INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Departamento de Engenharia de Eletrónica e Telecomunicações e de Computadores

Security in Hybrid ITS Networks

Ricardo Filipe Quelhas Severino

Licenciado em Engenharia Informática, Redes e Telecomunicações

Dissertação para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

Orientadores : Professor Doutor José Manuel de Campos Lages Garcia Simão
Professor Doutor Nuno Miguel Soares Datia

Júri:

Presidente: Professor Doutor Nuno Miguel Machado Cruz

Vogais: Professor Doutor João Carlos Ferreira
Professor Doutor José Manuel de Campos Lages Garcia Simão

Setembro, 2023

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Departamento de Engenharia de Eletrónica e Telecomunicações e de Computadores

Security in Hybrid ITS Networks

Ricardo Filipe Quelhas Severino

Licenciado em Engenharia Informática, Redes e Telecomunicações

Dissertação para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

Orientadores : Professor Doutor José Manuel de Campos Lages Garcia Simão
Professor Doutor Nuno Miguel Soares Datia

Júri:

Presidente: Professor Doutor Nuno Miguel Machado Cruz

Vogais: Professor Doutor João Carlos Ferreira
Professor Doutor José Manuel de Campos Lages Garcia Simão

Setembro, 2023

À minha mãe.

Acknowledgments

Firstly, I would like to express my gratitude to my advisors, professors Nuno Datia and José Simão, for their continuous support and availability throughout this year. Their guidance and motivation have played an essential role in the successful completion of this work. I also want to thank everyone at ISEL who contributed, somehow, to my academic or personal development.

Finally, I owe my dear family and friends a profound debt of gratitude. The unwavering support and patience, in good and bad times, have been the cornerstone of my personal and academic success throughout this long journey.

From the bottom of my heart, thank you all.

Abstract

Cooperative Intelligent Transport Systems (C-ITS) continue to be developed to enhance transportation safety and sustainability. However, the communication of *Vehicle-to-Everything* systems is inherently open, leading to vulnerabilities that attackers can exploit. This represents a threat to all road users, as security failures can lead to privacy violations or even fatalities. Moreover, a high fatality rate is correlated with soft-mobility road users. So, in the development of C-ITS systems, it is crucial to broaden the perspective beyond connected vehicles to soft-mobility users and legacy vehicles.

This study presents a novel approach developed in the context of emerging hybrid networks, combining ITS-G5 and cellular technologies. Two protocols, MFSPV and DLAPP, were implemented and evaluated to introduce security guarantees (such as privacy and integrity) in communications within the developed C-ITS hybrid environment. As a result, this work securely integrates G5-connected ITS stations and soft-mobility users through a smartphone application via cellular networks. Real equipment was utilised for this goal, including on-board and roadside units.

Computational, latency and end-to-end times were used to assess the system performance. MFSPV outperforms DLAPP in computational efficiency, but DLAPP achieves a slightly lower network latency. Nevertheless, both only introduce an additional 11% delay in hybrid end-to-end communications. Hybrid communication imposes, on average, an extra 28.29ms of end-to-end time. The proposal shows promise as it reaches end-to-end times below the latency requirements imposed in most C-ITS use cases.

Keywords: C-ITS, ITS-G5, V2X, Cellular Network, Hybrid Network, Security, Privacy, Integrity

Resumo

Sistemas de Transportes Inteligentes e Cooperativos (C-ITS) visam melhorar a segurança e a sustentabilidade dos transportes. No entanto, a comunicação dos sistemas *Vehicle-to-Everything* é inerentemente aberta, levando a vulnerabilidades que atacantes podem explorar. Isto é uma ameaça a todos os utilizadores rodoviários, pois falhas de segurança podem levar a violações de privacidade ou a fatalidades. Além disso, elevadas taxas de mortalidade estão correlacionadas com utilizadores de mobilidade suave. Logo, no desenvolvimento de sistemas C-ITS, é crucial considerar, além dos veículos conectados, os utilizadores de mobilidade suave e os veículos sem a devida tecnologia.

