm. Notably, our Schnorr-based scheme is a CL-based protocol, eliminating the need for certificates and addressing the key escrow issue. During the authentication process, RSU can engage a MultiSig computation when it receives identical safety-related messages from various signers. In addition, the received signatures are aggregated into a single signature, which is then sent to a TCC. Additionally, RSU has the capability to conduct a batch verification for various safety-related messages from multiple signers. It can merge these signatures into a unified aggregate signature before forwarding it to the TCC Importantly, Schnorr-based scheme satisfies all known requirements and effectively mitigates rogue attacks, which can be an issue when aggregating signatures. When broadcasting a message, our protocol shows resilience by requiring each signer to provide a proof of possession, demonstrating their ownership of the *sk*. Furthermore, our protocol is proved to be EUF-CMA. According to our simulation, the results show that the Schnorr-based scheme significantly reduce the time required for authentication and minimizes communication costs compared to many protocols. Regarding the analysis of distance, our system can provide drivers with an extra 4.5 m of safety distance compared to many protocols, when it comes to stopping the VC after detecting a dangerous event. Consequently, our scheme reduces the risk of accidents, and enhance overall road safety.

CHAPTER 5: CL-BASED PROTOCOL WITH MAPPING AUTHENTICATION

In this Chapter, we introduce our CLAS (Certificateless Aggregate Signature). We begin by giving an introduction of the context of application of our scheme. In the section 2, we explain the challenge faced in VANET and how our scheme can counter the present vulnerabilities. In the section 3, we outline the security objectives of our CLAS scheme. The section 4 explains how nodes can use those protocols in the network, while the section 5 describes the system model. We provide an overview and cryptographic operations details of our schemes in the sections 6 and 7, respectively. Additionally, we present a security proof and analysis in the sections 8. A simulation was performed, as described in the section 9 and 10. Then, we conclude the chapter in the section 11.

5.1 Introduction

Context related to our CLAS scheme. In VANET, a CLAS scheme satisfies both privacy and security requirements. Additionally, it allows a RSU to aggregate multiple signatures during V2I communication and send the aggregate signature to other RSUs and to a TCC. However, according to the classification of TLs defined by Girault [13], the existing CLAS schemes achieve only a TL 2, in which an unauthorized TA or VC can impersonate a legitimate VC by launching a PK-Replacement during the authentication process. According to the literature, the BP-based existing works provide a weak non-repudiation since it cannot be proven whether the authority or VC has replaced the original *pk*.

5.2 Problem Statement regarding our CLAS

Many protocols offer limited monitoring capabilities as the TA can only identify an unauthorized VC using the VC's identity and not its *pk*. In this work, a new CLAS protocol with a TL 3 is constructed using the hashing technique introduced by Al-Riyami and Paterson [51]. This technique binds the *pk* to its identity, ensuring that only *pk*s certified by the TA can pass the authentication process. Consequently, our CLAS protocol prevents the attack of *pk* replacement because even if an adversary (TA or VC) knows a *psk* of a VC, they cannot use a certified *pk*, because only a legitimate VC can know its x_i, and only a certified *pk* by the TA can be valid. On the one hand, the creation of a *pk* that has been tampered by an unauthorized TA can be demonstrated, as the presence of two certified *pk*s associated to the same identity will result in the presence of two keys that only the TA could have created. On the other hand, only a legitimate VC possessing the corresponding *sk* can use a certified *pk* and proceed to authentication. Therefore, the creation of a fake *pk* by an unauthorized TA or VC can be proven. Furthermore, our protocol provides stronger network monitoring capabilities as the TA can identify an unauthorized VC using both pieces of information: the VC's identity and its *pk*. Additionally, a proof shows that our CLAS is EUF-CMA given the difficulty of the Computational Diffie-Hellman problem. Regarding the evaluation of performance, our simulations demonstrate that our CLAS protocol provides the best execution time for both the signature and verification processes of a single safety-related message, as well as the verification of an aggregated signature compared to the studied CLAS protocols.

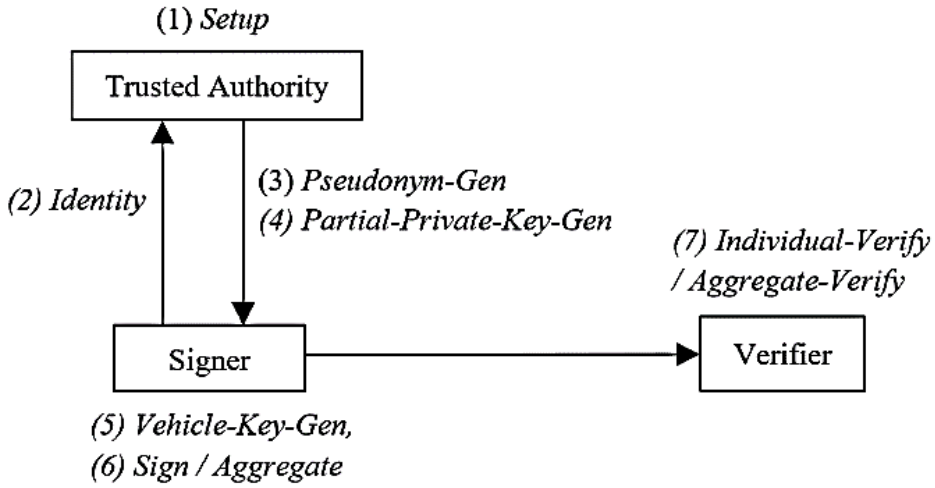
5.3 Objectives of Our CLAS

CL-based Aggregate Signature Protocol: The concept of an aggregate signature was introduced by Boneh et al. [103], in which many individual signatures can be compressed and aggregated into a single aggregate signature. To reduce the size of packets in high-traffic areas in VANET, several research works[38], [39], [40], [41], [64], [65], [66], [67], [68], [69], [70], [71], [72] suggest using a CL-based Aggregate Signature (CLAS) protocol, where a RSU can collect signatures from VCs within its range and then aggregate these signatures into a single aggregate signature, which is sent to other RSUs and a TCC. Generally, a CLAS protocol is based on eight algorithms, which can be defined as follows:

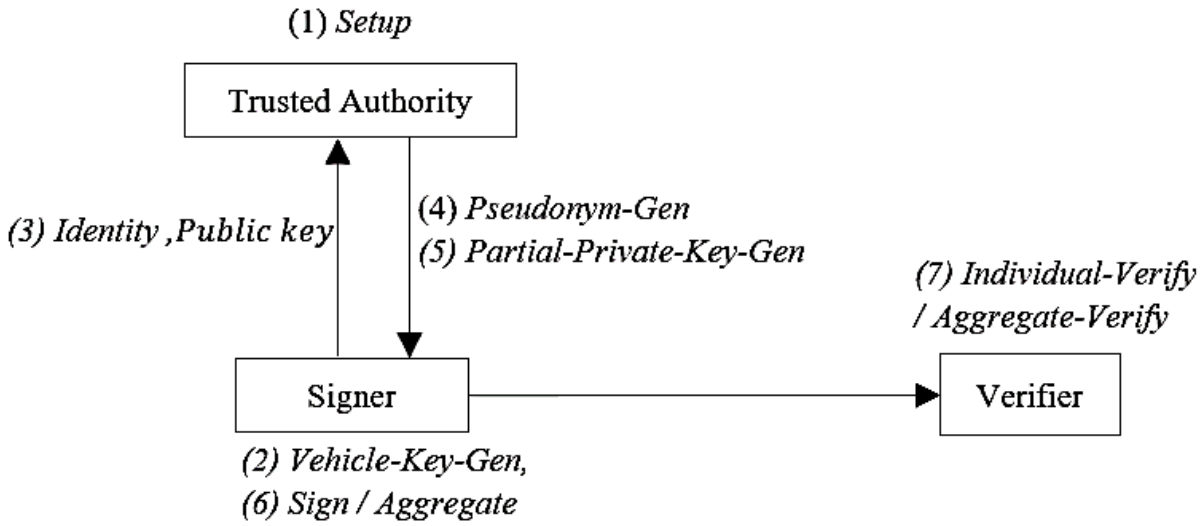
1. *Setup.* This algorithm is carried out by TA. First, it inputs a security parameter ℓ and outputs *params* and the master secret/ public key pairs (msk_{KGC} ,mpk_{KGC}) and (msk_{TRA} , mpk_{TRA}) of TRA and KGC.
2. *Vehicle-Key-Generation.* This algorithm is carried out by a VC that takes as input a VC's true identity ID_i and chooses $x_i \in Z_q^*$. Then, it outputs its *x/pk* pair (vsk_{PID_i}, vpk_{PID_i}), with $vsk_{PID_i} = x_i$. In addition, this algorithm outputs a pseudo identity $PID_{i,1}$. After that, the VC sends (ID_i, $PID_{i,1}$, vpk_{PID_i}) to TRA through a secure channel.
3. *Pseudo-Identity-Generation.* This algorithm is carried out by TRA. When the TRA receives (ID_i, $PID_{i,1}$, vpk_{PID_i}) from V_i, it checks VC's true identity ID_i. If ID_i is valid, the TRA stores its $vpk_{PID_i}$for monitoring purposes of the network and computes the VC 'pseudo identity as follows: $PID_i = (PID_{i,1}, PID_{i,2}, T_i)$. After that, the TRA sends PID_i to KGC.
4. *Partial-Private-Key-Extraction.* is run by KGC. When the KGC receives a pseudo identity PID_i from the TRA, it calculates the psk_{PID_i,vpk_i} using its master secret key msk_{KGC} and vpk_{ID_i}. Then KGC returns both: the pseudo identity PID_i and psk_{PID_i,vpk_i} to V_i.
5. *Sign.* This algorithm is carried out by a vehicle V_i that takes as input (PID_i, psk_{PID_i,vpk_i}) and a safety-related message $M_i \in \{0, 1\}^*$ and outputs a signature σ_i. After that, the VC sends (M_i, PID_i , vpk_{PID_i}, σ_i, t_i) to other VCs or a nearby RSU.
6. *Individual-Verify.* This algorithm is carried out by a VC during V2V communication or by a RSU during V2I communication. It takes mpk_{KGC}, M_i, PID_i, vpk_{PID_i}, σ_i and t_i. Then, it outputs *true* if σ_i is valid. Otherwise, *false* if σ_i is invalid.
7. *Aggregate.* This algorithm is carried out by an aggregator that collects multiple signatures, in which a VC's signature is defined as follows: $\sigma_i = (R_i, S_i)$. For instance, an aggregator RSU that receives multiple safety-related messages $\{M_1, \sigma_1 = (R_1, S_1)\}$, $\{M_2, \sigma_2 = (R_2, S_2)\} \ldots \{M_n, \sigma_n = (R_n, S_n)\}$ with their corresponding $\{PID_1 , PID_2 \ldots PID_n\}$ and VC's *pk*s $\{vpk_{ID_1}, vpk_{ID_2} \ldots vpk_{ID_n}\}$from *n* VCs $\{V_1, V_2, \ldots, V_n\}$ can aggregate the signatures of safety-related messages as follows: $S = \sum_{i=1}^{n} S_i$. Finally, the RSU aggregator sends $(M_1, M_2 \ldots, M_n, \sigma)$ to other RSUs and to a TCC.
8. *Aggregate-Verify.* This algorithm is carried out by a verifier RSU or a TCC. It takes $(M_1, M_2 \ldots, M_n, \sigma)$ and their corresponding (PID_i, vpk_{PID_i}) with $i \in [1,n]$ coming from *n* VCs $(V_1, V_2, \ldots, V_n)$ and outputs *true* if the σ is valid. Otherwise, *false* if invalid.

5.4 Introduction of our new CLAS Protocol

In this work, a new CL-based Aggregate Signature (CLAS) scheme is presented. Our scheme uses a CL-based cryptography that resolves challenges related to the insertion of a certificate and the escrow problem. To avoid a potential overhead in the network, a RSU can aggregate signatures coming from multiple VCs and send an aggregate signature to other RSUs and to a TCC. According to the above-mentioned classification given by Girault, our CLAS scheme allows a normal CLAS with a TL 2 as shown in Fig 29- (a) to be lift to a CLAS with a TL 3 as mentioned in Fig 29- (b). As a result, our scheme can prevent a PK-Replacement [75], [76], [77]. Even though an adversary can obtain a *psk*, it will not be able to use it to sign its safety-related message, because only the legitimate VC which possesses the corresponding secret is able to compute the private key *vsk*. In addition, an adversary cannot replace the *pk* which is *vpk* by a new *vpk** since it will need to request the new *psk** associated to *vpk**. Thus, the adversary cannot replace the original *pk* because only the TA can provide legitimate VCs with certified *pk*s. Regarding the architecture of the TA, we consider that the TA is composed of two entities. The first one is the TRA which delivers pseudo identities to VCs. The second one is the KGC which delivers *psk*s to VCs.



(a) Existing CLAS Protocols with TL 2



(b) Our CLAS Protocol with TL 3

Fig 29. (a) Description of Existing CL-based Protocols and (b) Our CLAS Protocol

Our work comes up with a new CLAS protocol that achieves a TL 3 in VANET, in which our contributions can be summarized as follows:

i. *TL 3.* Unlike a normal CLAS scheme with a TL 2, in which a *pk* is randomly generated by a signer, our CLAS scheme can be classified as a TL 3 since only certified *pk*s can pass the authentication process. In this regard, the TA binds each pseudo identity to a unique *pk* when a *psk* is created, as shown in the phase 4 - Fig 29 - (b).

ii. *PK-Replacement.* In a normal CLAS scheme, an adversary who can obtain a *psk*, he/she can create a *pk* by its choice, replace the original one, and then sign its safety-related message. But our scheme prevents the *pk* from being replaced, since even if an adversary who obtains a *psk*, he/she cannot calculate the corresponding *pk*. In this case, only a legitimate VC that knows the corresponding *x* can combine it with the *psk*.

iii. *Strong non-repudiation.* In our scheme, both the TA and VC are involved in generating a valid *pk*. The presence of two *pk*s associated with the same identity will cause to the presence of two valid *psk*s that are bound to the same identity as well. In this case, the TA will be accountable. Thus, it can be proven that the TA has harmfully created a falsified public *pk*. Furthermore, a legitimate signer can be the only one that possesses the corresponding *x* to the *pk* and sign a safety-related message. Thus, it cannot deny this action.

iv. *Tracing an unauthorized VC using VC's pk and pseudo identity credentials.* In our scheme, the TA can identify an unauthorized VC using both credentials: VC's *pk*s and pseudo identities, since only the TA can provide valid pseudonyms and certified *pk*s.

v. *Security analysis.* We prove that our scheme is EUF-CMA using Computational Diffie–Hellman (CDH) problem.

In our CLAS, an asymmetric pairing is considered instead of a symmetric pairing. Plus, our CLAS does not use supersingular and hypersingular curves due to their limitation of

performance. Several research works recommend using *Type-2* or *Type-3* setting when developing a new protocol or modifying a protocol from a symmetric to an asymmetric setting. In this context, Boneh and al [101] mentioned that an efficiently-computable isomorphism ψ: $G_2 \rightarrow G_1$ is important to secure a protocol and can be skipped if only stronger complexity assumption is considered. In our protocol, MNT curves are considered as an alternative to supersingular curves since MNT curves have the advantage of using fields of high characteristic. Plus, they are considered safe against the Coppersmith attack and use the Complex Multiplication (CM) method to generate elliptic curves [102]. In addition, *Type-1* pairings provide a limited choice of curves and are significantly slower than asymmetric pairings at higher SL [103]. In this regard, our CLAS scheme is based on asymmetric pairing with a *type-2* sitting, and its intractability is based on CDH problem in the group in which the signature is created.

Description of the steps in our scheme. The steps used in our scheme are described as follows:

Step1: The TA is composed of (TRA and KGC) and provides the signer and verifier with public parameters.

Step2: The signer uses vsk_{PID_i} and calculates vpk_{PID_i}. The notation vpk_{PID_i}means that the *pk* and pseudo identity are bound and can be only used by the legitimate signer.

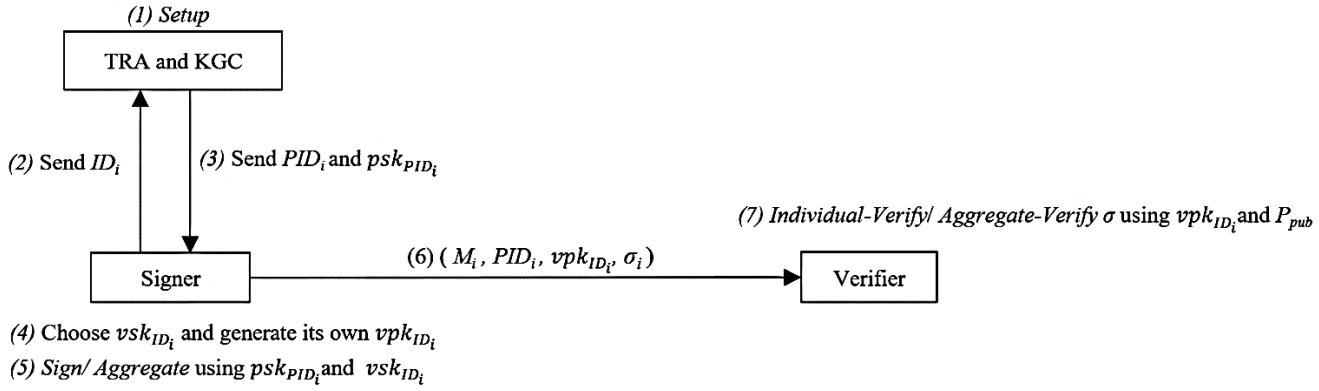
Step3: The signer transmits its ID_i and vpk_{PID_i} to the TRA.

Step4: TRA creates a pseudo identity PID_i and sends it to KGC. This latter generates a psk_{PID_i,vpk_i} and sends (PID_i, psk_i) to the signer. Unlike the existing CL-based schemes, our scheme ensures that psk_{PID_i,vpk_i} binds a pseudo identity PID_i to the *pk* of the signer vpk_{PID_i}, which will ensure the uniqueness of vpk_{PID_i}.

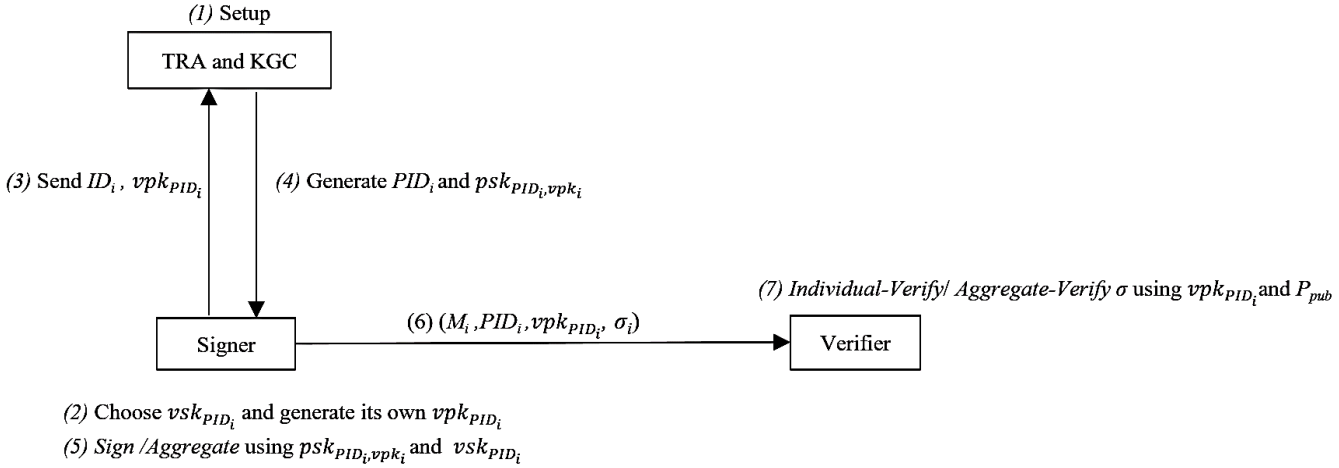
Step5: The signer computes a signature σ_i over its safety-related message using psk_{PID_i,vpk_i} and vsk_{PID_i}. In addition, the signer can also sign multiple safety-related messages by generating an aggregate signature, in which the algorithm *Aggregate* is used to output a single signature for multiple safety-related messages.

Step6: The signer sends its safety-related message(s) (M_i, PID_i, vpk_{PID_i}, σ_i) to the verifier.

Step7: The verifier uses *Individual-Verify* to verify σ in case it received one safety-related message. Also, the verifier can use *Aggregate-Verify* in case it received multiple safety-related messages that were sent by an aggregator signer. In our CLAS scheme, only certified $vpk_{ID_i}$by the TA can pass the authentication process.



(a) Existing CLAS schemes with a TL 2



(b) Our CLAS scheme with a TL 3

ID_i: Identity of a VC
PID_i: Pseudo identity of the signer
P_{pub}, T_{pub}: respectively the pks of KGC and TRA
M_i: Safety-related message
psk: Partial private key of the signer
vsk: Secret key generated by the signer
vpk: pk generated by the signer
σ_i: Signature

Fig. 30 Architecture of the existing CLAS schemes (a) and our CLAS scheme (b)

5.5 VANET model of our CLAS scheme

The architecture of our CLAS consists of two layers:

1) The upper layer, which is composed of KGC, TRA as well as an AS. Those entities exchange information through a wired connection.
2) The lower layer is composed of RSUs and VCs that exchange messages using DSRC protocol as defined in IEEE 802.11p.

The components of our model are described in Figure 31, and are defined as follows:

TA. is composed of two entities: TRA and KGC. During the registration phase, the TA sets system parameters, and preloads them in a TPD of VCs and RSUs before they have access to the network.

1) *TRA*. It stores identities of VCs and RSUs in a database before a node has access to the network. In our protocol, TRA provides VCs with pseudo identities to prevent a location-based attack. In this case, TRA can monitor participants to the network using: an identity and/or *pk*. Moreover, the true identity of a node is considered as sensitive data, a database encryption is assumed to be deployed, as explained in [9] to ensure security and privacy.
2) *KGC*. It provides RSUs and VCs with public parameters before they join the network. Additionally, the KGC regularly provides VCs with *psk*s after the TRA checks their true identities.

AS. The application server is responsible for collecting data from RSUs and analysing them to make decisions regarding the management of the network such as adjusting traffic lights.

RSU. It ensures V2I communications. When a RSU receives multiple signatures, it proceeds to a batch verification of signatures to check if all the signatures are valid. After that, the signatures are aggregated into one signature which is sent to other RSUs in other areas and to the AS for further analysis of the events occurring in the network. When a receiver RSU or AS receives the aggregate signature, it will only verify the aggregate signature which was carried out on behalf of multiple VCs.

VC. Each VC broadcasts its safety-related messages using an OBU and DSCR protocol. Before having access to the network, TRA preloads the identity of a VC in a TPD which is embedded in the VC. Hence, a VC can use its true identity ID_i to obtain pseudo identities and *psk*s from the authority.

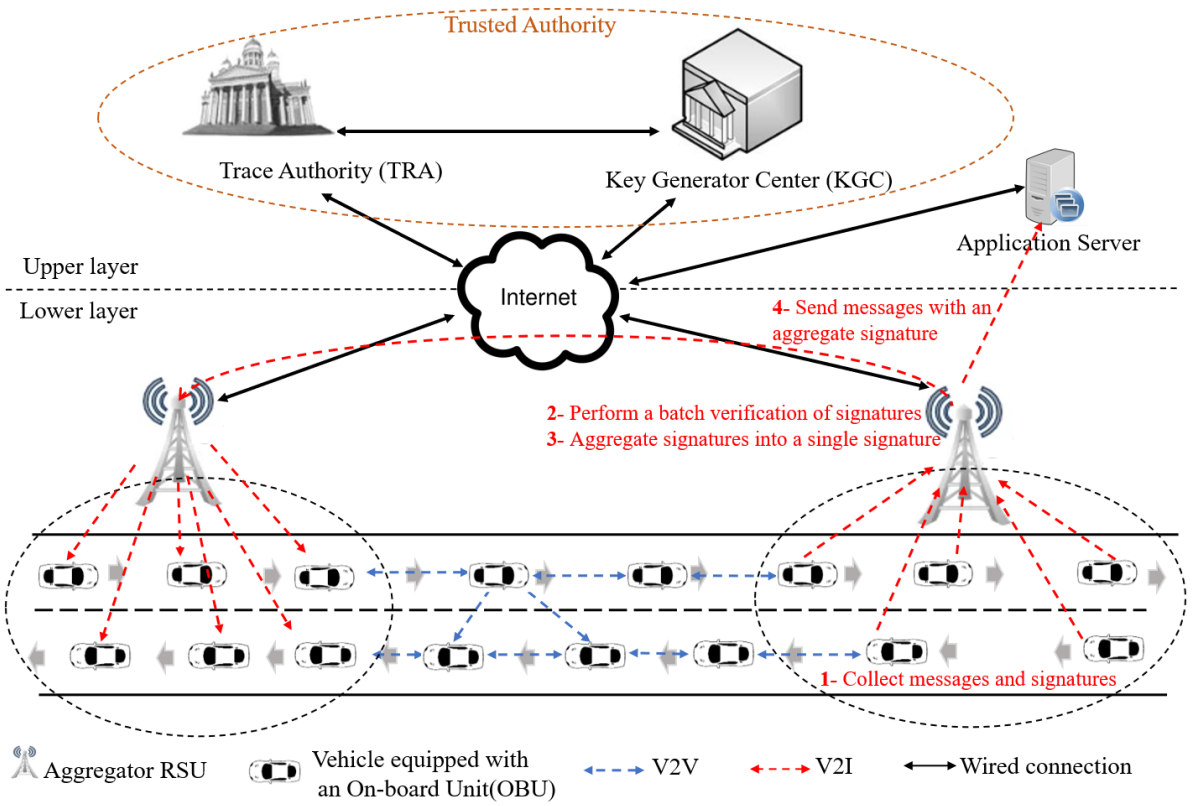


Fig 31. Our CLAS system model

5.6 Overview of Our CLAS Protocol

In this subsection, an overview of our CLAS and the symbols used in the scheme are given as described in Table 12.

Table 12. Notations used in our CLAS

Symbol	Description
$GF(p)$	a finite field with characteristic p.
$GF(p^k)$	an extension field of degree k.
$GF(p^k)^*$	A multiplicative group of $GF(p)$
G_1,G_2	Two cyclic additive groups of order q
G_T	A cyclic multiplicative group of order q
P, Q	Two different generators respectively in G_1 and G_2
e	A bilinear map e: $G_1 \times G_2 \longrightarrow G_T$.
p	Characteristic of the field
q	Order of the groups G_1, G_2 and G_T.
(α, P_{Pub})	Master key pair of the KGC
(β, T_{Pub})	Master key pair of the TRA
$(vsk_{PID_i}, vpk_{PID_i})$	x and pk of a vehicle V_i
psk_{PID_i,vpk_i}	A psk that binds PID_i to the VC's public key vpk_{PID_i}
$H_1(.)$	A MapToPoint Hash function [24], [104] such that: H_1: $\{0, 1\}^* \longrightarrow G_2$

$H_2(.)$	One way Hash [105] function such that: $H_2: \{0, 1\}^* \longrightarrow Z_q^*$
ID_i	True identity of a vehicle V_i
$PID_{i,1}$	Pseudonym computed by a vehicle V_i
$PID_{i,2}$	Pseudonym computed by TRA for V_i
PID_i	Pseudonym computed by TRA for V_i with $PID_i = (PID_{i,1}, PID_{i,2}, T_i)$
T_i	Valid period of a pseudo identity PID_i
t_i	timestamp
M_i	a safety-related message

Our CLAS scheme consists of eight algorithms: 1-*Setup*, 2-*Vehicle-Key-Generation*, 3-*Pseudo-Identity-Generation*, 4- *Partial-Private-Key-Extraction*, 5-*Sign*, 6-*Individual-Verify*, 7-*Aggregate* 8- *Aggregate-Verify*. Those algorithms can be described as follows:

- *Setup.* This algorithm is carried out by the TA. First, it inputs a security parameter ℓ and outputs parameters *params* and respectively the key pairs of TRA and KGC (msk_{KGC} ,mpk_{KGC}) and (msk_{TRA} , mpk_{TRA}).
- *Vehicle-Key-Generation.* This algorithm is carried out by a VC that takes as input a VC's real identity ID_i and chooses $x_i \in Z_q^*$. Then, it outputs its *x*/*pk* pair (vsk_{PID_i}, vpk_{PID_i}), with $vsk_{PID_i} = x_i$. In addition, this algorithm outputs a pseudo identity $PID_{i,1}$. After that, the VC sends (ID_i, $PID_{i,1}$, vpk_{PID_i}) to TRA through a secure channel.
- *Pseudo-Identity-Generation.* This algorithm is carried out by TRA. When the TRA receives (ID_i, $PID_{i,1}$, vpk_{PID_i}) from V_i, it checks VC's real identity ID_i. If ID_i is valid, the TRA stores its vpk_{PID_i} for monitoring purposes of the network and computes PID_i. After that, the TRA sends PID_i to KGC.
- *Partial-Private-Key-Extraction.* is run by KGC. When the KGC receives a pseudo identity PID_i from the TRA, it calculates psk_{PID_i,vpk_i} using its master secret key msk_{KGC} and vpk_{ID_i}. Then KGC returns both: the pseudo identity PID_i and psk_{PID_i,vpk_i} to V_i.
- *Sign.* This algorithm is carried out by a vehicle V_i that takes as input (PID_i, psk_{PID_i,vpk_i}) and a message $M_i \in \{0, 1\}^*$ and outputs a signature σ_i. After that, the VC sends (M_i, PID_i , vpk_{PID_i}, σ_i, t_i) to other VCs or a nearby RSU.
- *Individual-Verify.* This algorithm is carried out by a VC during V2V communication or by a RSU during V2I communication. It takes mpk_{KGC}, M_i, PID_i, vpk_{PID_i}, σ_i and t_i. Then, it outputs *true* if σ_i is valid. Otherwise, *false* if σ_i is invalid.
- *Aggregate.* This algorithm is carried out by an aggregator that collects multiple signatures, in which a VC's signature is defined as follows: $\sigma_i = (R_i, S_i)$. For instance, an aggregator RSU that receives multiple messages $\{M_1, \sigma_1 = (R_1, S_1)\}$, $\{M_2, \sigma_2 = (R_2, S_2)\} \ldots \{M_n, \sigma_n = (R_n, S_n)\}$ with their corresponding $\{PID_1 , PID_2 \ldots PID_n\}$ and VC's *pk*s $\{vpk_{ID_1}, vpk_{ID_2} \ldots vpk_{ID_n}\}$from *n* VCs $\{V_1, V_2, \ldots , V_n\}$ can aggregate the signatures of messages as follows: $S = \sum_{i=1}^{n} S_i$. Finally, the RSU aggregator sends $(M_1, M_2 \ldots, M_n, \sigma)$ to other RSUs and to a TCC.

- *Aggregate-Verify.* This algorithm is carried out by a verifier RSU or a TCC. It takes *(M_1, M_2 …, M_n, σ)* and their corresponding (PID_i, vpk_{PID_i}) with $i \in [1,n]$ coming from *n* VCs *(V_1, V_2,…, V_n)* and outputs *true* if the σ is valid. Otherwise, *false* if invalid.

5.7 Cryptographic computations of our CLAS scheme

In this subsection, we give details of the cryptographic computations used in our CLAS as follows:

1. *Setup.* The TA sets system parameters *params*. First, the TA takes a security parameter ℓ and generates a prime *q*, three groups G_1, G_2 and G_T of order *q*, two generators *P* and *Q* respectively in G_1 and G_2 and a bilinear map e: $G_1 \times G_2 \longrightarrow G_T$, in which G_1 and G_2 are respectively a cyclic additive group and G_T a cyclic multiplicative group. First, the KGC chooses its master secret key $\alpha \in Z_q^*$ and then calculates its master public key $P_{Pub} = \alpha \cdot P$. Besides, the TRA also chooses master secret key $\beta \in Z_q^*$, and then computes its master public key $T_{Pub} = \beta \cdot P$. In addition, the TA sets the following hash functions H_1: *{0, 1}*$\longrightarrow G_2$ and H_2: *{0, 1}*$\longrightarrow Z_q^*$. The computation of *Q* is performed according to [39], in which an unauthorized-but-passive KGC is allowed to compute a key pair in any way. Then, the TA preloads *Params* into TPDs of RSUs and OBUs, while the master secrets α and β are respectively kept secret by KGC and TRA.
2. *Vehicle-Key-Generation:* First, a vehicle V_i chooses a secret key $x_{PID_i} \in Z_q^*$ and sets $x_{PID_i} = vsk_{PID_i}$. Then V_i calculates its *pk* as follows: $vpk_{PID_i} = x_{PID_i} \cdot P$. After that, V_i chooses $c_i \in Z_q^*$ and computes $PID_{i,1}$ as follows: $PID_{i,1} = c_i \cdot P$. Next, V_i transmits (ID_i, $PID_{i,1}$,vpk_{PID_i}) to the TRA.
3. *Pseudo-Identity-Generation.* When the TRA receives (ID_i, $PID_{i,1}$,vpk_{PID_i}) from V_i, the TRA checks first if the ID_i is valid. If it is, the TRA stores vpk_{PID_i} to trace the VC in case of an unauthorized event. In addition, it computes $PID_{i,2}$,where T_i is the valid period of PID_i. After that, the TRA transmits (PID_i , vpk_{PID_i}) to KGC.
4. *Partial-Private-Key-Extraction.* When KGC receives (PID_i,vpk_{ID_i}), this authority computes Q_{PID_i,vpk_i} and outputs the *psk*. After that, the KGC sends (PID_i , psk_{PID_i,vpk_i}) to V_i.
5. *Sign:* To broadcast a safety-related message M_i, the vehicle V_i uses (PID_i, psk_{PID_i,vpk_i}) to sign its safety-related message as follows:
   *5.1.*It selects $r_i \in Z_q^*$ and then computes R_i.
   *5.2.*It computes S_i in which h_i and t_i is a timestamp in order to prevent a replay attack.
   5.3. Finally, V_i transmits (M_i, PID_i , vpk_{PID_i}, σ_i, t_i) where $\sigma_i = (R_i, S_i)$ to other VCs and nearby RSU.
6. *Individual-Verify:* During V2V or V2I communications, a VC/ RSU which receives one signature under the format (M_i, PID_i , vpk_{PID_i}, σ_i, t_i), it proceeds as follows:
   *6.1.*It computes $Q_{PID_i,vpk_i} = H_1(PID_i,vpk_{PID_i})$ and $h_i = H_2(M_i, PID_i,vpk_{PID_i}, R_i, t_i)$.

*6.2.*It checks: $e(P, S_i) = e(P_{Pub}, Q_{PID_i,vpk_i}).\ e(vpk_{PID_i} + h_i \cdot R_i, Q)$, in which: $\sigma_i = (R_i, S_i)$. Then, the verifier outputs *true* if the equation holds. *false* otherwise.

7. *Aggregate:* When a RSU receives *n* signatures from multiple VCs with the format $\{(M_1, PID_1, vpk_{PID_1}, \sigma_1, t_1), (M_2, PID_2, vpk_{PID_2}, \sigma_2, t_2) \ldots (M_n, PID_n, vpk_{PID_n}, \sigma_n, t_n)\}$, it first performs a batch verification of those signatures to make sure that all the received signatures are valid as follows:

   *7.1.*It computes Q_{PID_i,vpk_i} and h_i for each signature.

   *7.2.*It generates a vector $u = (u_1, u_2, \ldots, u_n)$ in which each $u_i \in Z_q^*$.

   7.3. It checks the set of signatures as follows:

   $$e(P, \sum_{i=1}^{n} u_i S_i) = e(P_{pub}, \sum_{i=1}^{n} u_i Q_{PID_i,vpk_i}).\ e(\sum_{i=1}^{n} u_i (vpk_{PID_i} + h_i \cdot R_i), Q)$$

   and outputs *true* if the equation holds.

   Proof : $e(P, \sum_{i=1}^{n} u_i S_i) = e(P, \sum_{i=1}^{n} u_i (psk_{PID_i,vpk_i} + (x_{PID_i} + h_i \cdot r_i) \cdot Q))$

   $$= e(P_{pub}, \sum_{i=1}^{n} u_i Q_{PID_i,vpk_i}).\ e(\sum_{i=1}^{n} u_i (vpk_{PID_i} + h_i \cdot R_i), Q)$$

   7.4. If the batch verification passes the authentication process, the RSU aggregate the set of signatures $\{(M_1, PID_1, vpk_{PID_1}, \sigma_1, t_1), (M_2, PID_2, vpk_{PID_2}, \sigma_2, t_2) \ldots (M_n, PID_n, vpk_{PID_n}, \sigma_n, t_n)\}$, and computes the aggregate signature σ with $S = \sum_{i=1}^{n} S_i$. Hence, the RSU outputs $\{(M_1, PID_1, vpk_{PID_1}, t_1), (M_2, PID_2, vpk_{PID_2}, t_2) \ldots (M_n, PID_n, vpk_{PID_n}, t_n), \sigma = (R_1, R_2 \ldots R_n, S)\}$.

8. *Aggregate-Verify:* When other RSUs or a TCC receives an aggregate signature from an RSU aggregator with the format $\{(M_1, PID_1, vpk_{PID_1}, t_1), (M_2, PID_2, vpk_{PID_2}, t_1) \ldots (M_n, PID_n, vpk_{PID_n}, t_n), \sigma = (R_1, R_2 \ldots R_n, S)\}$, it checks if the following equation holds:

   8.1 It computes $Q_{PID_i,vpk_i} = H_1(PID_i, vpk_{PID_i})$ and $h_i = H_2(M_i, PID_i, vpk_{PID_i}, R_i, t_i)$ for $i \in [1,n]$.

   8.2 It checks: $e(P, S_i) = e(P_{Pub}, \sum_{i=1}^{n} Q_{PID_i,vpk_i}).\ e(vpk_{PID_i} + h_i \cdot R_i, Q)$ for $i \in [1, n]$, in which: $\sigma = (R_1, R_2 \ldots R_n, S)$.

   The RSU receiver or TCC outputs *true* if the equation holds. *false* otherwise.

5.8 Security proof of our CLAS

1. Adversary model

CLS scheme. In our CLS scheme, we define two types of adversaries:

Adversary A_I. This adversary models a malicious vehicle which can replace the original public key of a legitimate vehicle with a forged public key. But it does not possess the master secret key msk_{KGC}.

Adversary A_{II}. This adversary models a malicious KGC which possesses the master secret key msk_{KGC}. But it cannot perform a replacement of a public key of a legitimate vehicle.

CLAS scheme. In our CLAS scheme, two aggregator adversaries *Adversary* A_{III} and A_{IV} are considered, which can aggregate multiple signatures. In addition, they have respectively the same description as the adversaries A_I and A_{II}.

Regarding the queries that are submitted to the oracle, the adversaries A_I, A_{II}, A_{III} and A_{IV} have access to the following queries:

- *Create-Vehicle oracle.* Upon request of $PID_i \in \{0, 1\}^*$. If PID_i has already been created, then nothing to be carried out. Otherwise, the oracle executes the algorithm *Vehicle-Key-Generation* (vsk_{PID_i}, vpk_{PID_i}) to output a public key. Additionally, the algorithm *Partial-Private-Key-Extraction* (msk_{KGC}, PID_i, vpk_{PID_i})=psk_{PID_i,vpk_i}. Then, the oracle inserts (PID_i, psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i}) in the list L. In both cases, vpk_{PID_i} is returned.
- *Partial-Private-Key oracle*. Upon request of (PID_i, vpk_{PID_i}). The oracle browses the list L= (PID_i, psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i}) with the entry (PID_i, vpk_{PID_i}). If (PID_i, vpk_{PID_i}) has not been created, return $\perp$. Otherwise, the corresponding psk_{PID_i,vpk_i} is returned.
- *Secret-Key oracle.* Upon request of PID_i. The oracle browses the list L with the entry PID_i. If PID_i has not been created, return $\perp$. Otherwise, the corresponding vsk_{PID_i}.
- *Vehicle-Key-Replacement oracle.* Upon request of PID_i and a vehicle key pair (vsk'_{PID_i}, vpk'_{PID_i}). The oracle browses the list L with the entry PID_i. If PID_i has not been created, nothing to be carried out. Otherwise, the oracle updates (PID_i, psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i}) to (PID_i, psk_{PID_i,vpk_i}, vsk'_{PID_i}, vpk'_{PID_i}) in L.
- *Sign oracle.* Upon request of PID_i and $M_i \in \{0,1\}^*$, the sign oracle proceeds as follows:
  - If PID_i has been created but (vsk_{PID_i}, vpk_{PID_i}) has not been replaced. The oracle *sign* outputs a signature.
  - If PID_i has not been created, the *sign* oracle returns $\perp$.
  - If (vsk_{PID_i}, vpk_{PID_i}) of PID_i has been replaced by (vsk'_{PID_i}, vpk'_{PID_i}), the oracle returns *sign* (psk_{PID_i,vpk_i}, vsk'_{PID_i}, M_i).

Regarding security of our CLS, we consider *Game 1* and *Game 2* that are modeled by a challenger, adversaries A_I, A_{II} and the oracles. While our CLAS scheme is modeled by using *Game 3* and *Game 4* between a challenger, adversaries A_{III} and A_{IV}, and the oracles. We define the games as follows:

Game 1. *is considered between a challenger C that interacts with an adversary* A_I *as shown in Fig 4. First, the challenger runs the algorithm Setup in the setup phase to obtain params and the master secret/public key pair* (α, P_{Pub}). *Then, the challenger provides* A_I *with params but the master secret key* α *is kept secret. Second,* A_I *has access during the query phase to the following queries: Hash, Create-Vehicle, Partial-Private-Key, Secret-Key, Vehicle-Key-Replacement and Sign. After that,* A_I *outputs a message* M_i^*, *a signature* σ_i^* *that corresponds to the pseudo identity* PID_i^* *and the public key* $vpk_{PID_i^*}^*$. *We say that* A_I *wins the Game 1 if:*

1. σ_i^* *is valid forgery on* M_i^* *under* PID_i^* *and* $vpk_{PID_i^*}^*$.