Este estudo apresenta uma nova abordagem desenvolvida no contexto emergente das redes híbridas, combinando tecnologias ITS-G5 e celulares. Dois protocolos, MFSPV e DLAPP, foram implementados e avaliados para introduzir garantias de segurança (como privacidade e integridade) nas comunicações dentro do ambiente híbrido C-ITS desenvolvido. Assim, este trabalho integra, com segurança, estações ITS conectadas por G5 e utilizadores de mobilidade suave, através de uma aplicação móvel via redes celulares. Para tal, utilizou-se equipamentos reais, incluindo *on-board* e *roadside units*.

Tempos computacionais, de latência e de ponta-a-ponta (E2E) foram usados para avaliar o desempenho do sistema. O protocolo MFSPV supera o DLAPP em eficiência computacional, mas o DLAPP atinge uma latência de rede ligeiramente menor. No entanto, ambos introduzem apenas um atraso adicional de 11% nas comunicações híbridas E2E. A comunicação híbrida impõe, em média, 28.29ms extra de tempo E2E. A proposta mostra-se promissora, visto que atinge tempos de E2E abaixo dos requisitos de latência impostos na maioria dos casos de utilização do C-ITS.

Palavras-chave: C-ITS, ITS-G5, V2X, Redes Celulares, Redes Híbridas, Segurança, Privacidade, Integridade

Contents

List of Figures

List of Tables

List of Listings

Acronyms

2FA Two-Factor Authentication. 32
2FLIP Two-Factor Lightweight Privacy-preserving Authentication scheme. 32

AA Authorisation Authority. 17
ASN.1 Abstract Syntax Notation One. 42
AT Authorisation Ticket. 17

BD Biometric Device. 28
BTP Basic Transport Protocol. 10

CA Certificate Authority. 15, 28
CAM Cooperative Awareness Message. 10, 11
C-ITS Cooperative Intelligent Transport Systems. 1, 2, 7, 8
CPA Certificate Policy Authority. 20
CPOC Central Point Of Contact. 20
CRL Certificate Revocation List. 22
CT computation time. 53
CTL Certificate Trust List. 20

DC Distribution Centre. 19
DENM Decentralised Environmental Notification Message. 11
DLAPP Decentralised Lightweight Authentication and Privacy Protocol. 28
DoS Denial of Service. 12

DSRC Dedicated Short-Range Communications. 8

E2E end-to-end. 51, 65, 66
EA Enrolment Authority. 17, 18, 19
EC Enrolment Credential. 17
EN European Norm. 2
ETSI European Telecommunications Standards Institute. 2, 9, 13, 14

FDI False Data Injection. 12

GN GeoNetworking. 10
GUI Graphic User Interface. 45

I2V Infrastructure-to-Vehicle. 10
IEEE Institute of Electrical and Electronics Engineers. 8
IPv6 Internet Protocol version 6. 10
IQR Interquartile Range. 68
ITS-S ITS Station. 2

M_RSU Middleware RSU. 45, 46
MA Misbehaviour Authority. 18
MAC Message Authentication Code. 32
MFSPV Multi-Factor Secured and lightweight Privacy-preserving authentication scheme for VANETs. 30
MITM Man-in-the-middle. 14, 15

NS3 Network Simulator 3. 28

OBU On-board Unit. 2, 4, 32
ONE Opportunistic Network Environment. 32
OS Operating System. 42
OSI Open Systems Interconnection. 9

P_RSU Physical RSU. 45, 46
PKI Public Key Infrastructure. 7, 14, 15, 16
PUF Physically Unclonable Functions. 30

RSU Roadside Unit. 4
RTT Round-Trip Time. 60

SDK Software Development Kit. 43
STRIDE Spoofing, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privilege. 13