2. *PID_i^* has never been submitted to the oracle Partial-Private-Key to obtain the partial private key $psk^*_{PID_i^*, vpk_i^*}$.*
3. *(PID_i^*, M_i^*) has never been submitted to the oracle Sign.*

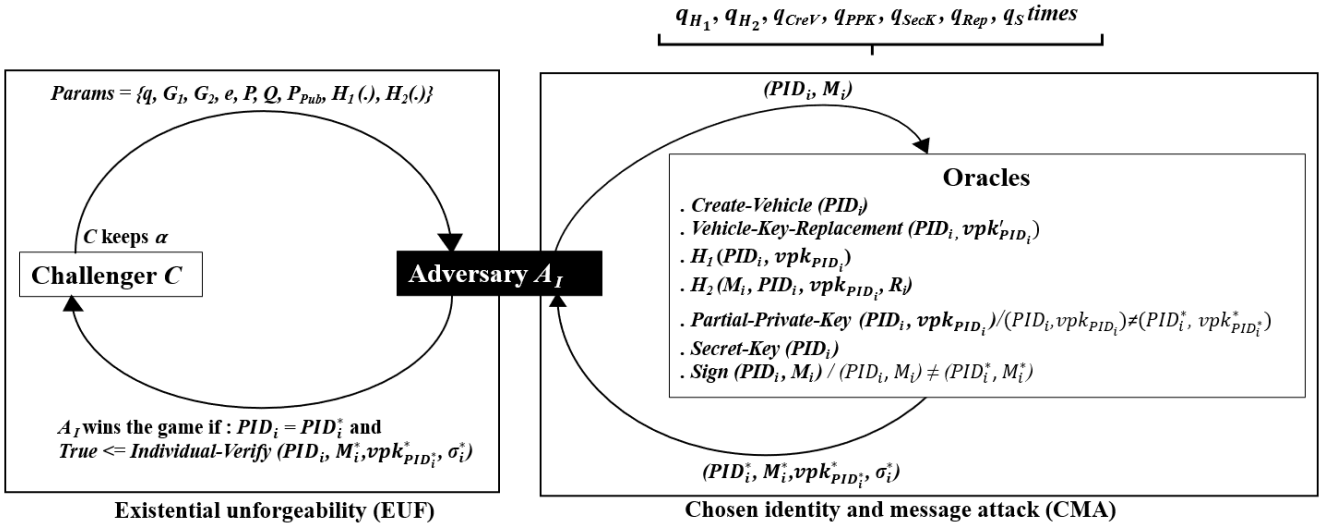


Fig 32. Security model of A_I based on EUF-CMA

Definition 1. *Our CLS scheme is said secure against A_I, if there exists no probabilistic polynomial-time in which an adversary A_I wins Game 1 with a non-negligible probability.*

Game 2. *is considered between a challenger C that interacts with an adversary A_{II} as shown in Fig 5. First, the challenger runs the algorithm Setup in the setup phase to obtain params and the master secret/public key pair (α, P_{Pub}). Then, the challenger provides A_{II} with the master secret key α and params. Second, A_{II} has access during the query phase to the following queries: Hash, Create-Vehicle, Secret-Key and Sign. In this game, A_{II} does not need to submit queries to Partial-Private-Key since A_{II}* knows *the secret α. After that, A_{II} outputs a message M_i^*, a signature σ_i^* that corresponds to the pseudo identity PID_i^* and public key $vpk^*_{PID_i^*}$. It is said that A_{II} wins the Game 2 if:*

1. *σ_i^* is a valid forgery on M_i^* under PID_i^* and $vpk^*_{PID_i^*}$.*
2. *PID_i^* has never been submitted the oracle Secret-Key to obtain the vehicle's secret key $vsk^*_{PID_i^*}$.*
3. *(PID_i^*, M_i^*) has never been submitted to the oracle Sign.*

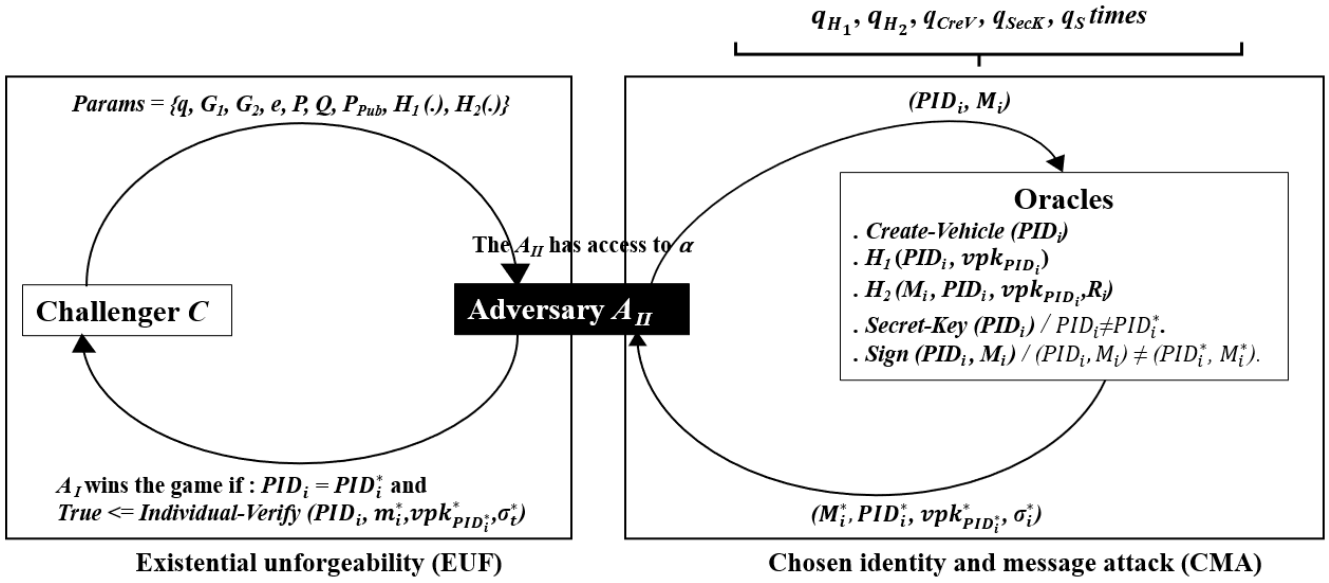


Fig 33. Security model of A_{II} based on EUF-CMA

Definition 2. *Our CLS scheme is said secure against A_{II}, if there exists no probabilistic polynomial-time in which an adversary A_{II} wins Game 2 with a non-negligible probability.*

Game 3. *is considered between a challenger C that interacts with an adversary A_{III} as shown in Fig 6. First, the challenger runs the algorithm Setup in the setup phase to obtain params and master secret/public key pair (α, P_{Pub}). Then, the challenger provides A_{III} with params but the master secret key α is kept secret. Second, A_{III} has access in the query phase to the following queries: Hash, Create-Vehicle, Partial-Private-Key, Secret-Key, Vehicle-Key-Replacement and Sign. After that, A_{III} outputs n pseudo identities $L_{PID}^* = \{ PID_1^*, PID_2^* \dots PID_n^* \}$, the corresponding public keys $L_{vpk}^* = \{ vpk_{PID_1^*}^*, vpk_{PID_2^*}^* \dots vpk_{PID_n^*}^* \}$, n messages $L_M^* = \{ M_1^*, M_2^* \dots M_n^* \}$ and an aggregate signature σ^*. It is said that A_{III} wins the Game 3 if:*

1. σ_i^* *is a valid forgery on* $\{ M_1^*, M_2^* \dots M_n^* \}$ *with the pseudo identities* $\{ PID_1^*, PID_2^* \dots PID_n^* \}$ *and the corresponding public keys* $\{ vpk_{PID_1^*}^*, vpk_{PID_2^*}^* \dots vpk_{PID_n^*}^* \}$.
2. *At least one pseudo identity has not been submitted. For instance, the pseudo identity* PID_1^* *has never been submitted to the oracle Partial-Private-Key to obtain* $psk_{PID_1^*, vpk_1^*}^*$.
3. *The oracle Sign has never been submitted with* (PID_1^*, M_1^*).

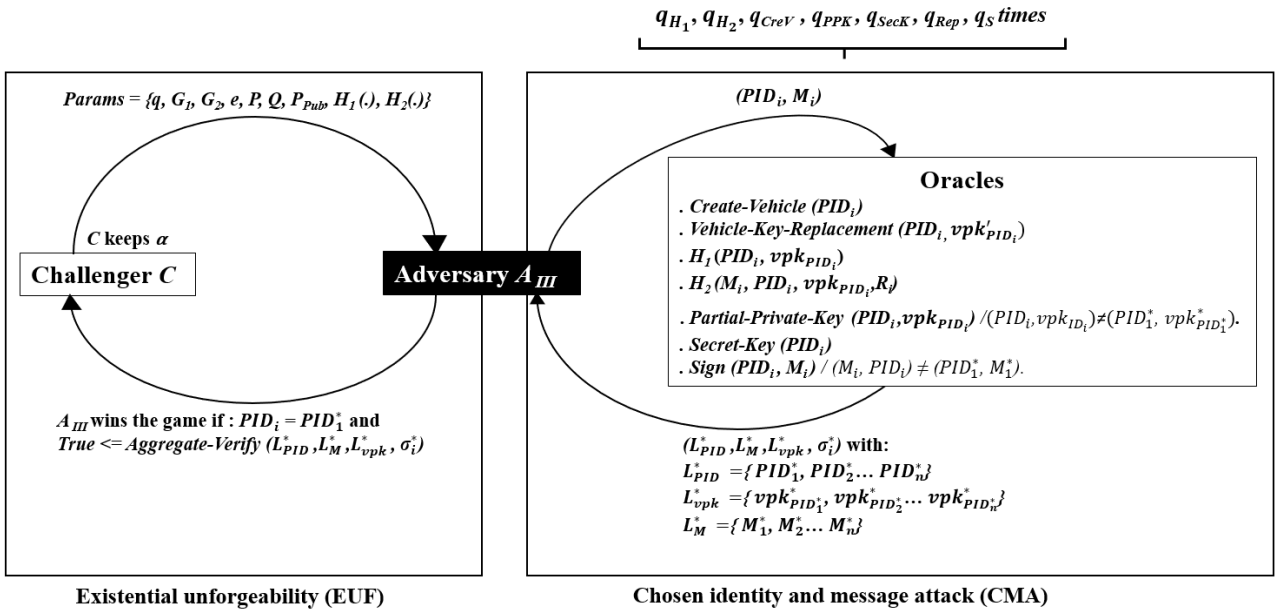


Fig 34. Security model of A_{III} based on EUF-CMA

Definition 3. *Our CLAS scheme is said secure against A_{III}, if there exists no probabilistic polynomial-time in which an adversary A_{III} wins Game 3 with a non-negligible probability.*

Game 4. *is considered between a challenger C that interacts with as adversary A_{IV} as shown in Fig 7. First, the challenger runs the algorithm Setup in the setup phase to obtain params and the master secret/public key pair (α, P_{Pub}). Then, the challenger provides A_{IV} with the master secret key α and params. Second, A_{IV} has access in the queries phase to the following queries: Hash, Create-Vehicle, Secret-Key and Sign. In this game, A_{II} does not need to submit queries to Partial-Private-Key since A_{IV}* knows *the secret α. After that, A_{IV} outputs n pseudo identities $L_{PID}^* = \{PID_1^*, PID_2^* \dots PID_n^*\}$, the corresponding public keys $L_{vpk}^* = \{vpk_{PID_1^*}^*, vpk_{PID_2^*}^* \dots vpk_{PID_n^*}^*\}$, on n messages $L_M^* = \{M_1^*, M_2^* \dots M_n^*\}$ and an aggregate signature σ^*. It is said that A_{IV} wins the Game 4 if:*

1. *σ_i^* is a valid forgery on $\{M_1^*, M_2^* \dots M_n^*\}$ with the pseudo identities $\{PID_1^*, PID_2^* \dots PID_n^*\}$ and the corresponding public keys $\{vpk_{PID_1^*}^*, vpk_{PID_2^*}^* \dots vpk_{PID_n^*}^*\}$.*
2. *At least one pseudo identity has not been submitted. For instance, the pseudo identity PID_1^* has never been submitted to the oracle Secret-Key to obtain the vehicle's secret key $vsk_{PID_1^*}^*$.*
3. *The oracle Sign has never been submitted with (PID_1^*, M_1^*).*

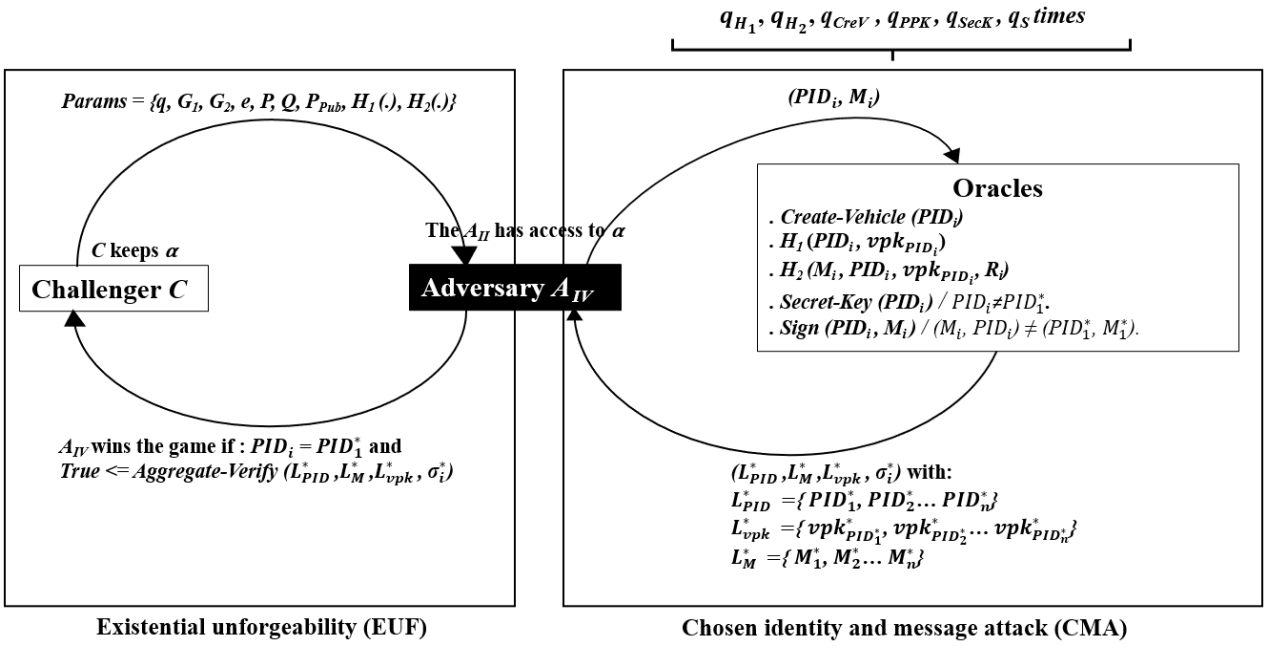


Fig 35. Security model of A_{IV} based on EUF-CMA

Definition 4. *Our CLAS scheme is said secure against A_{IV}, if there exists no probabilistic polynomial-time in which an adversary A_{IV} wins Game 4 with a non-negligible probability.*

2. Security proof

Regarding security of our CLS and CLAS. we demonstrate that our CLS scheme is EUF-CMA secure against A_I and A_{II}, and that our CLAS scheme is EUF-CMA secure against A_{III} and A_{IV} in the random oracle given the hardness of resolving CDH problem in G_1.

Theorem 1. *Our CLS scheme is existentially unforgeable against an adversary A_I. That is, if there exists an adversary A_I which can respectively submit q_{H_1} and q_{H_2} queries to the random oracles H_1, H_2, and q_{CreV} queries to the Create-Vehicle oracle, q_{PPK} queries to the Partial-Private-Key oracle, q_{SecK} queries to the Secret-Key oracle, q_{Rep} to Vehicle-Key-Replacement and q_S queries to the Sign oracle, and wins the game 1 with a probability $succA_I$, then there exists an algorithm ζ which can solve a random instance of CDH problem in G_1 in polynomial time with a success probability that is defined as follows:*

$$succ\zeta \geq \frac{1}{\varepsilon \cdot (q_{PPK} + 1)} succA_I$$

Proof. Let assume that an adversary can compromise our CLS with a non-negligible probability $SuccA_I$. If there exists an adversary A_I that can break the unforgeability of our CLS as defined in *definition 1*. Then, we can create an algorithm ζ, such that ζ uses A_I as a black box to solve a CDH problem instance $\{P, X=xP, Y=yP /x, y \in Z_q^*\} \in G_1 \times G_1 \times G_1$ by outputting xyP with a non-negligible probability $succ\zeta$, in which P is a generator in G_1 with order q and $x, y \in Z_q^*$.

Setup: For solving the CDH problem instance, ζ utilizes A_I as a black box. First, ζ simulates the environment of our CLS scheme and the oracles that A_I can access to them. Then, ζ sets $P_{pub} = x \cdot P = X$ and $Q = z \cdot P$ with $z \in Z_q^*$. After that, it provides A_I with *Params = {q, G_1, G_2, G_T, e, P, Q,*

P_{Pub}, $H_1(.)$, $H_2(.)\}$ and the CDH problem instance $\{P, X, Y\} \in G_1 \times G_1 \times G_1$ and allows A_I to run. Eventually, ζ maintains the following lists:

$L = <PID_i, psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i}>$.

$L_1 = <PID_i, w_{1i}, c_i, Q_{PID_i,vpk_i}>$.

$L_2 = <M_i, PID_i, vpk_{PID_i}, R_i, w_{2i}>$.

Query: In this phase, ζ simulates the oracles that A_I can access to them in a polynomial number of times as follows:

- *H_1 queries:* Upon receiving (PID_i, vpk_{PID_i}) from A_I, ζ flips a coin $d_i \in \{0, 1\}$ with a probability $Pr(d_i=0) = \lambda$ and a probability $Pr(d_i=1) = 1-\lambda$. Then, it chooses $w_{1i} \in Z_q^*$. If $d_i = 0$, then the hash value $H_1(PID_i, vpk_{PID_i})$ is defined as $Q_{PID_i} = w_{1i}P \in G_1$. If $d_i = 1$, ζ respond with $Q_{PID_i,vpk_i} = w_{1i}Y \in G_1$. In both cases, ζ adds $(PID_i, w_{1i}, d_i, Q_{PID_i,vpk_i})$ to the list L_1.

- *H_2 queries*: Upon receiving $(M_i, PID_i, vpk_{PID_i}, R_i)$ from A_I, ζ first browses L_2. If $(M_i, PID_i, vpk_{PID_i}, R_i)$ is in L_2, ζ responds with the previous value. Otherwise, ζ chooses $w_{2i} \in Z_q^*$ and responds with w_{2i} as the hash value of H_2 $(M_i, PID_i, vpk_{PID_i}, R_i)$ to A_I and adds the value to L_2.

- *Create-Vehicle queries:* Upon receiving PID_i:
  1) If $(PID_i, psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i})$ exists in the list L. ζ first checks whether $vpk_{PID_i} = \perp$.
     - If $vpk_{PID_i} \neq \perp$. ζ responds vpk_{PID_i} to A_I.
     - Otherwise, ζ chooses $v_i \in Z_q^*$ and $vpk_{PID_i} = v_iP$ and $vsk_{PID_i} = v_i$. Then, ζ updates $(vsk_{PID_i}, vpk_{PID_i})$ in L and responds with vpk_{PID_i} to A_I.
  2) If $(PID_i, psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i})$ doesn't exist in L. ζ sets $psk_{PID_i,vpk_i} = \perp$, and then chooses $v_i \in Z_q^*$ and sets $vpk_{PID_i} = v_iP$ and $vsk_{PID_i} = v_i$. After that, ζ adds $(PID_i, \perp, vsk_{PID_i}, vpk_{PID_i})$ in L and responds with vpk_{PID_i} to A_I.

- *Secret-Key queries:* Upon request of PID_i:
  1) If $(PID_i, psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i})$ exists in L, ζ first checks whether $vsk_{PID_i} = \perp$.
     - If $vsk_{PID_i} \neq \perp$, ζ responds vsk_{PID_i} to A_I.
     - Otherwise, ζ executes *Create-Vehicle query* to create $(vsk_{PID_i}, vpk_{PID_i}) = (v_i, v_iP)$.