TCP Transmission Control Protocol. 10
TD Telematics Device. 32
TLM Trust List Manager. 20
TPD Tamper Proof Device. 28, 30, 32
TS Technical Specification. 16
TVRA Threat, Vulnerability and Risk Analysis. 13

UDP User Datagram Protocol. 10

V2I Vehicle-to-Infrastructure. 10
V2V Vehicle-to-Vehicle. 10
V2X Vehicle-to-Everything. 2, 3
VANET Vehicular Ad-hoc Network. 32

WAVE Wireless Access in Vehicular Environment. 8
WHO World Health Organisation. 1
WSS WebSocket Secure. 45

Glossary

connected vehicles Vehicles communicating via ITS-G5, through an on-board unit. 2

ITS station Functional entity specified by the ITS station reference architecture. For example, OBUs in vehicles or RSUs in road infrastructure (such as traffic lights). 9

legacy vehicles Vehicles without an on-board unit, so not connected via ITS-G5. 2

node Physical computational entity (such as OBUs, RSUs and smartphones). 36

soft-mobility Transport type using non-motorised means. Forms of soft mobility include walking, cycling, skateboarding, running, among others. 2

1

Introduction

This chapter will briefly place the study in a broad context and highlight its importance. Then, the thesis objectives and contributions are presented. The chapter concludes by outlining the document's structure.

1.1 Context

Transportation has always been crucial in human society. It connects people and facilitates the movement of individuals, animals, and goods. Thus, it allows access to essential services and promotes prosperity. However, the growing number of vehicles [46] has led to concerns about road traffic and safety. Despite stricter European road safety regulations [4], accidents persist, leading to fatalities. Moreover, increased road traffic has resulted in congestion, higher gas emissions, and decreased air quality [45].

According to the *Global Status Report on Road Safety* in 2018 [46], from the World Health Organisation (WHO): *"Deaths and injuries resulting from road traffic crashes remain a serious problem globally and current trends suggest that this will continue to be the case in the foreseeable future."*. The report also alerts that the number of road traffic deaths worldwide remains unacceptably high, with 1.35 million people dying each year — 8^{th} leading cause of death for people of all ages and the number one cause for children and young adults.

In light of this data, finding strategies to make transportation safer becomes essential. In the last few years, progress has been made in the field of Cooperative Intelligent

Transport Systems (C-ITS) [9], particularly in the architecture of solutions that enable vehicles to exchange information with each other (V2V), with the road infrastructure (V2I), and with pedestrians (V2P), being therefore known as Vehicle-to-Everything (V2X). The main objective of C-ITS is to integrate communication and information technologies with road elements cooperation to provide greater safety, mobility, and sustainability [29], thus addressing the previous issues.

1.2 Problem and Motivation

Despite the potential to significantly improve people's lives, the communication of C-ITS / V2X systems is inherently open. This openness creates vulnerabilities that attackers can exploit, representing a significant threat to all road users, as security failures could lead to privacy violations (e.g., entity theft or tracking) or even fatalities. These security and privacy challenges must be addressed to ensure that road safety is not compromised [27]. According to Serban *et al*. [39], "*security plays a crucial role in co-operative applications because a security breach can easily lead to human casualties*".

Furthermore, the operation of V2X systems such as C-ITS is built heavily around communication between only vehicles with proper equipment. This issue is also raised by Yoshizawa *et al*. [48] where it is referred that although in the European Norm (EN) 302 665 (V1.1.1), ETSI[1] has defined handheld devices as one of the types of ITS Station (ITS-S), subsequent ETSI specifications have mainly focused on a vehicle-centric view. In this regard, a high fatality rate is correlated with soft-mobility road users [46]. So, in the development of C-ITS-based systems, it is essential to broaden the perspective beyond connected vehicles, considering the needs of soft-mobility transportation users (e.g. pedestrians and cyclists) and legacy vehicles that do not have the necessary equipment, On-board Units (OBUs).