     After that, ζ adds (v_i, v_iP) in L and responds with vsk_{PID_i} to A_I.
  2) If $(PID_i, psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i})$ doesn't exist in L, A_I submits a *Create-Vehicle query* and inserts $(PID_i, psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i})$ in L. Then, ζ responds with vsk_{PID_i} to A_I.

- *Partial-Private-Key queries*: Upon request of (PID_i, vpk_{PID_i}), ζ retrieves the corresponding $(PID_i, w_{1i}, d_i, Q_{PID_i})$ from L_1. If d_i=1, then ζ outputs $\perp$ and terminates the session. Otherwise, ζ browses L and responds as follows:

1) If *(PID$_i$,* psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i}) exists in *L*. ζ first checks whether (psk_{PID_i,vpk_i}, vpk_{PID_i})$\neq \perp$:
   - If (psk_{PID_i,vpk_i}, vpk_{PID_i}) $\neq \perp$, ζ responds with (psk_{PID_i,vpk_i}, vpk_{PID_i}) to A_I.
   - If (psk_{PID_i,vpk_i}, vpk_{PID_i}) = $\perp$, ζ sets psk_{PID_i,vpk_i} = $w_{1i}P_{pub}$ = $w_{1i}X \in G_1$. In addition, A_I submits a *Create-Vehicle query* to obtain to obtain (vsk_{PID_i},vpk_{PID_i}) = (v_i, v_iP). Then, ζ responds with (psk_{PID_i,vpk_i}, vpk_{PID_i}) to A_I and updates (psk_{PID_i,vpk_i}, vsk_{PID_i},vpk_{PID_i}) in *L*.

2) If *(PID$_i$,* psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i}) does not exist in *L*, ζ executes H_1 *query* on (*PID$_i$,* vpk_{PID_i}) to obtain *(PID$_i$, w_{1i}, d_i,* Q_{PID_i,vpk_i}*)* . Then, it sets psk_{PID_i,vpk_i} = $w_{1i}P_{pub}$ = $w_{1i}X \in G_1$. After that, ζ adds (*PID$_i$,* psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i}) in *L* and responds with $psk_{PID_i,vpk_i}$to A_I.

- *Vehicle-Key-Replacement queries:* Upon request of (*PID$_i$,* vpk'_{ID_i}):
  1) If *(PID$_i$,* psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i}) exists in *L*, ζ sets vpk_{PID_i}= vpk'_{PID_i} and vsk_{PID_i}=$\perp$ and adds *(PID$_i$,* psk_{PID_i,vpk_i} , $\perp$, vpk'_{PID_i}) in *L*.
  2) If *(PID$_i$,* psk_{PID_i,vpk_i},vsk_{PID_i}, vpk_{PID_i}) does not exist in *L*, ζ sets vpk_{PID_i}= vpk'_{PID_i} and vsk_{PID_i}=$\perp$ and psk_{PID_i,vpk_i}= $\perp$. After that, ζ adds *(PID$_i$,* $\perp$, $\perp$, vpk'_{PID_i}) in *L*.
- *Sign queries:* Upon request of (*M$_i$, PID$_i$*), A_I retrieves the corresponding *(PID$_i$,* psk_{PID_i,vpk_i} ,vsk_{PID_i},vpk_{PID_i}) from *L*.
  1) If *(PID$_i$,* psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i}) exists in *L*, A_I checks whether psk_{PID_i,vpk_i}=$\perp$:
     - If psk_{PID_i,vpk_i}=$\perp$, A_I submits a *Partial-Private-Key query* to create $psk_{PID_i,vpk_i}$and adds psk_{PID_i,vpk_i}in *L*.
  2) If <*PID$_i$,* psk_{PID_i,vpk_i},vsk_{PID_i},vpk_{PID_i}> does not exist in *L*, A_I looks for the corresponding <*PID$_i$, w_{1i}, d_i,* Q_{PID_i}> in L_1, sets psk_{PID_i,vpk_i} = $w_{1i}P_{pub}$ = $w_{1i}X$ and then adds <*PID$_i$,* psk_{PID_i,vpk_i} ,vsk_{PID_i}, vpk_{PID_i}> to *L*.
  3) Then, A_I checks whether vsk_{PID_i}=$\perp$. If vsk_{PID_i}=$\perp$, A_I submits *Create-Vehicle queries* to obtain (vsk_{PID_i},vpk_{PID_i}) =(v_i, v_iP) and then updates (vsk_{PID_i},vpk_{PID_i}) in *L*.

Next, ζ chooses r_i, $h_i \in Z_q^*$ and sets R_i= $r_i \cdot P$ - $h_i^{-1} \cdot vpk_{PID_i}$ and S_i = psk_{PID_i,vpk_i} + $h_i \cdot r_i \cdot Q$. Then, ζ adds *(M$_i$, PID$_i$,* vpk_{PID_i}, R_i, h_i*)* in L_2. Finally, ζ responds to A_I with a valid signature σ_i= (R_i, S_i), in which the following equation is satisfied:

$$e(P, S_i) = e(P, psk_{PID_i,vpk_i} + h_i \cdot r_i \cdot Q) = e(P, psk_{PID_i,vpk_i}) \cdot e(P, h_i \cdot r_i \cdot Q) = e(P_{pub}, Q_{PID_i,vpk_i}) \cdot e(h_i \cdot r_i \cdot P, Q)$$

$$= e(P_{pub}, Q_{PID_i}) \cdot e(h_i \cdot (h_i^{-1} \cdot vpk_{PID_i} + R_i), Q) = e(P_{pub}, Q_{PID_i}) \cdot e(vpk_{PID_i} + h_i \cdot R_i, Q)$$

After submitting *Create-Vehicle queries, Partial-Private-Key queries, Secret-Key queries, Vehicle-Key-Replacement queries, Sign queries*, A_I outputs a forged but valid signature σ_i^* = (R_i^*, S_i^*) on (PID_i^*, h_i^*, M_i^*, $vpk_{PID_i^*}^*$) with a non-negligeable probability $succA_I$. If $PID_i^* \neq PID_i$, ζ aborts. Otherwise, $PID_i^* = PID_i$*(This condition satisfies condition 1 in Game 1)*, where the

Partial-Private-Key query has never been submitted *(This condition satisfies condition 2 in Game1)* and *Sign query* has never been submitted *(This condition satisfies condition 3 in Game1)*. In this case, according to the forking lemma in [106], if ζ replays the same game in polynomial time with the same random tape and different choices of hash functions, A_I can generate another valid signature $\sigma_i'^* = (R_i^*, S_i'^*)$ on $(PID_i^*, M_i^*, h_i'^*, vpk_{PID_i^*}^*)$, in which:

$S_i^* = psk_{PID_i^*, vpk_i^*}^* + (vsk_{PID_i^*}^* + h_i^* \cdot r_i^*) \cdot Q$ and $S_i'^* = psk_{PID_i^*, vpk_i^*}^* + (vsk_{PID_i^*}^* + h_i'^* \cdot r_i^*) \cdot Q$. Then, A_I retrieves the corresponding $(PID_i^*, w_{1i}^*, d_i^*, Q_{PID_i^*, vpk_i^*}^*)$ and $(PID_i^*, psk_{PID_i^*, vpk_i^*}^*, vsk_{PID_i^*}^*, vpk_{PID_i^*}^*)$ respectively from L_1 and L. If $d_i^* = 0$, then A_I halts and fails. Otherwise $d_i^* = 1$. Then, A_I computes: $(h_i^*)^{-1} \cdot S_i^* - (h_i'^*)^{-1} \cdot S_i'^* = ((h_i^*)^{-1} - (h_i'^*)^{-1}) \cdot [x(w_{1i}^* yP) + vsk_{PID_i^*}^*(zP)]$

Finally, A_I outputs xyP as a solution to the CDH instance by calculating:

$xyP = (w_{1i}^*)^{-1}\{[(h_i^*)^{-1} S_i^* - ((h_i'^*)^{-1} S_i'^*)\ ((h_i'^*)^{-1} - (h_i'^*)^{-1})^{-1}] - vsk_{PID_i^*}^*(zP)\}$

Regarding the probability with which ζ wins the Game1 and solves the CDH instance problem, we consider the following events:

Ev1: ζ does not abort during *Partial-Private-Key queries*.

Ev2: σ^* is a valid forgery on $(PID_i^*, M_i^*, vpk_{PID_i^*}^*)$.

Ev3: σ^* satisfies $PID_i^* = PID_i$ with $d_i^* = 1$.

The probability that ζ solves the given CDH instance is: $succ\zeta = Pr[Ev1 \cap Ev2 \cap Ev3] = Pr[Ev1] \cdot Pr[Ev2|Ev1] \cdot Pr[E3|Ev1 \cap Ev2]$. In addition, the probabilities *Ev1*, *Ev2* and *Ev3* can be defined as follows:

- *Ev1:* ζ does not abort during *Partial-Private-Key queries*. This happens with a probability $Pr[Ev1] \geq \lambda^{q_{PPK}}$.

Proof. *Pr* $(d_i=0) = \lambda$ for one a *Partial-Private-Key query*. Thus, it takes q_{PPK} times to submit *Partial-Private-Key queries*. As a result, ζ does not abort with a probability $Pr[Ev2|Ev1] \geq. \lambda^{q_{PPK}}$.

- *Ev2:* This probability reflects the probability of success of A_I.

Proof. $Pr[Ev2|Ev1] = succA_I$.

- *Ev3:* This event happens after that A_I outputs a valid forgery and ζ does not abort. This happens with a probability $Pr[Ev3|Ev1 \cap Ev2] \geq 1-\lambda$.

Proof. Suppose that *Ev1* and *Ev2* have occurred. Then, A_I creates a valid forgery on PID_i^* and satisfies $PID_i^* = PID_i$ with $d_i^* = 1$. Therefore, $Pr[Ev3|Ev1 \cap Ev2] \geq 1-\lambda$.

As a result, $succ\zeta = (1-\lambda) \cdot \lambda^{q_{PPK}} \cdot succA_I$ with the maximum value: $\lambda_{max} = \frac{q_{PPK}}{q_{PPK}+1}$. Thus:

$succ\zeta \geq (1 - \frac{q_{PPK}}{q_{PPK}+1}) \cdot (\frac{q_{PPK}}{q_{PPK}+1})^{q_{PPK}} \cdot succA_I$. Plus, $succ\zeta \geq \frac{1}{\varepsilon \cdot (q_{PPK}+1)} succA_I$ with $(\frac{q_{PPK}}{q_{PPK}+1})^{q_{PPK}} = \frac{1}{\varepsilon}$. Consequently, ζ can solve a CDH problem instance with a non-negligible probability $succ\zeta$ since $succA_I$ is assumed to be non-negligible. As a result, that contradicts the assumption of solving a CDH instance problem in a polynomial number of times.

Theorem 2*. Our CLS scheme is existentially unforgeable against an adversary A_{II}. That is, if there exists an adversary A_{II} which can respectively submit q_{H_1} and q_{H_2} queries to the random oracles H_1, H_2, and q_{CreV} queries to the Create-Vehicle oracle, q_{SecK} queries to the Secret-Key oracle, and q_S queries to the Sign oracle, and win the game 2 with a probability $succA_{II}$. Then, there is an algorithm ζ which can solve a random instance of CDH problem in G_1 in polynomial time with the success probability that is defined as follows:* $succ\zeta \geq \frac{1}{\varepsilon \cdot (q_{SecK}+1)} \cdot succA_{II}$

Proof. Let assume that an adversary A_{II} can compromise our CLS with a non-negligible probability $SuccA_{II}$. If there exists an adversary A_{II} who can break the unforgeability of our CLS scheme as defined in *definition 2*. Then, we can create an algorithm ζ, such that ζ uses A_{II} as a black box to solve a CDH problem instance $\{P, X=xP, Y = yP\}$ $G_1 \times G_1 \times G_1$ by outputting xyP with a non-negligible probability $succ\zeta$, in which P is a generator in G_1 with an order q and $x, y \in Z_q^*$.

Setup: For solving the instance of CDH problem, ζ utilizes A_{II} as a black box. First, ζ simulates the environment of our CLS scheme and the oracles that A_{II} can access to them. ζ sets $P_{pub} = \alpha \cdot P$ and $Q = x \cdot P = X$. Then, it provides A_{II} with $Params = \{q, G_1, G_2, G_T, e, P, Q, P_{Pub}, H_1(.), H_2(.)\}$ and the CDH problem instance $\{P, X, Y\}$ $G_1 \times G_1 \times G_1$ and allows A_{II} to run. In this case, $psk_{PID_i,vpk_i} = \alpha \cdot Q_{PID_i,vpk_i} = \alpha \cdot H_1(PID_i, vpk_{PID_i})$ can be computed by both ζ and A_{II}. Eventually, ζ contains the following lists:

$L = \langle PID_i, vsk_{PID_i}, vpk_{PID_i}, d_i \rangle$.

$L_1 = \langle M_i, PID_i, vpk_{PID_i}, R_i, w_{2i}, H_2 \rangle$.

Query: In this phase, ζ simulates the oracles that A_{II} can access to them in a polynomial number of times as follows:

- *Create-Vehicle oracle.* Upon request of PID_i:
  1) If PID_i exists in L, ζ responds with vpk_{PID_i} to A_{II}.
  2) If PID_i does not exist in L, ζ flips a coin $d_i \in \{0, 1\}$ with a probability $Pr(d_i=0) = \lambda$ and $Pr(d_i=1) = 1-\lambda$. Then, it chooses $w_{1i} \in Z_q^*$. If $d_i = 0$, then vpk_{ID_i} is defined as $Q_{PID_i,vpk_i} = w_{1i}P \in G_1$. If $d_i = 1$, then ζ responds with $w_{1i}Y \in G_1$. In both cases, ζ sets $vsk_{PID_i} = w_{1i}$ and adds $(PID_i, vsk_{PID_i}, vpk_{PID_i}, d_i)$ in L. Then, ζ responds with vpk_{PID_i} to A_{II}.
- *Secret-Key oracle.* Upon request of PID_i:
  1) If $(PID_i, vsk_{PID_i}, vpk_{PID_i}, d_i)$ exists in L. In case $d_i = 0$, ζ responds with vsk_{PID_i} to A_{II}. Otherwise, $d_i = 1$, ζ outputs $\perp$ and terminates the session.
  2) If $(PID_i, vsk_{PID_i}, vpk_{PID_i}, d_i)$ does not exists in L, ζ executes *Create-Vehicle query* and adds $(PID_i, vsk_{PID_i}, vpk_{PID_i}, d_i)$ in L. Then, ζ responses with vsk_{PID_i} to A_{II}.
- H_2 *queries*: Upon request of $(M_i, PID_i, vpk_{PID_i}, R_i)$ from A_{II}, ζ first browses L_1. If H_2 has already been submitted. Then, ζ responds with the previous value. Otherwise, ζ chooses $w_{2i} \in Z_q^*$ and responds with $H_2 = w_{2i}$ as the hash value of H_2 $(M_i, PID_i, vpk_{PID_i}, R_i)$ to A_{II} and adds the value to L_1.
- *Sign queries:* The same model is considered as the one in *theorem 1*.

After submitting *Create-Vehicle queries, Secret-Key queries,* and *Sign queries, A_{II}* outputs a forged but valid signature $\sigma_i^* = (R_i^*, S_i^*)$ on $(PID_i^*, M_i^*, vpk_{PID_i^*}^*)$ with a non-negligeable probability *succA_{II}*. If $PID_i^* \neq PID_i$, ζ aborts. Otherwise, $PID_i^* = PID_i$*(This condition satisfies condition 1 in Game 2).* By replaying the same game, we assume that A_{II} can output another valid signature $\sigma_i'^* = (R_i^*, S_i'^*)$ on $(PID_i'^*, M_i'^*, vpk_{PID_i'^*}'^*)$. Eventually, A_{II} retrieves the corresponding $(PID_i^*, vsk_{PID_i^*}^*, vpk_{PID_i^*}^*, d_i^*)$ and $(PID_i'^*, vsk_{PID_i^*}'^*, vpk_{PID_i'^*}'^*, d_i'^*)$ from *L*, and $(M_i^*, PID_i^*, vpk_{PID_i^*}^*, R_i^*, w_{2i}^*, H_2^*)$ and $(M_i'^*, PID_i'^*, vpk_{PID_i'^*}'^*, R_i^*, w_{2i}^*, H_2^*)$ from L_1, with $d_i^* = 1$ and $d_i'^* = 1$, and where the *Secret-Key query* has never been submitted *(This condition satisfies condition 2 in Game2)* and *Sign query* has never been submitted on $(PID_i^*, M_i^*, vpk_{PID_i^*}^*)$ and $(PID_i'^*, M_i'^*, vpk_{PID_i'^*}'^*)$ *(This condition satisfies condition 3 in Game2)*. In addition, the forged signatures should satisfy the following equations:

$e(P, S_i^*) = e(P_{Pub}, Q_{PID_i^*, vpk_i^*}^*) \cdot e(vpk_{PID_i^*}^* + h_i^* \cdot R_i^*, Q)$ and $e(P, S_i'^*) = e(P_{Pub}, Q_{PID_i'^*, vpk_i'^*}'^*) \cdot e(vpk_{PID_i'^*}'^* + h_i^* \cdot R_i^*, Q)$

Where: $Q_{PID_i^*, vpk_i^*}^* = H_1(PID_i^*, vpk_{PID_i^*}^*)$ and $Q_{PID_i'^*, vpk_i'^*}'^* = H_1(PID_i'^*, vpk_{PID_i'^*}'^*)$. If ζ does not abort, we obtain:

$e(vpk_{PID_i^*}^* + h_i^* \cdot R_i^*, Q) = e(P, S_i^*) \cdot (e(P_{Pub}, Q_{PID_i^*, vpk_i^*}^*))^{-1}$

and $e(vpk_{PID_i'^*}'^* + h_i^* \cdot R_i^*, Q) = e(S_i'^*, P) \cdot (e(P_{Pub}, Q_{PID_i'^*, vpk_i'^*}'^*))^{-1}$.

Considering $Q_{PID_i^*, vpk_i^*}^* = H_1(PID_i^*, vpk_{PID_i^*}^*)$, $Q_{PID_i'^*, vpk_i'^*}'^* = H_1(PID_i'^*, vpk_{PID_i'^*}'^*)$, $h_i^* = w_{2i}^*$ and $Q = X$ and $vpk_{PID_i^*}^* = vsk_{PID_i^*}^* Y = z_{1i}^* Y$, $vpk_{PID_i'^*}'^* = vsk_{PID_i'^*}'^* Y = z_{1i}'^* \cdot Y$,. We can deduce that ζ solves the given CDH problem with a probability *succ*ζ as follows: $xyP = (z_{1i}^* - z_{1i}'^*)^{-1} \cdot (S_i^* - S_i'^*) - \alpha \cdot (Q_{PID_i^*, vpk_i^*}^* - Q_{PID_i'^*, vpk_i'^*}'^*)$

Regarding the probability with which ζ wins the Game2 and solves the CDH instance problem, we consider the following events:

Ev1: ζ does not abort during *Secret-key queries.*

Ev2: σ^* are valid forgery on $(PID_i^*, M_i^*, vpk_{PID_i^*}^*)$.

Ev3: σ^* satisfies $PID_i^* = PID_i$ with $d_i^* = 1$.

The probability of success that ζ solves the CDH instance is: *succ*ζ = *Pr*[*Ev*1 ∩ *Ev*2 ∩ *Ev*3] =*Pr*[*Ev*1] · *Pr*[*Ev*2|*Ev*1] · *Pr*[*Ev*3|*Ev*1∩*Ev*2]. In addition, the probabilities of the events *Ev1*, *Ev2* and *Ev3* can be defined as follows:

• *Ev1:* ζ does not abort during *Secret-key queries*. This happens with a probability *Pr* [*Ev1*] ≥ $\lambda^{q_{SecK}}$.

Proof. *Pr* (d_i=0) = λ for one *Secret-key query*. Thus, it takes q_{SecK} times to submit *Secret-key queries.* As a result, ζ does not abort with a probability *Pr*[*Ev*2|*Ev*1] ≥. $\lambda^{q_{SecK}}$.

• *Ev2:* This probability reflects the probability of success of A_{II}.

Proof. *Pr*[*Ev*2|*Ev*1] = *succA_{II}.*

- *Ev3:* Suppose that *Ev1* and *Ev2* have occurred. Then, A_{II} creates a valid forgery on PID_i^* and satisfies $PID_i^* = PID_i$ with $d_i^* = 1$. Therefore, $Pr[Ev3|Ev1 \cap Ev2] \geq 1-\lambda$.

Proof. Suppose that *Ev1* and *Ev2* have occurred. Then, A_{II} generates a valid forgery on PID_i^* and satisfies $PID_i^* = PID_i$ with $d_i^* = 1$. As a result, $Pr[Ev3|Ev1 \cap Ev2] \geq 1 - \lambda$.

As a result, $succ\zeta = \lambda^{q_{SecK}} \cdot (1-\lambda) \cdot succA_{II}$ with the maximum value: $\delta_{max} = \frac{q_{SecK}}{q_{SecK}+1}$. Thus:

$succ\zeta \geq (\frac{q_{SecK}}{q_{SecK}+1})^{q_{SecK}} \cdot (1 - \frac{q_{SecK}}{q_{SecK}+1}) \cdot succA_{II}$. Plus, $succ\zeta \geq \frac{1}{\varepsilon \cdot (q_{SecK}+1)} \cdot succA_{II}$ with $(\frac{q_{SecK}}{q_{SecK}+1})^{q_{SecK}} = \frac{1}{\varepsilon}$. Consequently, ζ can solve a CDH problem instance with a non-negligible probability $succ\zeta$ since $succA_{II}$ is assumed to be non-negligible. As a result, that contradicts the assumption of solving a CDH instance problem in a polynomial number of times.

Theorem 3. *Our CLAS scheme is considered secure against adversaries A_{III} and A_{IV}, if our CLS scheme is secure against adaptive chosen message attacks in the chosen aggregate model.*

Proof. Let's assume that a forger *Adv* can break our CLAS scheme. Then, we can create an algorithm ζ, such that ζ uses *Adv* as a black box and outputs a forgery of our CLS.

Setup: First, ζ chooses P_{pub}, $Q \in G_1$. Then, ζ simulates the environment that *Adv* can access to it. In addition, ζ contains the following list:

$L = (PID_i, psk_{PID_i,vpk_i}, vsk_{PID_i}, vpk_{PID_i})$.

$L_1 = (PID_i, w_{1i}, c_i, Q_{PID_i,vpk_i})$.

$L_2 = (M_i, PID_i, vpk_{PID_i}, R_i, w_{2i})$.

Query: In this phase, ζ simulates the oracles that *Adv* can access to them in a polynomial number of times as follows:

➢ *H_1 queries:* Upon request of (PID_i, vpk_{PID_i}) from *Adv*, ζ tosses a coin $d_i \in \{0, 1\}$ with a probability $Pr(d_i=0) = \lambda$ and $Pr(d_i=1) = 1-\lambda$. Then, it chooses $w_{1i} \in Z_q^*$. If $d_i = 0$, ζ responded with the hash value $H_1(PID_i, vpk_{PID_i})$ that is defined as follows $Q_{PID_i,vpk_i} = w_{1i} \cdot P \in G_1$. If $d_i = 1$, then ζ responded with $Q_{PID_i,vpk_i} = w_{1i} \cdot P_{pub} \in G_1$. In both cases, ζ adds $(PID_i, w_{1i}, d_i, Q_{PID_i,vpk_i})$ to L_1.

Eventually, *Adv* outputs *n* signatures with the corresponding pseudo identities $L_{PID}^* = \{PID_i^*, PID_i^* \dots PID_n^*\}$ and public keys $L_{vpk}^* = \{vpk_{PID_1^*}^*, vpk_{PID_2^*}^* \dots vpk_{PID_n^*}^*\}$, messages $L_M^* = \{M_1^*, M_2^* \dots M_n^*\}$ and an aggregate signature $\sigma^* = \{R_1^*, R_2^* \dots R_n^*, S^*\}$. Then, ζ looks for the corresponding *n* tuples in $(PID_i, w_{1i}, d_i, Q_{PID_i,vpk_i})$ for $i \in [1,n]$ in L_1. ζ considers tuples with: $d_t = 1$ and $d_{t'} = 0$ for $t' \in [1,n]$ and $t' \neq t$. Note that *Adv* has never submitted $(M_t^*, PID_t^*, vpk_{PID_t^*}^*, R_t^*)$ to the *sign oracle*. Otherwise, ζ fails and aborts *(This condition satisfies conditions 2 and 3 in both Games 3 and 4). Adv* succeeds to output a forged signature when $Q_{PID_t,vpk_t} = w_{1t} \cdot P_{pub}$ and $Q_{PID_{t'},vpk_{t'}} = w_{1t'} \cdot P$ for $t' \in [1,n]$ and $t' \neq t$. The aggregate signature $\sigma^* = \{R_1^*, R_2^* \dots R_n^*, S^*\}$ should satisfy the following equation: $e(P, S^*) = e(P_{pub}, \sum_{i=1}^n Q_{PID_i^*,vpk_i^*}^*). e(\sum_{i=1}^n vpk_{PID_i^*}^* + h_i^* \cdot R_i^*), Q)$ and $S^* = \sum_{i=1,}^n S_i^*$

After that, ζ looks for the corresponding $(M_i^*, PID_i^*, vpk_{PID_i^*}^*, R_i^*, w_{2i}^*)$ in L_2 and $(PID_i^*, psk_{PID_i^*,vpk_i^*}^*, vsk_{PID_i^*}^*, vpk_{PID_i^*}^*)$ in L. Considering that $S^* = w_{1i}^* \cdot P_{pub}$ and $e(P, S_i^*) = e(P_{pub}, Q_{PID_i^*,vpk_i^*}^*)$ for $i \in [1,n]$ and $i \neq t$, ζ inserts a forged signature S'^* as follows:

$S'^* = S^* - \sum_{i=1,i\neq t}^{n} S_i^* = [psk_t^* + \sum_{i=1,i\neq t}^{n} psk_i^* + \sum_{i=1}^{n} vsk_{PID_i^*}^* + w_{2i}^* \cdot r_i^*) \cdot Q] - \sum_{i=1,i\neq t}^{n} psk_i^* = psk_t^* + \sum_{i=1}^{n} (vsk_{PID_i^*}^* + w_{2i}^* \cdot r_i^*) \cdot Q)$

After that, ζ chooses $r_i^* \in Z_q^*$ for $i \in [1,n]$ and computes $R_i^* = r_i^* \cdot P$. In addition, ζ chooses $h_t^* \in Z_q^*$ and computes $R_i'^* = (h_t^*)^{-1} \cdot \sum_{i=1}^{n} w_{2i}^* \cdot R_i^*$. Then, ζ computes $vpk'^*_{PID_t'^*} = \sum_{i=1}^{n} vpk_{PID_i^*}^*$. Eventually, ζ replaces $vpk_{PID_t^*}^*$ by $vpk'^*_{PID_t'^*}$ using the query *Vehicle-Key-Replacement oracle.* Additionally, ζ sets the hash value $H_2(M_t^*, PID_t^*, vpk_{PID_t^*}^*, R_t^*) = h_t^*$. As a result, (R'^*, S'^*) is a valid forgery signature on $(M_t^*, PID_t^*, vpk'^*_{PID_t^*})$ that verifies the following equation:

$e(P_{pub}, Q_{PID_t^*,vpk_t^*}^*) \cdot e(vpk'^*_{PID_t'^*} + h_t^* \cdot R'^*), Q) = e(P_{pub}, Q_{PID_t^*,vpk_t^*}^*) \cdot e(\sum_{i=1}^{n} vsk_{PID_i^*}^* \cdot P + h_t^* \cdot (h_t^*)^{-1} \sum_{i=1}^{n} w_{2i}^* \cdot R_i^*), Q)$

$= e(P_{pub}, Q_{PID_t^*,vpk_t^*}^*) \cdot e(\sum_{i=1}^{n} vsk_{PID_i^*}^* \cdot P + \sum_{i=1}^{n} w_{2i}^* \cdot R_i^*), Q)$

$= e(P, psk_{PID_t^*,vpk_t^*}^*) \cdot e(P, \sum_{i=1}^{n} vsk_{PID_i^*}^* \cdot Q + h_t^* \cdot (h_t^*)^{-1} \sum_{i=1}^{n} w_{2i}^* \cdot r_i^*) \cdot Q)$

$= e(P, psk_{PID_t^*,vpk_t^*}^*) \cdot e(P, \sum_{i=1}^{n} (vsk_{PID_i^*}^* + \sum_{i=1}^{n} w_{2i}^* \cdot r_i^*)) \cdot Q) = e(P, S'^*)$

Finally, ζ outputs a valid forgery (R'^*, S'^*) *(This condition satisfies condition 1 in Game3 and 4).* Regarding the probability with which ζ succeed to outputs a forged signature with a probability $succ\zeta$, the following events are considered:

Ev1: ζ does not abort during *Partial-Private-Key queries*.

Ev2: $\sigma'^* = (R'^*, S'^*)$ is a valid forgery.

Ev3: The aggregate signature σ^* satisfies $PID_i = PID_t^*$ with $d_t^* = 1$ and $PID_i = PID_{t'}^*$ with $d_{t'}^* = 0$ for $t' \in [1,n]$ with $t' \neq t$.

The probability of success that ζ solves the CDH instance is: $succ\zeta = Pr[Ev1 \cap Ev2 \cap Ev3] = Pr[Ev1] \cdot Pr[Ev2|Ev1] \cdot Pr[Ev3|Ev1 \cap Ev2]$. The probabilities that reflect the above events are defined as follows:

- *Ev1:* for one *Partial-Private-Key query*. This event happens with a probability $Pr[Ev1] \geq \lambda^{q_{PPK}}$.

Proof. *Pr* $(d_i = 0) = \lambda$ for one a *Partial-Private-Key query*. Thus, it takes q_{PPK} times to submit *Partial-Private-Key queries.* As a result, ζ does not abort with a probability $Pr[Ev2|Ev1] \geq \lambda^{q_{PPK}}$.

- *Ev2:* ζ does not abort during *sign queries* and *Partial-Private-Key queries*. This probability reflects the probability of success of *Adv.*

Proof. $Pr[Ev2|Ev1] = succAdv.$

- *Ev3:* This event happens after that *Adv* outputs a valid forgery and ζ does not abort. This happens with a probability $Pr\,[Ev3|Ev1 \cap Ev2] \geq (1-\lambda) \cdot \lambda^{n-1}$

Proof. Suppose that *Ev1* and *Ev2* have occurred. Then, *Adv* generates a valid forgery on PID_t^* with $d_t^*= 1$ and $PID_i=PID_{t'}^*$ with $d_{t'}^*= 0$ for $t' \in [1,n]$. This happens with a probability $(1-\lambda) \cdot \lambda^{n-1}$.

As a result, $succ\zeta = \lambda^{q_{PPK}} \cdot (1-\lambda) \cdot \lambda^{n-1} \cdot succAdv = (1-\lambda) \cdot \lambda^{q_{PPK}+n-1} \cdot succAdv$ with the maximal value:

$\delta_{max}= 1 - \frac{1}{1+q_{PPK}+n}$. Thus, $succ\ \zeta \geq \frac{1}{1+q_{PPK}+n} \cdot (1 - \frac{1}{1+q_{PPK}+n})^{q_{PPK}+n-1} \cdot succAdv$. Plus, $succ\ \zeta \geq \frac{1}{\varepsilon \cdot (1+q_{PPK}+n)} \cdot succAdv$ with $(1 - \frac{1}{1+q_{PPK}+n})^{q_{PPK}+n-1} = \frac{1}{\varepsilon}$. Consequently, ζ can solve a CDH problem instance with a non-negligible probability $succ\zeta$ since *succAdv* is assumed to be non-negligible. As a result, that contradicts the assumption of solving a CDH instance problem in a polynomial number of times.

5.9 Simulation and performance evaluation of our CLAS

In this section, we analyse the performance of our CLAS scheme and several schemes [40], [41], [65], [69], [71], [72], [107] that are applied in VANET. In this regard, we compare our scheme with those schemes regarding time execution during the "Sign phase ", "Individual-Verify phase ", and "Aggregate-Verify phase ". During V2V communication, a VC signs its safety-related message and transmits it to other VCs or RSU in the network. The execution time of the signing process is reflected as "Sign phase" and the verification process is reflected as: "Individual-Verify" as shown in Table 13. In V2I communication, a RSU has the capability to combine numerous signatures it receives from VCs within its vicinity into a single aggregate signature. Subsequently, it transmits this aggregate signature to other RSUs. These latter entities then validate the aggregate signature through a verification process known as the "Aggregate-Verify phase". Regarding time execution, we adopted the same computation evaluation that is carried out in [108] by using MIRACL cryptographic library. In this regard, the computation method is running on an Intel i7 3.07 GHz CPU and uses a Tate pairing over MNT curve, with p =159-bit subgroup of order q = 158-bit, an embedding degree k=6 and where the equation of the curve is defined as follows: $E(F_p) : y^2 = x^3 - 3x + B$, with $B \in F_p$ and $k=6$, $h=3$ and $D=62003$ as mentioned in [109], providing a SL of 80-bit. Plus, the elements in G_1 take 160-bit, the elements in G_2 take in compact representation 320-bit and G_T take 954-bits [110]. The computation time of the different cryptographic computations can be defined as follows:

T_{bp}: Time that takes to perform an $e(\bar{P}, \bar{Q})$, such that $\{\bar{P}, \bar{Q}\} \in G_1 \times G_2$: 3.21 ms.

$T_{bp \cdot m}$: Time that takes to perform a $x \cdot \bar{P}$ in BP, such that $\bar{P} \in G_1$, $\bar{x} \in Z_q^*$: 0.39 ms.

T_{MTP}: Time that takes to perform a MTH function operation: 0.09 ms.

The Table 13 reflects the execution time during "Sign phase", "Individual-Verify phase ", and "Aggregate-Verify phase ". Additionally, the Figures 32,33 and 34 respectively reflect the "Sign phase ", "Individual-Verify phase ", and "Aggregate-Verify phase ", in which n is the number of signatures that are verified by a RSU or a TCC.

Table 13. Execution time analysis of our CLAS

Scheme	Sign phase (ms)	Individual-Verify phase (ms)	Aggregate-Verify phase (ms)
Malhi and Batra [40]	$3T_{bp\cdot m}$	$3T_{bp}+3T_{bp\cdot m}$	$3T_{bp}+3nT_{bp\cdot m}$
Kumar and Sharma [71]	$3T_{bp\cdot m}$	$3T_{bp}+3T_{bp\cdot m}$	$3T_{bp}+3nT_{bp\cdot m}$
Zhong et al. [41]	$3T_{bp\cdot m}$	$3T_{bp}+2T_{bp\cdot m}+T_{MTP}$	$3T_{bp}+2nT_{bp\cdot m}+nT_{MTP}$
Kamil and Ogundoyin [65]	$3T_{bp\cdot m}$	$3T_{bp}+2T_{bp\cdot m}+T_{MTP}$	$3T_{bp}+2nT_{bp\cdot m}+nT_{MTP}$
Mei et al. [69]	$4T_{bp\cdot m}+2\ T_{MTP}$	$4T_{bp}+2T_{bp\cdot m}$	$4T_{bp}+2nT_{bp\cdot m}$
Cahyadi et al. [72]	$3T_{bp\cdot m}$	$3T_{bp}+2T_{bp\cdot m}$	$3T_{bp}+2nT_{bp\cdot m}$
Our scheme	$2T_{bp\cdot m}$	$3T_{bp}+T_{bp\cdot m}+T_{MTP}$	$3T_{bp}+nT_{bp\cdot m}+nT_{MTP}$

The Figure, 32, 33 and 34 depict the results from the table 13.

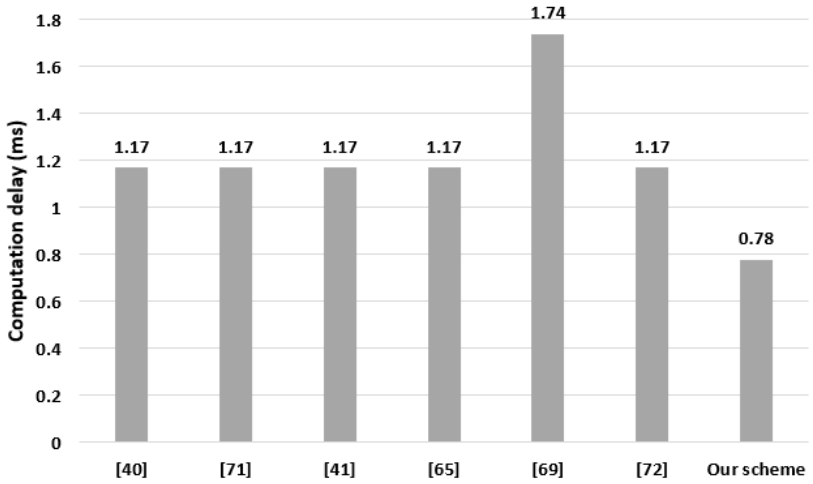


Fig 36. Execution Time during the signing phase for a Single safety-related message

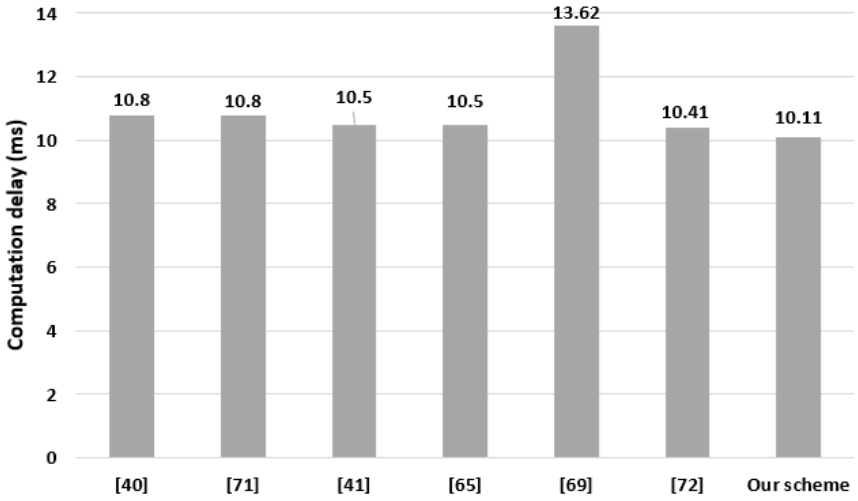


Fig 37. Execution Time during the Individual-Verify Phase

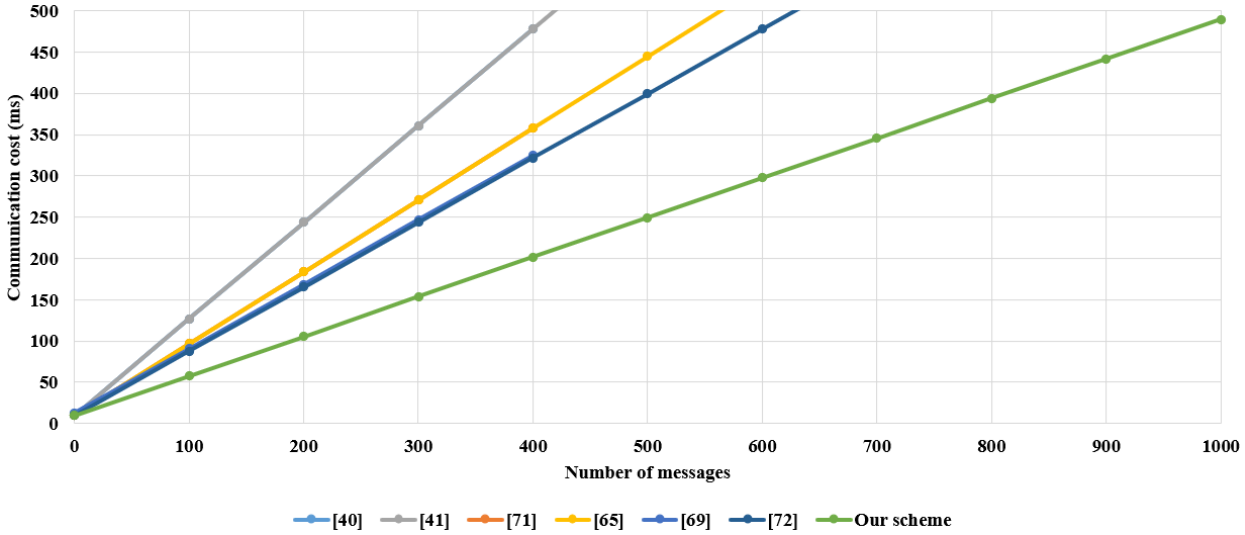


Fig 38. Execution time during the Aggregate-Verify phase for multiple safety-related messages

5.10 Communication cost

In this section, the communication cost is evaluated when safety-related messages are successfully exchanged during V2V and V2I communications. The parameter sizes used in the authentication process are the same as those considered in [67], where the sizes of a cyclic additive group G_1 and a multiplicative group G_2 are 128 and 40 bytes, respectively. The hash function has an output of 20 bytes, the timestamp has 4 bytes, and a safety-related message has a length of 67 bytes according to the IEEE Trial-Use standard.

Table 14. Overhead comparison between existing CLAS schemes with our CLAS scheme

Scheme	Single safety-related message	*n* safety-related messages
Malhi and Batra [40]	727 bytes	727n bytes
Kumar and Sharma [71]	727 bytes	727n bytes
Zhong et al. [41]	715 bytes	715n bytes
Kamil and Ogundoyin [65]	715 bytes	715n bytes
Mei et al. [69]	735 bytes	735n bytes
Cahyadi et al. [72]	583 bytes	583n bytes
Notre protocole CLAS	607 bytes	607n bytes

The figure 35 depicts the results from the figure 14.