Next, some descriptions of case scenarios where the aforementioned issues become apparent. For instance, neglecting C-ITS security could lead to:

- A malicious vehicle sending false observations about the road (e.g., road hazards) and biases other vehicles to "believe" its incorrect observation, implying a change in their behaviour.
- Privacy violations, e.g., by disclosing personal information or tracking a vehicle by collecting ITS messages.

[1] European Telecommunications Standards Institute (ETSI)

- Sybil attacks, where the attacker claims the existence of fake vehicles at multiple locations, creating confusion and disrupting communication. In a V2X scenario, this could lead to inaccurate traffic information (e.g., fake congestion), which could cause accidents due to misinformed decision-making by drivers (Figure 1.1).

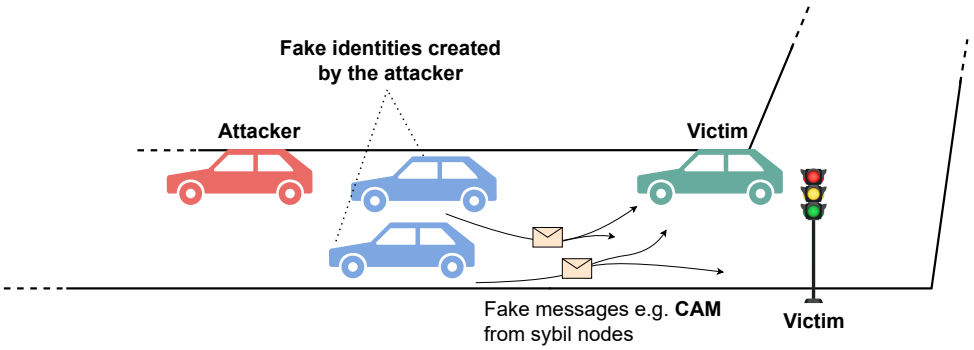


Figure 1.1: Sybil attack example where the attacker claims his existence at multiple locations.

- The situation illustrated in Figure 1.2 represents one reason why it is important to include soft-mobility users and legacy vehicles in a C-ITS ecosystem. In this scenario, an accident occurred, and vehicles (legacy and G5-connected) are approaching. In this situation, the cyclist and pedestrians present could securely notify the ITS systems. Information about this event could then be disseminated to legacy and connected vehicles. Thereby providing more information and cooperation between the various elements of the road.

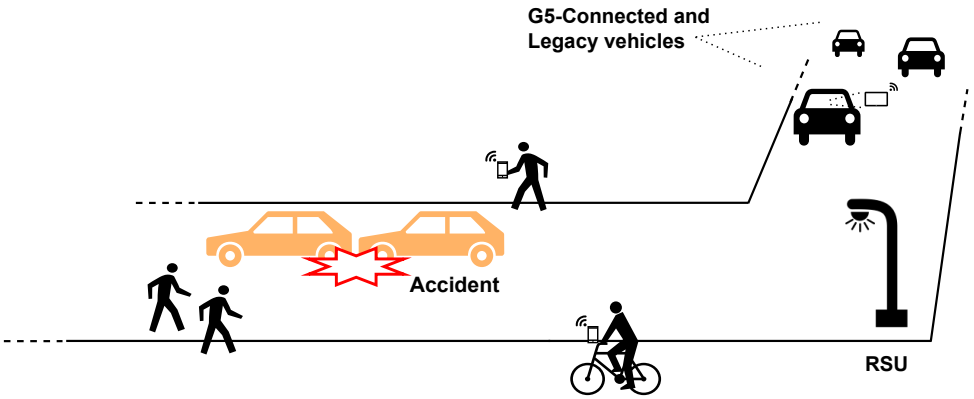


Figure 1.2: Accident scenario where cooperation between soft-mobility users and the G5 network would be beneficial.

1.3 Objectives and Approach

To address the previous issues, this thesis aims to develop a proof-of-concept approach that introduces security guarantees within a ITS ecosystem, while accommodating soft-mobility users and legacy vehicles. In addition to this objective, this work also intends to:

- Evaluate and compare security protocols proposed in the literature using real equipment.
- Assess how the security protocols affect performance.
- Measure the performance cost of incorporating soft-mobility users and legacy vehicles within a realistic testing environment.