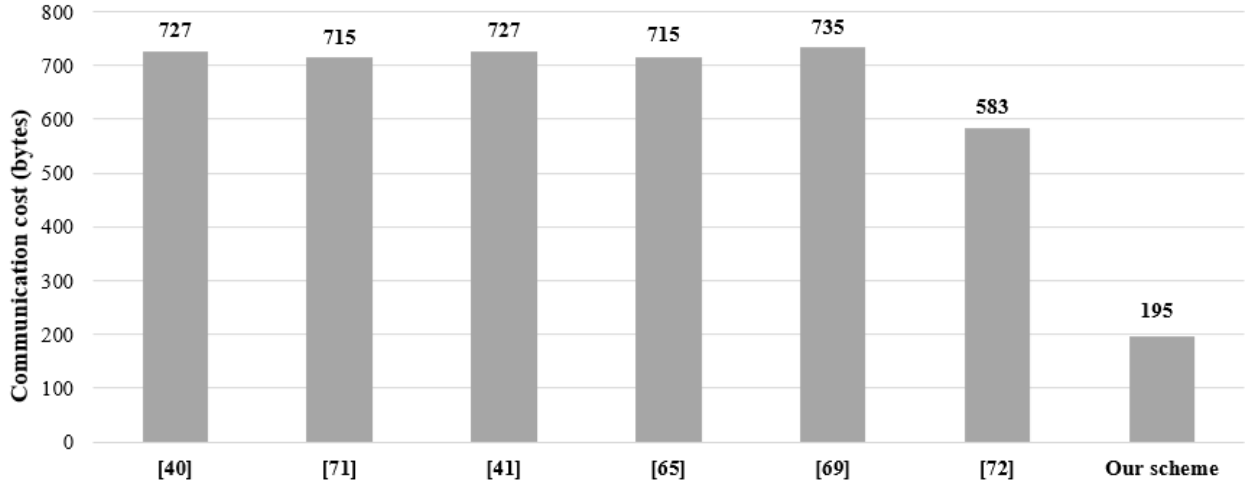


Fig 39. Overhead of aggregate-Verify phase of multiple safety-related messages

5.11 Conclusion Regarding Our CLAS Protocol

This work proposes a new CL-based Aggregate Signature (CLAS) protocol with a TL 3, providing strong non-repudiation in a vehicular ad hoc network. In this context, it can be proven that the TA or an unauthorized VC has issued a fake *pk*. Furthermore, only *pk*s that are certified by the TA can successfully pass the authentication process. Therefore, our CLAS protocol is resilient against PK-Replacement attacks, as the original *pk* cannot be replaced by an adversary, and only a legitimate VC can use it during the authentication process. Regarding the monitoring strategy in our model system, the TA can detect an adversary via both identifiers: its identity and *pk*. A security analysis is conducted, showing that our CLAS protocol is EUF-CMA, considering the Computational Diffie-Hellman problem's difficulty. Moreover, our protocol provides an authentication process for both V2V and V2I communications. When an RSU receives multiple safety-related messages from VCs, it performs a batch verification to ensure that all signatures are valid, reducing computation time compared to verifying safety-related messages individually. After that, the RSU aggregates the received signatures into a single aggregate signature and sends it to other RSUs and a TCC. Regarding the performance evaluation of our CLAS protocol, it turns out that our CLAS outperforms many studied protocols in terms of execution time for one and multiple safety-related messages.

CHAPTER 6: CL-BASED PROTOCOL WITH HYBRID AUTHENTICATION

In this Chapter, we introduce our Hybrid-based scheme [111]. We begin by giving an introduction of the context of application of our scheme. In the section 2, we explain the challenge faced in VANET and how our scheme can counter the present vulnerabilities. In the section 3, we outline the security objectives of our Hybrid-based scheme. The section 4 explains how nodes can use those protocols in the network, while the section 5 describes the system model. We provide an overview of our scheme in the sections 6. Additionally, we give a description of the used algorithms in the section 7. Then, we give the cryptographic computations in the sections 8. A simulation was performed, as described in the section 9. Then, we conclude the chapter in the section 10.

6.1 Introduction

Context related to Hybrid scheme. To secure V2X communications, many CPPA protocols utilize both symmetric and asymmetric encryption during the authentication process. However, several protocols have security limitations concerning VANET requirements. In many protocols based on symmetric cryptography, participants must share the same *sk*s, potentially compromising network security if one participant's key is compromised. On the other hand, many protocols based on asymmetric cryptography are time-consuming during the authentication process and do not address DOS attacks. In this work, we propose a CL-based protocol that doesn't require certificates and avoids the key escrow problem. Furthermore, it employs ECC while avoiding pairing and MTH functions. We refer to our protocol as the Hybrid Cryptography-Based Scheme with Conditional Privacy-Preserving Authentication (HCBS-CPPA) because it utilizes both symmetric and asymmetric cryptography during the authentication process.

6.2 Problem Statement regarding our hybrid scheme

A new CL-based protocol is developed and overcomes the key escrow problem. Concerning cryptographic computations, many research efforts use either symmetric encryption, asymmetric encryption, or hybrid encryption during the authentication process. The advantage of a protocol based on symmetric cryptography is its fast execution time and the absence of certificate requirements. However, it has security limitations due to the use of the same key by all network participants. Therefore, if one VC is compromised, it would potentially compromise the security of the entire network. On the other hand, a protocol based on asymmetric cryptography provides a more secure solution at the cost of higher execution time. Another drawback of many identity-based CL-based protocols is their vulnerability to memory-based DOS attacks. These cyberattacks occur when an adversary floods a receiver with invalid signatures, leading to significant memory and resource consumption during processing. In this work, we propose a new CL-based protocol that combines symmetric and asymmetric cryptography. On the one hand, symmetric encryption is used for fast message signature verification. On the other hand, asymmetric encryption is employed to authenticate the sender and ensure non-repudiation.

Additionally, our protocol offers a solution to resist memory-based DOS attacks that could compromise safety-related message availability in the network [112]. This attack can occur when a unauthorized VC inundates a receiver with invalid safety-related messages, and the adversary does not possess corresponding keys, which is also known as a pollution attack [19], [113].

6.3 Objectives of HCBS-CPPA

Our protocol shows efficiency regarding signing and verification delays and resistance to DOS-based memory attack. Our contribution. The main contributions in our research work can be defined as follows:

1- Our scheme HCBS-CPPA uses a CL-based protocol which does not need to deploy a certificate and don't suffer from the escrow problem.

2- HCBS-CPPA combines both strengths of a symmetric and asymmetric cryptography. On the one hand, HCBS-CPPA uses an asymmetric cryptography to provide non-repudiation. On the other hand, HCBS-CPPA uses a symmetric cryptography to provide fast computation during the authentication process. During the authentication process, HCBS-CPPA is based on Elliptic Curve Diffie-Hellman Key Exchange (ECDHKE) and Hash-based Message Authentication Code (HMAC).

3- HCBS-CPPA fulfills all known VANET requirements: Authentication, Integrity, Identity Privacy Preserving, Non-repudiation, unlinkability, non-repudiation. Additionally, it prevents the escrow problem, resilient to memory-based DOS attack and other known cyberattacks in VANET. Plus, it prevents a single point failure related to the key storage.

4- Our simulation shows that HCBS-CPPA requires the less computation time during V2V and V2I communications when compared with the existing CL-based asymmetric-based cryptography schemes.

6.4 Introduction of the new protocol HCBS-CPPA

Our protocol combines the advantage of asymmetric encryption that satisfies non-repudiation and the advantage of symmetric encryption that allows for lightweight authentication. Additionally, we demonstrate that our protocol is resilient to memory-based DoS attacks, which occur when an adversary floods a receiver's memory with invalid safety-related messages. A security proof shows that HCBS-CPPA is EUF-CMA. Regarding the performance of our protocol, it turns out that our protocol achieves better performance compared to several CL-based protocols. Furthermore, it requires less execution time during the signing and verification processes as well as less communication cost compared to many protocols.

6.5 VANET model of our HCBS-CPPA

In this case, VANET architecture mainly consists of four components:

CDA. Central Distribution Authority is responsible for issuing pseudonyms to VCs and RSUs after verifying their real identities. In case a VC or a RSU misbehaves and starts sending false messages, an unauthorized participant can be detected via its pseudonym or *pk*.

KDA. Key Distribution Authority is responsible for providing *psk* to VCs and RSUs as part of the concept of a CL-based scheme. In our scheme, KDA can calculate the *pk* of a participant without having knowledge of its key *x*. In addition, KDA binds each *pk* to a unique pseudonym. When a VC or a RSU receives a *pk* from KDA, it can compute the same *pk* using its key *x*.

VC. By using an OBU, a VC can broadcast messages to other VCs and RSUs in its surrounding via DSRC protocol.

RSU. Can collect information and broadcasts messages to nearby VCs.

Table 15. Notations used in HCBS-CPPA

Symbol	Description
p *and* q	primes
q	order q of the group G
G	group of order q
P	generator point of the elliptic curve $E(F_p)$
n	prime order of P
(α , P_{pub})	KDA's Key pair
(β, T_{pub})	CDA's Key pair
psk_i	partial private key delivered by the authority
(sk_1 , pk_1)	key pair of the sender
(sk_2 , pk_2)	key pair of the verifier
θ_1	cryptographic value created by the sender
θ_2	cryptographic value created by the verifier
ID_{real}	real identity of a node
$PID_{i,1}$	pseudonym generated by a node i
$PID_{i,2}$	pseudonym generated by CDA for a node i
ID_i	identity of a VC or RSU i generated by itself
PID_i	pseudonym generated by CDA
h	one way hash function defined as: $h: \{0, 1\}^* \longrightarrow Z_q^*$
T_i	Lifetime of a pseudonym
t_0	reception time of a massage
t_1	current timestamp used to sign a safety-related message
Δt_1	minimum time needed for a sender to perform an ECC multiplication computation
Δt_2	maximum delay allowed to receive a safety-related message from a sender

6.6 Overview of HCBS-CPPA

Our scheme HCBS-CPPA is a CL-based scheme since a VC or a RSU computes its *sk* by using the *psk* provided by the KDA. Then, it combines it to its key *x*. Our scheme follows the phases below during the authentication process:
Phase 1. A RSU or a VC sends a request to the CDA using its ID_{real} to obtain a pseudonym and a *psk*.
Phase 2. The CDA generates a pseudonym based on the real identity of a RSU or a VC. Then, the CDA transmits it to the KDA. Note that only CDA can trace the real identity from a pseudonym.
Phase 3. The KDA computes a *psk* based on the received pseudonym from CDA. In addition, KDA computes the corresponding *pk* to the pseudonym. Then, it sends the result back to the CDA for surveillance purposes.
Phase 4. KDA provides the RSU or a VC with a pseudonym and a *psk*.
Phase 5. The transmitter and verifier proceed both to ECDH Key exchange to create a symmetric key. After that, the sender signs the safety-related message using the shared key, and the receiver can in its turn proceed to the verification process of the signature using the same shared key. Note that our scheme allows only legitimate nodes to proceed to ECDHKE.

6.7 ECDHKE and HMAC

Elliptic Curve Diffie-Hellman Key Exchange (ECDHKE). is a cryptographic protocol based on elliptic curves, that authenticate two entities to anonymously create a shared key over a public channel. The shared key can be used for symmetric encryption or authentication. ECDHKE is based on the difficulty of Diffie-Hellman problem which is a variant to the discrete logarithm problem[114]. For instance, Alice and Bob both agree on an elliptic curve $E(F_p)$, a point P of order n on E. Then, Alice chooses $a \in Z_q^*$ and computes $P_a=a \cdot P$ while Bob chooses $b \in Z_q^*$ and computes $P_b=b \cdot P$. Alice sends P_a to Bob while Bob sends P_b to Alice. Alice and Bob respectively can compute the shared key as follows*:* $k = a \cdot P_b$ and $k = b \cdot P_a$.

Hash-based Message Authentication Code (HMAC). is a type of safety-related message authentication code that uses a hash function and a secret cryptographic key. It is used to verify both integrity and authentication of a safety-related message. HMAC can provide authentication using a symmetric key. For instance, SHA-256 operates on 512-bit blocks. The length of the output of HMAC is the same as that of the underlying hash function (the outcome of HMAC-*SHA-256* is 256 bits when using SHA-256). Additionally, the HMAC output can be truncated if needed according to [115], [116].

6.8 Cryptographic computations of HCBS-CPPA

In this section, we show that HCBS-CPPA can be broken down into two stages:

Stage1. includes the distribution of the pseudonym and *psk* to VCs and RSUs by the KDA and CDA authorities as shown in the Fig 37.

Stage2. includes the authentication process of signatures between VCs and RSUs as shown in the Fig 38.

6.8.1 Distribution of a Pseudonym and Partial Private Key

In this section, the steps related to the distribution of the pseudonym and *psk* are detailed. In addition, the Fig 37. gives a summary of the cryptographic computations.

Initialization

Step 1: The KDA and CDA publish public parameters that consist of G of order q, $E(F_p)$ and P of G, a hash function $h: \{0, 1\}^* \longrightarrow Z_q^*$ and an *HMAC* function. Then, KDA and CDA respectively generate their *pk*s P_{pub} and T_{pub} by using respectively their Master secret keys α and β as follows: $P_{pub} = \alpha \cdot P$ and $T_{pub} = \beta \cdot P$. After that, KDA and CDA publish *params* to VCs and RSUs.

Creation of a pseudonym

Step 2. When a VC or a RSU wants to make a request, it chooses its key $x_i \in Z_q^*$, then it computes X_i. After that, it generates an ID_i. Then, the VC or RSU sends (ID_i, X_i) to CDA.

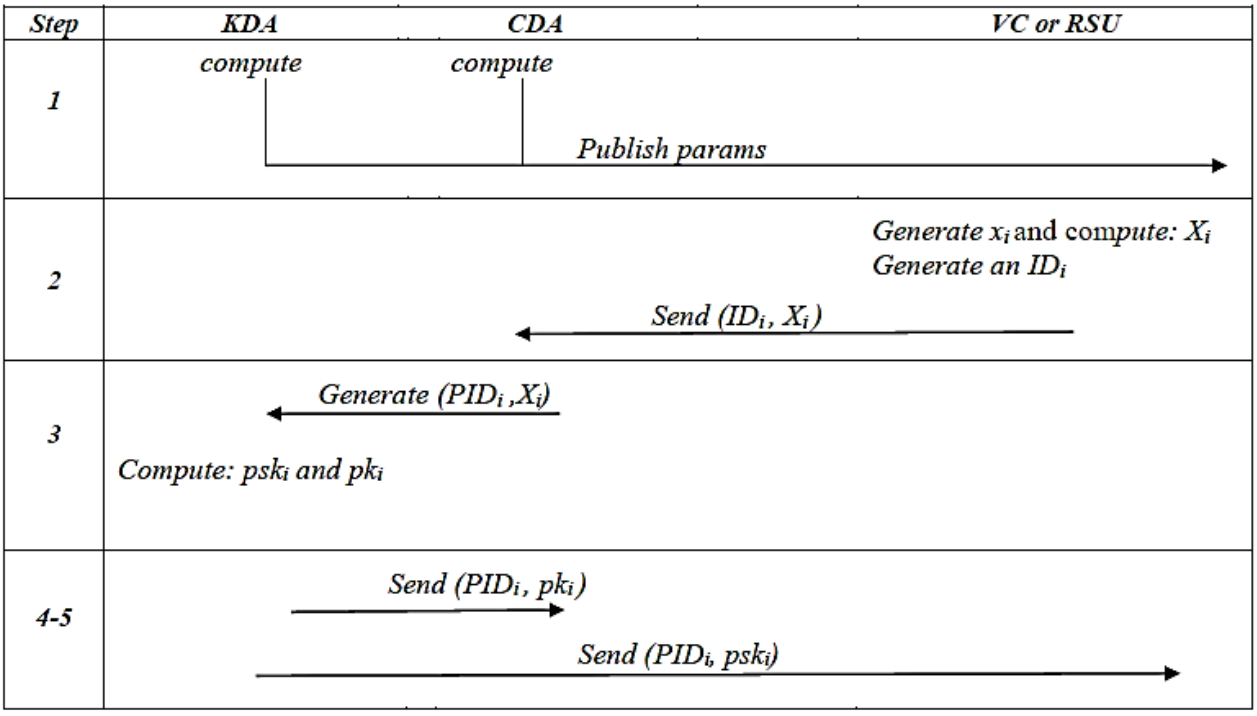


Fig 40. Distribution of a pseudonym and *psk* to a node from the KDA and CDA authorities

Creation of the partial private and *pk*s

Step 3. When CDA receives $ID_i = (ID_{Real}, PID_{i,1})$, it checks first if the real identity ID_{Real} is valid. If it is valid, CDA generates a $PID_{i,2}$. After that, CDA transmitts (PID_i, X_i) to KDA.

Step 4-5. When KDA receives (ID_i, X_i), it calculates *psk*. Then, KDA sends (PID_i, pk_i) to CDA for surveillance of the network. After that, KDA sends (PID_i, pk_i) to the VC or RSU.

6.8.2 Authentication process between VCs and RSUs

Generation of the private key

In this section, the steps related to the generation of *sk*, message signing, and message verification are detailed. In addition, the Figure 37. gives a summary of the used cryptographic computations:

Step 6. Prior to the authentication process, the sender and verifier respectively generate their *sk*s and cryptographic values. Now, the sender and verifier are both ready to enter in the authentication process.

Step 7. During the broadcast of a massage, a transmitter VC or a RSU sends θ_1 to a receiver VC or RSU. The idea here is to send θ_1 instead of pk_1 to reduce the communication cost. When the receiver receives θ_1, it stores the value (θ_1, t_0) until receiving the message, where t_0 is the reception time.

Step 8. When the sender receives θ_2 from the receiver, it computes the shared key. The idea behind sending θ_2 instead of pk_2 is also to reduce the communication cost and allows the sender to compute the shared key k.

Signature and Verification Process

Step 8. Using the shared key, the sender computes the signature σ as follows: σ. During message verification, the verifier verifies the safety-related message using the shared key k and proceeds as following: It performs a lookup function to find (θ_1, t_0) and checks if $\Delta t_1 < t_1 - t_0 < \Delta t_2$ is true, where Δt_1: is the minimum time needed to the sender to perform an ECC multiplication computation, and Δt_2 is the maximum delay to receive the message from the sender. If θ_1 is not found or $t_1 - t_0$ is not in the time interval, then the message is dropped. In case the receiver is exposed to a memory-based DOS attack, where it is flooded with invalid messages $(PID_1, m_1, \theta_1, t_1, \sigma)$, it proceeds as follows before any verification of messages:

1. The verifier discards the tuple (θ_1, t_0) after Δt_2, if the corresponding message is not received. In this case, the receiver assumes that the message was lost, or an adversary is trying to inject θ_1.
2. The reciever applies a lookup function to check if (θ_1, t_0) exists. If the tuple is not found, the message is dropped.
3. When the reciever finds (θ_1, t_0), it checks if $\Delta t_1 < t_1 - t_0 < \Delta t_2$ is true. Otherwise, the message is dropped. The idea behind setting Δt_1 is to force the sender to expend some computation computations to prevent the adversary from sending invalid messages in a very short time. Plus, the idea behind setting Δt_2 is to prevent (θ_1, t_0) to be stored in the memory for a long period of time.

Step	Vehicle or Roadside unit (Sender) Signing process	Vehicle or Roadside unit (Verifier) Verification process
6	Compute $sk_1 = psk_1 \cdot x_1$ Compute $\theta_1 = x_1 \cdot h\ (PID_1, X_1)\ mod\ n$ Where: $pk_1 = \theta_1 \cdot P_{pub}$	Compute $sk_2 = psk_2 \cdot x_2$ Compute $\theta_2 = x_2 \cdot h\ (PID_2, X_2)\ mod\ n$ Where: $pk_2 = \theta_2 \cdot P_{pub}$
7	Send θ_1 →	Store (θ_1, t_0) with t_0 is the reception time ← Send θ_2
8	Compute shared key: $k = sk_1 \cdot \theta_2 \cdot P$ Compute σ = MAC signature using HMAC-SHA-256 and k over $(PID_1, m_1, \theta_1, t_1)$ Send $(PID_1, m_1, \theta_1, t_1, \sigma)$ →	Lookup (θ_1, t_0). If the tuple exists with $\Delta t_1 < t_1 - t_0 < \Delta t_2$, proceed to the verification of the message / Otherwise Ignore Compute shared key: $k = sk_2 \cdot \theta_1 \cdot P$ Verify σ = MAC signature using HMAC-SHA-256 and k over $(PID_1, m_1, \theta_1, t_1)$

Fig 41. Authentication process between VCs and road infrastructures

6.9 Simulation and performance evaluation

6.9.1 Computation Time

Simulation. The simulation related to the execution time of the cryptographic computations was performed using MIRACL library. The execution time was considered from [115], as mentioned in the table 16. In addition, we neglect the execution time relation to the computations in Z_q^* and hash function operation.

Table 16. Execution time of cryptographic computations

Symbol	Description of the cryptographic operation
δ_{bp}	Time required to perform a BP operation
δ_{bp-m}	Scalar multiplication over BP
δ_{bp-a}	Addition operation over BP
$\delta_{m\text{-}ecc}$	Multiplication operation over ECC

$\delta_{a\text{-}ecc}$	Addition operation over ECC
δ_{MTP}	MTH function operation of an integer to a point in the group
δ_h	One-way hash function operation using SHA-256 algorithm
δ_{MAC}	Message Authentication Code operation using HMAC- SHA-256

By using the execution time related to the cryptographic computations in the Table 17, our scheme is compared with the existing CL-based schemes [35], [38], [39], [40], [41].

Table 17. Comparison between HCBS-CPPA and existing CL-based schemes

Protocole	Signing time (ms)	Verification time (ms)	Authentication time (ms)
[38]	$2\ \delta_{bp-m}$	$3\delta_{bp} + \delta_{bp-m} + \delta_{MTP}$	22.166
[39]	$2\delta_{bp-m} + T_{MTP}$	$3\delta_{bp} + \delta_{bp-m} + 2\delta_{MTP}$	30.978
[40]	$4\delta_{bp-m}$	$3\delta_{bp} + 3\delta_{bp-m}$	24.596
[16]	$3\delta_{bp-m}$	$3\delta_{bp} + \delta_{MTP} + 2\delta_{bp-m}$	25.584
[35]	$3\delta_{m-ecc}$	$4\delta_{m-ecc}$	3.094
HCBS-CPPA	$\delta_{m-ecc} + \delta_{MAC}$	$\delta_{m-ecc} + \delta_{MAC}$	0.9134

The figure 42 depicts the results from the table 17.

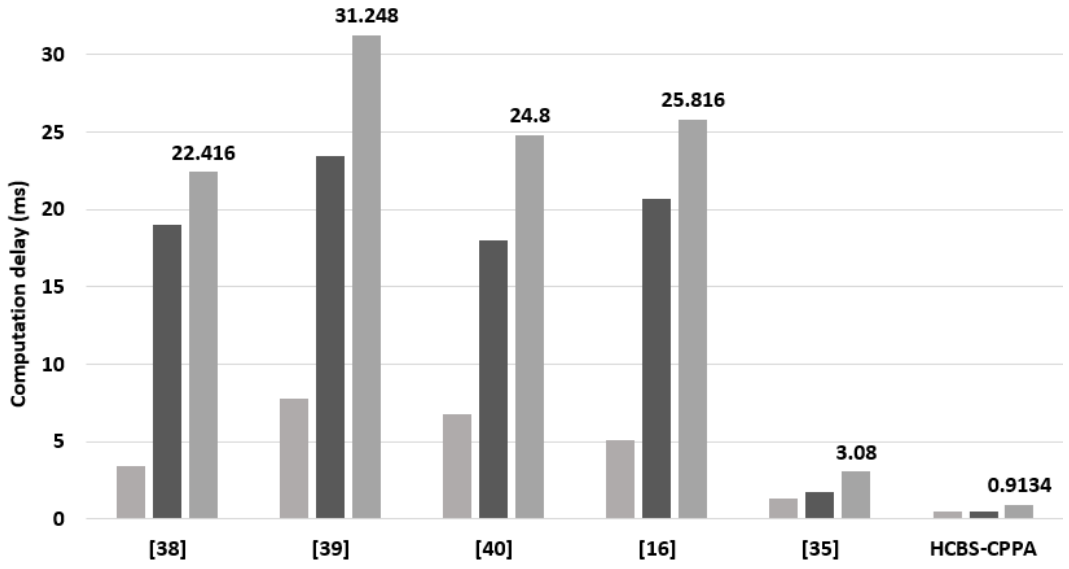


Fig 42. Authentication time including signing and verification process of a message

6.9.2 Communication cost

Regarding the communication overhead, we compare our scheme HCBS-CPPA with the existing CL-based schemes. The considered size for a safety-related message is 67 bytes. In BP, the elements in G_1 have the size of 128 bytes. In elliptic curve group, the elements in G have a size of 40 bytes. Additionally, we assume that the output of a hash function has 20 bytes, the elements in Z_q^* have 20 bytes and a timestamp has 4 bytes [16]. Regarding the HMAC output considered in our scheme, we truncate the HMAC output and use the truncated value for authentication as mentioned in the HMAC SHA standard [115], [116]. In this case, the length of the output value of the HMAC-SHA-256 algorithm is 32 bytes and can be truncated to 12 bytes. Thus, the signature output in our scheme can only keep the least significant 12 bytes. Regarding communication overhead, the Table 18 compares HCBS-CPPA and the existing CL-based schemes when considering the same sizes of the different parameters as mentioned in [9].

Table 18. Message communication cost of our scheme and existing CL-based schemes

Scheme	Message format	Communication cost (bytes)
[38]	$\|M_i\|+ \|PID_i\|+\| vpkID_i \|+ \|t_i \|+ \|\sigma_i\|$	607
[39]	$\|M_i\|+ \| PS1_j \|+ \| P_i \|+ \|U_i\|+ \| V_{ijk}\|$	579
[40]	$\|M_i\|+ \|PID_i\|+ \|vpk_i\| +\|t_i \| + \|\sigma_i\|$	255
[16]	$\|M_i\|+ \|PID_i\|+ \|t_i \|+\| vpkID_i \|+ \|QID_i\| + \|\sigma_i\|$	275
[35]	$\|M_i\|+ \|PID_i\|+\| t_i\|+ \|P_i\|+ \|D_i\|+ \|R_i\|+ \|\sigma_i\|$	275
HCBS-CPPA	$\| \theta_1\|+ \| \theta_2\|+\| \theta_1\|+ \|m_1\|+ \|PID_1\|+ \|t_1\|+\|\sigma\|$	207

The Fig 43. compares our scheme HCBS-CPPA to the CL-based schemes [35], [38], [39], [40], [41]. The results show that our scheme has the less overhead communication during V2V and V2I communications between VCs and RSUs.

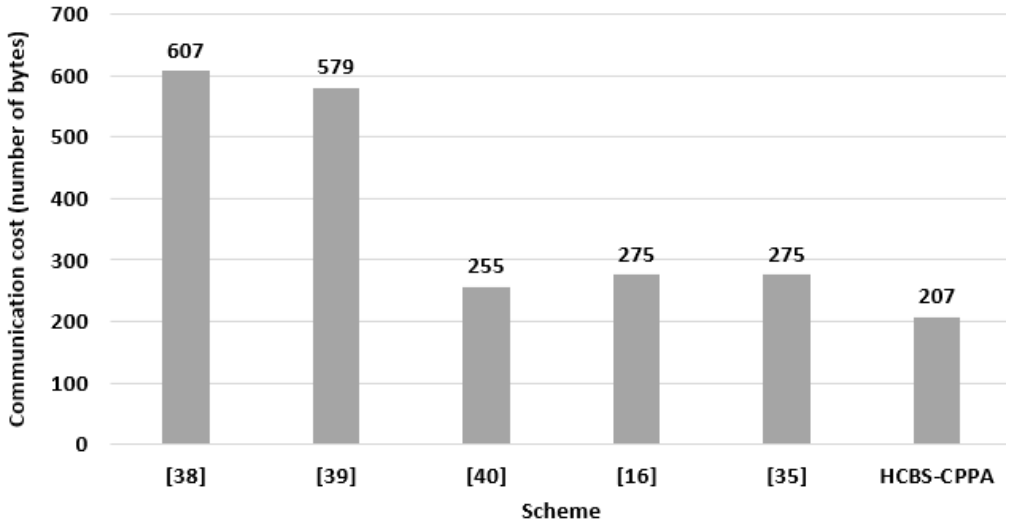


Fig 43. communication cost per scheme

6.10 Conclusion regarding HCBS-CPPA

A new CPPA is developed that satisfies all VANET security requirements. We call our scheme HCBS-CPPA since it is considered as a Hybrid Cryptography-Based Scheme with a CPPA. HCBS-CPPA is a CL-based scheme uses ECC and avoids BP and MTH functions that are known as complex computations. Plus, it uses both symmetric and asymmetric cryptography during the authentication process. Regarding cryptography approaches, HCBS-CPPA combines the strengths of both symmetric cryptography-based and asymmetric cryptography-based concepts. On the one hand, it uses asymmetric encryption to provide a non-repudiation. On the other hand, it uses a symmetric encryption to provide a lightweight authentication process. Regarding the security of the scheme, we show that HCBS-CPPA is EUF-CMA. Plus, we show that HCBS-CPPA is resilient against memory-based DOS attack, since the verifier deploys some security mechanisms related to a time interval to counter a "pollution attack".[19]. This latter attack happens when an adversary tries to flood a VC or a RSU with invalid messages during V2V and V2I communications. Finally, our simulation shows that our scheme HCBS-CPPA requires less execution time than many CL-based schemes and less overhead during the authentication process.

CHAPTER 7: GENERAL CONCLUSION AND FUTURE PROSPECTS

Today, PKI is considered a secure solution in VANET. However, its certificate management, generation, distribution, and revocation processes are perceived as complex [18]. Moreover, adding a certificate to each transmitted message increases the packet size and may potentially saturate the network bandwidth [17], especially as the number of VCs on the road increases each year, leading to an increase in the number of distributed certificates. In this research work, four CL-based protocols have been developed, and they can be described as follows:

1. *ECDSA*-based scheme:* This is a new asymmetric conditional anonymity-preserving authentication protocol based on the CL-based concept. It uses ECC and avoids mapping the hash function to a point and pairing. ECDSA*-based scheme achieves a TL 3 and meets all security requirements in VANET. Furthermore, it allows nodes to authenticate using the ECDSA* algorithm. A security proof demonstrates that this protocol is secure, given the difficulty of solving ECC problems. In IoT, ECDSA*-based scheme can be used for a single signature and a batch verification of signatures without disclosing the y-coordinate of point *R*. Thus, a RSU can assist VCs in verifying their messages in congested traffic areas. Our evaluation results show that ECDSA*-based scheme outperforms many protocols. Additionally, the ECDSA*-based scheme incurs lower communication costs compared to the studied CL-based protocols.
2. *Schnorr-based scheme:* This is a second new asymmetric CPPA based on the CL-based concept and ECC cryptography. It enables VCs and road infrastructures to authenticate using the Schnorr algorithm. To verify the same safety-related message from different transmitters, a RSU can carry out a MultiSig, aggregate the signatures and send a single signature to TCC. Moreover, a RSU can carry out a batch verification when receiving different safety-related messages from different senders and send one aggregated signature to TCC. Our protocol achieves VANET requirements and turns out resilient to the rogue attack. We also demonstrate that this protocol is EUF-CMA given the difficulty of ECCDHP and ECDLP. According to our simulations, the Schnorr-based scheme requires less time and incurs lower communication cost compared to the existing protocols. When a hazardous event is detected, our protocol can enhance the safety distance by stopping the VC by 4.5 meters in advance, when compared to the existing protocols. As a result, a driver can stop the VC earlier and prevent dangerous situations.
3. *CL-based Aggregate Signature (CLAS):* This is a third new asymmetric CL-based aggregate signature protocol based on pairing, with a TL 3, providing a strong non-repudiation and resistance to cyberattacks involving the replacement of *pk*s. Additionally, our protocol offers an authentication process during V2V and V2I communications. When a RSU receives multiple messages from VCs, it performs a batch verification to ensure that all signatures are valid, reducing computation time compared to verifying messages one by one. Afterward, the RSU aggregates the signatures and sends the aggregate signature to other RSUs and TCC. Regarding the performance evaluation of

our CLAS protocol, it turns out that our CLAS outperforms many studied protocols concerning the execution time for the signing and verification process of single and multiple messages. A security analysis is conducted, demonstrating that our CLAS protocol is EUF-CMA given the difficulty of the CDH problem.

4. *HCBS-CPPA:* This is a fourth new CPPA protocol based on the CL-based concept and ECC cryptography with hybrid encryption. This protocol also satisfies all VANET security requirements. On the one hand, it uses asymmetric encryption to provide non-repudiation. On the other hand, it employs symmetric encryption to offer a lightweight authentication process. We demonstrate that HCBS-CPPA is EUF-CMA and resilient against memory-based DoS attacks, as a verifier deploys certain security mechanisms related to a time interval to counter a potential "pollution attack".[19]. This latter attack occurs when an adversary attempts to flood a VC or road infrastructure with invalid messages. Finally, our simulation shows that our HCBS-CPPA protocol requires less execution time than many CL-based protocols and incurs lower communication costs during the authentication process.

Note: In conclusion, the four new CL-based protocols offer different authentication features that can be deployed in a real VANET. During the evaluation of our protocols, comparisons in terms of security and performance were made using 80 bits. However, in, it is advisable to utilize these protocols with a SL of 128 bits, as referenced in both [11], [12]. Furthermore, they hold potential for application in other IoT scenarios in future works.

Limitation of our four protocols: Our four certificateless protocols rely on a single authority that provides vehicles and roadside units with cryptographic parameters for authentication. the limitation of relying on a single authority is considered as a significant concern. The exposure of the master key belonging to the single authority could potentially compromise the entire network, leading to a complete breakdown of the system's security.

To address this limitation and enhance the resilience of VANET protocols, it is necessary to consider implementing a multi-authority approach. By having multiple authorities that can act as backups in case one authority fails or is compromised, the network's security can be improved. These authorities should be able to synchronize and exchange data with each other in real-time to ensure seamless operation and maintain the integrity of the system. The multi-authority approach offers several advantages:

- *Redundancy:* Having multiple authorities reduces the risk of a single point of failure. If one authority is compromised or experiences a breakdown, the other authorities can continue to provide the necessary cryptographic parameters, ensuring the network's continued operation.
- *Improved security:* By distributing the trust and responsibility among multiple authorities, the impact of a single authority's compromise is minimized. An attacker would need to compromise multiple authorities simultaneously to gain control over the entire network, making it more challenging to execute successful attacks.

- *Scalability:* As VANETs grow in size and complexity, a multi-authority approach allows for better scalability. The authorities can share the workload and manage larger numbers of vehicles and roadside units without overburdening a single entity.
- *Flexibility:* With multiple authorities, the network can adapt more easily to changes, such as the addition or removal of authorities, without disrupting the entire system.

To implement a multi-authority approach in VANET protocols, several considerations should be taken into account:

- *Secure communication channels:* Establish secure communication channels between the authorities to ensure the confidentiality and integrity of the exchanged data.
- *Synchronization mechanisms:* Develop robust synchronization mechanisms that allow authorities to share and update cryptographic parameters in real-time, ensuring consistency across the network.
- *Authority management:* Implement procedures for managing authorities, including adding, removing, or replacing authorities as needed, while maintaining the overall security of the system.
- *Key management:* Establish secure key management protocols to handle the generation, distribution, and revocation of cryptographic keys among the authorities and network participants.

By addressing the limitation of relying on a single authority and adopting a multi-authority approach, VANET protocols can enhance their security, resilience, and scalability, making them better equipped to handle the challenges posed by the dynamic and distributed nature of vehicular networks.
As perspectives of a research work, our future work aims to address the limitations mentioned above. In addition, we plan to address other critical challenges in VANET security through innovative approaches to certificateless schemes. We propose to explore the following key areas:

Optimization of pairing computations:

- Develop faster pairing algorithms tailored for vehicular environments to reduce computational overhead.
- Investigate optimal pairing-friendly curves that balance security and efficiency for VANET applications.

Design of hybrid certificateless schemes:

- Create new schemes that combine classical and post-quantum primitives to facilitate a smooth transition as quantum threats emerge.
- Analyze trade-offs between security levels and performance in these hybrid approaches.

Development of quantum-resistant protocols:

- Adapt existing VANET protocols and standards to incorporate post-quantum certificateless schemes without major infrastructure overhauls.

BIBLIOGRAPHIE

[1] "Qureshi, K. N., & Abdullah, H. (2013). Topology based routing protocols for VANET and their comparison with MANET. Journal of Theoretical and Applied Information Technology, 58(3), 707-715.".

[2] "Saraswathi, P., RAJKUMAR, G., & Rao, P. M. (2022). Intelligent Transport System using IoT-V2X: Communication Technologies, Security Issues, Challenges and Countermeasures.".

[3] "Shrestha, R., Bajracharya, R., & Nam, S. Y. (2018). Challenges of future VANET and cloud-based approaches. Wireless Communications and Mobile Computing, 2018.".

[4] "Al-ani, R., Zhou, B., Shi, Q., & Sagheer, A. (2018, June). A survey on secure safety applications in VANET. In 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (pp. 1485-1490). IEEE.".

[5] "Hossain, E., Chow, G., Leung, V. C., McLeod, R. D., Mišić, J., Wong, V. W., & Yang, O. (2010). Vehicular telematics over heterogeneous wireless networks: A survey. Computer communications, 33(7), 775-793.".

[6] "Ahmed, W., Di, W., & Mukathe, D. (2022). Privacy-preserving blockchain-based authentication and trust management in VANETs. IET Networks.".

[7] "Manivannan, D., Moni, S. S., & Zeadally, S. (2020). Secure authentication and privacy-preserving techniques in Vehicular Ad-hoc NETworks (VANETs). Vehicular Communications, 25, 100247.".

[8] "Hou, L., Yao, N., Lu, Z., Zhan, F., & Liu, Z. (2021). Tracking based mix-zone location privacy evaluation in VANET. IEEE Transactions on Vehicular Technology, 70(10), 10957-10969.".

[9] "Imghoure, A., El-Yahyaoui, A., & Omary, F. (2022). ECDSA-based certificateless conditional privacy-preserving authentication scheme in Vehicular Ad Hoc Network. Vehicular Communications, 37, 100504.".

[10] "Mansour, M. B., Salama, C., Mohamed, H. K., & Hammad, S. A. (2018). VANET security and privacy-an overview. International Journal of Network Security & Its Applications (IJNSA) Vol, 10.".

[11] "IEEE 1609.2: IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages Amendment 2: PDU Functional Types and Encryption Key Management. (2016)".

[12] "ETSI TS 103 097: Intelligent Transport Systems (ITS); Security; Security header and certificate formats. V1.4.1 (2020-10)".

[13] "Girault, M. (1991, April). Self-certified public keys. In Workshop on the Theory and Application of Cryptographic Techniques (pp. 490-497). Springer, Berlin, Heidelberg.".

[14] "Zhang, C., Lu, R., Lin, X., Ho, P. H., & Shen, X. (2008, April). An efficient identity-based batch verification scheme for vehicular sensor networks. In IEEE INFOCOM 2008-The 27th Conference on Computer Communications (pp. 246-250). IEEE.".

[15] "Studer, A., Bai, F., Bellur, B., & Perrig, A. (2009). Flexible, extensible, and efficient VANET authentication. Journal of Communications and Networks, 11(6), 574-588.".

[16] "He, D., Zeadally, S., Xu, B., & Huang, X. (2015). An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. IEEE Transactions on Information Forensics and Security, 10(12), 2681-2691.".

[17] "Sheikh, M. S., Liang, J., & Wang, W. (2019). A survey of security services, attacks, and applications for vehicular ad hoc networks (VANETs). Sensors, 19(16), 3589.".

[18] "Singh, P., Basit, A., Kumar, N. C., & Venkaiah, V. C. (2019). Towards a Hybrid Public Key Infrastructure (PKI): A Review. Cryptology ePrint Archive.".

[19] “Ruan, N., & Hori, Y. (2012, July). DoS attack-tolerant TESLA-based broadcast authentication protocol in Internet of Things. In 2012 International Conference on Selected Topics in Mobile and Wireless Networking (pp. 60-65). IEEE.”.
[20] “Antipa, A., Brown, D., Gallant, R., Lambert, R., Struik, R., & Vanstone, S. (2005, August). Accelerated verification of ECDSA signatures. In International Workshop on Selected Areas in Cryptography (pp. 307-318). Springer, Berlin, Heidelberg.”.
[21] “Yeh, K. H., Su, C., Choo, K. K. R., & Chiu, W. (2017). A novel certificateless signature scheme for smart objects in the Internet-of-Things. Sensors, 17(5), 1001.”.
[22] “Ming, Y., & Shen, X. (2018). PCPA: A practical certificateless conditional privacy preserving authentication scheme for vehicular ad hoc networks. Sensors, 18(5), 1573.”.
[23] “Okano, H., Emura, K., Ishibashi, T., Ohigashi, T., & Suzuki, T. (2020). Implementation of a strongly robust identity-based encryption scheme over type-3 pairings. International Journal of Networking and Computing, 10(2), 174-188.”.
[24] “Xu, R., Wang, X., & Morozov, K. (2021). Group authentication for cloud-to-things computing: Review and improvement. Computer Networks, 198, 108374.”.
[25] “Miyaji, A., Nakabayashi, M., & Takano, S. (2001). New explicit conditions of elliptic curve traces for FR-reduction. IEICE transactions on fundamentals of electronics, communications and computer sciences, 84(5), 1234-1243.”.
[26] “Barreto, P. S., & Naehrig, M. (2006). Pairing-friendly elliptic curves of prime order. In Selected Areas in Cryptography: 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers 12 (pp. 319-331). Springer Berlin Heidelberg.”.
[27] “Barreto, P. S., Lynn, B., & Scott, M. (2003). Constructing elliptic curves with prescribed embedding degrees. In Security in Communication Networks: Third International Conference, SCN 2002 Amalfi, Italy, September 11–13, 2002 Revised Papers 3 (pp. 257-267). Springer Berlin Heidelberg.”.
[28] “Boneh, D., & Franklin, M. (2003). Identity-based encryption from the Weil pairing. SIAM journal on computing, 32(3), 586-615.”.
[29] “A. Miyaji, M. Nakabayashi, S. Takano, New explicit conditions of elliptic curve traces for FR-reduction, IEICE Trans. Fundam. E84-A (5) (2001) 1234–838 1243.”.
[30] “Hess, F., Smart, N. P., & Vercauteren, F. (2006). The eta pairing revisited. IEEE transactions on information theory, 52(10), 4595-4602.”.
[31] “Lee, E., Lee, H. S., & Park, C. M. (2009). Efficient and generalized pairing computation on abelian varieties. IEEE Transactions on Information Theory, 55(4), 1793-1803.”.
[32] “Lo, N. W., & Tsai, J. L. (2015). An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings. IEEE Transactions on Intelligent Transportation Systems, 17(5), 1319-1328.”.
[33] “Wu, L., Fan, J., Xie, Y., Wang, J., & Liu, Q. (2017). Efficient location-based conditional privacy-preserving authentication scheme for vehicle ad hoc networks. International Journal of Distributed Sensor Networks, 13(3), 1550147717700899.”.
[34] “Li, K., Au, M. H., Ho, W. H., & Wang, Y. L. (2019, October). An efficient conditional privacy-preserving authentication scheme for vehicular ad hoc networks using online/offline certificateless aggregate signature. In International Conference on Provable Security (pp. 59-76). Springer, Cham.”.
[35] “Ming, Y., & Cheng, H. (2019). Efficient certificateless conditional privacy-preserving authentication scheme in VANETs. Mobile Information Systems, 2019.”.
[36] “Li, J., & Zhang, Y. (2020). Cryptanalysis and improvement of batch verification certificateless signature scheme for VANETs. Wireless Personal Communications, 111(2), 1255-1269.”.

[37] “Li, J., Yuan, H., & Zhang, Y. (2018). Cryptanalysis and improvement for certificateless aggregate signature. Fundamenta Informaticae, 157(1-2), 111-123.”.
[38] “Horng, S. J., Tzeng, S. F., Huang, P. H., Wang, X., Li, T., & Khan, M. K. (2015). An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks. Information Sciences, 317, 48-66.”.
[39] “Li, J., Yuan, H., & Zhang, Y. (2016). Cryptanalysis and improvement of certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks. Cryptology ePrint Archive.”.
[40] “Batra, S., & Malhi, A. K. (2015). An efficient certificateless aggregate signature scheme for vehicular ad-hoc networks. Discrete Mathematics & Theoretical Computer Science, 17.”.
[41] “Zhong, H., Han, S., Cui, J., Zhang, J., & Xu, Y. (2019). Privacy-preserving authentication scheme with full aggregation in VANET. Information Sciences, 476, 211-221.”.
[42] “Al-Shareeda, M. A., Anbar, M., Manickam, S., & Yassin, A. A. (2020). VPPCS: VANET-based privacy-preserving communication scheme. IEEE Access, 8, 150914-150928.”.
[43] “IEEE Std. 1609.2, IEEE Trial-Use Standard for Wireless Access in Vehicular Environments – Security Services for Applications and Management Messages, 2006.”.
[44] “Khan, T., Ahmad, N., Cao, Y., Jalal, S. A., Asif, M., & Cruichshank, H. (2017). Certificate revocation in vehicular ad hoc networks techniques and protocols: a survey. Science China Information Sciences, 60(10), 1-18.”.
[45] “R. Lu, X. Lin, H. Zhu, P.H. Ho, X. Shen, ECPP: efficient conditional privacy preservation protocol for secure vehicular communications, in: Proceedings of the IEEE INFOCOM, 2008, pp. 1229–1237.”.
[46] *[15] Yang, G., & Tan, C. H. (2011). Certificateless cryptography with KGC trust level 3. Theoretical computer science, 412(39), 5446-5457.*
[47] “M. Raya, P. Papadimitratos, J.P. Hubaux, Securing vehicular communications, IEEE Wireless Commun. 13 (5) (2006) 8–15.”.
[48] “M. Raya, J.P. Hubaux, Securing vehicular ad hoc networks, J. Comput. Security 15 (1) (2007) 39–68.”.
[49] “Shamir, A. (1984, August). Identity-based cryptosystems and signature schemes. In Workshop on the theory and application of cryptographic techniques (pp. 47-53). Springer, Berlin, Heidelberg.”.
[50] “Zhao, Z., Susilo, W., Guo, F., Lai, J., & Wang, B. (2023). Full black-box retrievable and accountable identity-based encryption. Computer Standards & Interfaces, 103741.”.
[51] “Al-Riyami, S. S., & Paterson, K. G. (2003, November). Certificateless public key cryptography. In International conference on the theory and application of cryptology and information security (pp. 452-473). Springer, Berlin, Heidelberg.”.
[52] “Yang, G., & Tan, C. H. (2011). Certificateless cryptography with KGC trust level 3. Theoretical computer science, 412(39), 5446-5457.”.
[53] “Hassouna, M. A., & Hashim, M. (2014). A Strong and Efficient Certificateless Digital Signature Scheme. Cryptology ePrint Archive.”.
[54] “Liu, J. K., Au, M. H., & Susilo, W. (2007, March). Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model. In Proceedings of the 2nd ACM symposium on Information, computer and communications security (pp. 273-283).”.
[55] “Petersenl, H., & Horster, P. (1997). Self-certified keys—concepts and applications. In Communications and Multimedia Security (pp. 102-116). Springer, Boston, MA.”.
[56] “Saeednia, S. (1997, July). Identity-based and self-certified key-exchange protocols. In Australasian conference on information security and privacy (pp. 303-313). Springer, Berlin, Heidelberg.”.

[57] “Saeednia, S. (2003). A note on Girault’s self-certified model. Information Processing Letters, 86(6), 323-327.”.
[58] “Zhang, J., Zhen, W., & Xu, M. (2013, December). An efficient privacy-preserving authentication protocol in VANETs. In 2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks (pp. 272-277). IEEE.”.
[59] “Oulhaci, T., Omar, M., Harzine, F., & Harfi, I. (2017). Secure and distributed certification system architecture for safety message authentication in VANET. Telecommunication Systems, 64(4), 679-694.”.
[60] “Hubaux, J. P., Capkun, S., & Luo, J. (2004). The security and privacy of smart vehicles. IEEE Security & Privacy, 2(3), 49-55.”.
[61] “Lu, R., Lin, X., Zhu, H., Ho, P. H., & Shen, X. (2008, April). ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. In IEEE INFOCOM 2008-The 27th Conference on Computer Communications (pp. 1229-1237). IEEE.”.
[62] “Jin, H., Debiao, H., & Jianhua, C. (2010, March). An identity based digital signature from ECDSA. In 2010 Second International Workshop on Education Technology and Computer Science (Vol. 1, pp. 627-630). IEEE.”.
[63] “Li, J., Yan, H., & Zhang, Y. (2018). Certificateless public integrity checking of group shared data on cloud storage. IEEE Transactions on Services Computing, 14(1), 71-81.”.
[64] “Cui, J., Zhang, J., Zhong, H., Shi, R., & Xu, Y. (2018). An efficient certificateless aggregate signature without pairings for vehicular ad hoc networks. Information Sciences, 451, 1-15.”.
[65] “Kamil, I. A., & Ogundoyin, S. O. (2019). An improved certificateless aggregate signature scheme without bilinear pairings for vehicular ad hoc networks. Journal of information security and applications, 44, 184-200.”.
[66] “Zhao, Y., Hou, Y., Wang, L., Kumari, S., Khan, M. K., & Xiong, H. (2020). An efficient certificateless aggregate signature scheme for the Internet of Vehicles. Transactions on Emerging Telecommunications Technologies, 31(5), e3708.”.
[67] “Kamil, I. A., & Ogundoyin, S. O. (2020). On the security of privacy-preserving authentication scheme with full aggregation in vehicular ad hoc network. Security and Privacy, 3(3), e104.”.
[68] “Cui, L., Gang, W., Xiaofeng, S., Feng, Z., & Liang, Z. (2020). An efficient certificateless aggregate signature scheme designed for VANET. Computers, Materials & Continua, 63(2), 725-742.”.
[69] “Mei, Q., Xiong, H., Chen, J., Yang, M., Kumari, S., & Khan, M. K. (2020). Efficient certificateless aggregate signature with conditional privacy preservation in IoV. IEEE Systems Journal, 15(1), 245-256.”.
[70] “Thumbur, G., Rao, G. S., Reddy, P. V., Gayathri, N. B., Reddy, D. K., & Padmavathamma, M. (2020). Efficient and secure certificateless aggregate signature-based authentication scheme for vehicular ad hoc networks. IEEE Internet of Things Journal, 8(3), 1908-1920.”.
[71] “Kumar, P., & Sharma, V. (2018). On the security of certificateless aggregate signature scheme in vehicular ad hoc networks. In Soft Computing: Theories and Applications (pp. 715-722). Springer, Singapore.”.
[72] “Cahyadi, E. F., Su, T. W., Yang, C. C., & Hwang, M. S. (2022). A certificateless aggregate signature scheme for security and privacy protection in VANET. International Journal of Distributed Sensor Networks, 18(5), 15501329221080658.”.
[73] “Qu, F., Wu, Z., Wang, F. Y., & Cho, W. (2015). A security and privacy review of VANETs. IEEE Transactions on Intelligent Transportation Systems, 16(6), 2985-2996.”.
[74] “Petit, J., Schaub, F., Feiri, M., & Kargl, F. (2014). Pseudonym schemes in vehicular networks: A survey. IEEE communications surveys & tutorials, 17(1), 228-255.”.
[75] “Zhang, Z., & Feng, D. (2006). Key replacement attack on a certificateless signature scheme. Cryptology ePrint Archive.”.

[76] “Hu, B. C., Wong, D. S., Zhang, Z., & Deng, X. (2007). Certificateless signature: a new security model and an improved generic construction. Designs, Codes and Cryptography, 42(2), 109-126.”.
[77] “Hu, B. C., Wong, D. S., Zhang, Z., & Deng, X. (2006, July). Key replacement attack against a generic construction of certificateless signature. In Australasian Conference on Information Security and Privacy (pp. 235-246). Springer, Berlin, Heidelberg.”.
[78] “Zhang, C., Lin, X., Lu, R., & Ho, P. H. (2008, May). RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks. In 2008 IEEE international conference on communications (pp. 1451-1457). IEEE.”.
[79] “Huang, X., Mu, Y., Susilo, W., Wong, D. S., & Wu, W. (2007, July). Certificateless signature revisited. In Australasian Conference on Information Security and Privacy (pp. 308-322). Springer, Berlin, Heidelberg.”.
[80] “Kittur, A. S., & Pais, A. R. (2019). A new batch verification scheme for ECDSA* signatures. Sādhanā, 44(7), 1-12.”.
[81] “Kietzmann, P., Schmidt, T. C., & Wählisch, M. (2021). A guideline on pseudorandom number generation (PRNG) in the IoT. ACM Computing Surveys (CSUR), 54(6), 1-38.”.
[82] “Bostancı, F. N., Olgun, A., Orosa, L., Yağlıkçı, A. G., Kim, J. S., Hassan, H., ... & Mutlu, O. (2022, April). DR-STRaNGe: End-to-End System Design for DRAM-based True Random Number Generators. In 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA) (pp. 1141-1155). IEEE.”.
[83] “Imghoure, A., Omary, F., & El-Yahyaoui, A. (2023). Schnorr-based conditional privacy-preserving authentication scheme with multisignature and batch verification in vanet. Internet of Things, 23, 100850.”.
[84] “Hathal, W. S., Cruickshank, H., Asuquo, P., Sun, Z., & Bao, S. (2019). Token-based lightweight authentication scheme for vehicle to infrastructure communications.”.
[85] “Schnorr, C. P. (1995). for SMART CARDs’. In Advances in Cryptology-CRYPTO’89: Proceedings (Vol. 435, p. 239). Springer.”.
[86] “Sripathi Venkata Naga, S. K., Yesuraj, R., Munuswamy, S., & Arputharaj, K. (2023). A comprehensive survey on certificate-less authentication schemes for vehicular ad hoc networks in intelligent transportation systems. Sensors, 23(5), 2682.”
[87] “Ristenpart, T., Yilek, S.: The power of proofs-of-possession: Securing multiparty signatures against rogue-key attacks. In: Naor, M. (ed.) Advances in Cryptology – EUROCRYPT 2007. Lecture Notes in Computer Science, vol. 4515, pp. 228–245. Springer, Heidelberg, Germany, Barcelona, Spain (May 20–24, 2007)”.
[88] “Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In: Desmedt, Y. (ed.) PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography. Lecture Notes in Computer Science, vol. 2567, pp. 31–46. Springer, Heidelberg, Germany, Miami, FL, USA (Jan 6–8, 2003)”.
[89] “Lu, S., Ostrovsky, R., Sahai, A., Shacham, H., Waters, B.: Sequential aggregate signatures and multisignatures without random oracles. In: Vaudenay, S. (ed.) Advances in Cryptology – EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 465–485. Springer, Heidelberg, Germany, St. Petersburg, Russia (May 28 – Jun 1, 2006)”.
[90] “Boneh, D., Drijvers, M., & Neven, G. (2018, December). Compact multi-signatures for smaller blockchains. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 435-464). Springer, Cham.”.
[91] “Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) Advances in Cryptology – EUROCRYPT 2003. Lecture Notes in Computer Science, vol. 2656, pp. 416–432. Springer, Heidelberg, Germany, Warsaw, Poland (May 4–8, 2003)”.

[92] “Mei, Q., Xiong, H., Chen, J., Yang, M., Kumari, S., & Khan, M. K. (2021). Efficient certificateless aggregate signature with conditional privacy preservation in IoV. IEEE Systems Journal, 15(1), 245-256.”.
[93] “C. C. Lee and Y. M. Lai, ‘Toward a secure batch verification with group testing for VANET,’ Wireless Netw., vol. 19, no. 6, pp. 1441–1449, 2013”.
[94] “Claus-Peter Schnorr. Efficient signature generation by smart cards. Journal of Cryptology, 4(3):161–174, January 1991.”.
[95] “Pieter Wuille. Schnorr signatures for secp256k1. Bitcoin Improvement Proposal, 2018. See https://github.com/sipa/bips/blob/bip-schnorr/bip-schnorr.mediawiki.”.
[96] “Dan Boneh, Manu Drijvers, and Gregory Neven. Compact multi-signatures for smaller blockchains. In Thomas Peyrin and Steven Galbraith, editors, ASIACRYPT 2018, Part II, volume 11273 of LNCS, pages 435–464. Springer, December 2018.”.
[97] “Gregory Maxwell, Andrew Poelstra, Yannick Seurin, and Pieter Wuille. Simple Schnorr multisignatures with applications to Bitcoin. Designs, Codes and Cryptography, 2019.”.
[98] “David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli M. Maurer, editor, EUROCRYPT’96, volume 1070 of LNCS, pages 387–398. Springer, May 1996.”.
[99] “David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. Journal of Cryptology, 13(3):361–396, June 2000.”.
[100] “Petit, J., & Mammeri, Z. (2013). Authentication and consensus overhead in vehicular ad hoc networks. Telecommunication systems, 52, 2699-2712.”.
[101] “D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairing, Journal of Cryptology 17 (2004) 297–319.”.
[102] “https://crypto.stanford.edu/pbc/notes/ep/curve.html”.
[103] “Joye, M., & Neven, G. (2009). Software implementation of pairings. Identity-Based Cryptography, 2, 188.”.
[104] “Boneh, D., Lynn, B., & Shacham, H. (2004). Short signatures from the Weil pairing. Journal of cryptology, 17(4), 297-319.”.
[105] “Eastlake 3rd, D., & Jones, P. (2001). US secure hash algorithm 1 (SHA1) (No. rfc3174).”.
[106] “Pointcheval, D., & Stern, J. (1996, May). Security proofs for signature schemes. In International conference on the theory and applications of cryptographic techniques (pp. 387-398). Springer, Berlin, Heidelberg.”.
[107] “Altaf, F., & Maity, S. (2021). PLHAS: Privacy-preserving localized hybrid authentication scheme for large scale vehicular ad hoc networks. Vehicular Communications, 30, 100347.”.
[108] “Shim, K. A. (2012). CPAS: an efficient conditional privacy-preserving authentication scheme for vehicular sensor networks. IEEE transactions on vehicular technology, 61(4), 1874-1883.”.
[109] “Scott, M., & Barreto, P. S. (2006). Generating more MNT elliptic curves. Designs, Codes and Cryptography, 38(2), 209-217.”.
[110] “https://members.loria.fr/AGuillevic/pairing-friendly-curves/”.
[111] “Imghoure, A., Omary, F., & El-Yahyaoui, A. (2024). Hybrid Cryptography-based Scheme with Conditional Privacy-Preserving Authentication and Memory-based DOS Resilience in V2X. Vehicular Communications, https://doi.org/10.1016/j.vehcom.2024.100810.”.
[112] “Sun, C., Liu, J., Xu, X., & Ma, J. (2017). A privacy-preserving mutual authentication resisting DoS attacks in VANETs. IEEE Access, 5, 24012-24022.”.
[113] “Perrig, A., Canetti, R., Song, D., & Tygar, J. D. (2001, February). Efficient and secure source authentication for multicast. In Network and Distributed System Security Symposium, NDSS (Vol. 1, No. 2001, pp. 35-46).”.
[114] “Diffie–Hellman Key Exchange - Practical Cryptography for Developers (nakov.com)”.

[115] “Hakeem, S. A. A., Abd El-Gawad, M. A., & Kim, H. (2019). A decentralized lightweight authentication and privacy protocol for vehicular networks. IEEE Access, 7, 119689-119705.”.
[116] “D. Eastlake, HMAC SHA TSIG Algorithm Identifiers, document RFC: 4635, 2006.”.

Résumé

Dans les réseaux véhiculaires (VANET), les nœuds sont composés de véhicules et d'unités d'infrastructure routière (RSU) communiquant via des liens V2V et V2I. Assurer simultanément l'authentification et l'anonymat des nœuds représente un défi de sécurité majeur. Les VANETs doivent satisfaire des exigences de sécurité fondamentales : authentification, intégrité, non-répudiation, protection de la vie privée, traçabilité et résistance aux cyberattaques. Le temps d'authentification des signatures est aussi critique étant donné la nature dynamique des échanges entre nœuds mobiles. Bien que de nombreux travaux aient proposé des protocoles d'authentification conditionnelle préservant l'anonymat, ces protocoles existants présentent souvent des limitations en termes de non-répudiation entre l'autorité de confiance et les nœuds en cas de litige. Notre recherche vise à concevoir de nouveaux protocoles CPPA offrant un niveau de sécurité supérieur, en garantissant une non-répudiation renforcée entre la TA et les nœuds, tout en satisfaisant les exigences de sécurité connues pour les VANET. Chaque nouveau protocole exploite différentes approches cryptographiques (symétriques, asymétriques, hybrides) basées sur les courbes elliptiques ou les couplages. Ils visent à optimiser les temps d'exécution et à réduire les coûts de communication lors de l'authentification, en introduisant des techniques avancées comme la vérification par lot, les multi-signatures et l'agrégation de signatures.

Mots clés : Réseau Ad-hoc de Véhicules, Authentification, Anonymat, signature, Agrégation.

Abstract

In VANET, vehicles and RSUs are considered nodes that exchange information using V2V and V2I communications. In this context, satisfying authentication and anonymity simultaneously is a significant security challenge. Additionally, VANET has to meet well-known security requirements, namely authentication, integrity, non-repudiation between nodes, privacy preservation, traceability, and resistance to several known attacks. During signature verification, execution time is also a challenging component, given that nodes exchange messages in a dynamic environment. Although many works are based on Conditional Privacy Preserving Authentication, existing protocols seem to have a limited level of security regarding non-repudiation between the TA and nodes. Our research aims to develop new protocols that provide a higher level of security compared to existing ones. Our new protocols utilize different cryptographic approaches, i.e., symmetric, asymmetric, and hybrid based on ECC and BP, while meeting the known security requirements in VANET. Furthermore, our protocols guarantee true non-repudiation between the TA and a node in a dispute and provide efficient execution time and communication costs during message transmission and signature verification. Each protocol also offers efficient authentication features such as batch verification of signatures, multi-signatures, and signature aggregation.

Keywords: Vehicular Ad-hoc Network, Authentication, Anonymity, Signature, Aggregation.