To accomplish these goals, our methodology/approach is to build and assess a proof-of-concept system that employs a security protocol in a C-ITS environment while operating within a hybrid network, combining ITS-G5 and cellular technologies. Thus, the proposed approach integrates G5-connected ITS stations and soft-mobility users connected through their smartphones via cellular networks.

Two security protocols, DLAPP [26] and MFSPV [5], were implemented using hardware equipment — OBUs, Roadside Units (RSUs) and smartphones. An application was developed for each of these computing environments. In particular, it was developed an Android application for smartphones. These applications allow sending and receiving/verifying protected messages using a protocol. Lastly, computational, latency and end-to-end times were measured and used to assess the system performance.

It is important to note that there was no prior knowledge or experience on our part regarding ITS or in interacting with the respective equipment, nor previous experience in developing Android applications. Consequently, a significant portion of our research efforts for this thesis was dedicated to acquiring the necessary expertise in these areas.

1.4 Contribution

This work's main contribution lies in the presentation, development and assessment of a novel approach that employs a security protocol in a C-ITS hybrid environment.

Thus, securely integrating G5-connected ITS stations (OBUs/RSUs) and soft-mobility users connected through their smartphones via cellular networks.

This work helps fill the gap between theory/simulation and real-world implementations. For example, by providing empirical evidence of DLAPP [26] and MFSPV's [5] effectiveness in securing C-ITS hybrid networks. Thus also providing knowledge about the challenges and feasibility of implementing protocols in ITS equipment. Furthermore, this research goes beyond the conventional focus on connected vehicles by securely accommodating soft-mobility users and legacy vehicles into C-ITS networks, addressing an emerging and critical concern in the transportation safety and mobility field. Thus, our implementation and evaluation of the security protocols and the development of the proposed approach using actual equipment in a realistic environment yields valuable insights for the research community. These insights may be a foundation for future studies and applications in this area.

All the developed code is available and documented on GitHub in [33].

1.5 Document Structure

After contextualising the study, emphasising its importance, and outlining its objectives and contributions, the initial chapter is concluded with an overview of the document's structure. The remainder of this document comprises four chapters and is organised as follows.

Chapter 2 (Background) covers important background information and standards.

Chapter 3 (Securing Classing and Hybrid ITS Networks) analyses some of the related work found in the literature, including the security protocols used in this work.

Chapter 4 (Proposed Approach) presents the proposed approach and describes the system implementation.

Chapter 5 (Experimental Evaluation) describes the testing environment. Then, the experimental results are reported and analysed.

Lastly, Chapter 6 (Conclusions) concludes the document by summarising the developed work and achieved results. The study's main remarks and possible future work are also outlined.

2

Background

This chapter covers important background information and standards. Firstly, Section 2.1 describes relevant aspects of C-ITS technology. Provides an overview of its main concepts, architecture, essential communication strategies, and equipment. Section 2.2 focuses on analysing the major security threats that ITS may face. It also refers to possible motivations/reasons why someone might attempt to breach the security of ITS. Section 2.3 introduces the concept, the need and how a Public Key Infrastructure (PKI) works. Section 2.4, in turn, is based on the previous and provides detailed information on how the C-ITS PKI operates. Lastly, Section 2.5 briefly describes the pub/sub communication pattern.

2.1 Cooperative Intelligent Transport Systems

This section introduces C-ITS, providing an overview of its main concepts, architecture, essential communication strategies, and equipment.

2.1.1 What is C-ITS?

A general overview of V2X systems will be given to better understand C-ITS technology. V2X refers to the communication among road elements, allowing vehicles to exchange information between themselves (V2V) and with surrounding road infrastructure (V2I), such as traffic lights. It also contemplates communication between vehicles

and pedestrians, known as Vehicle-to-Person/Pedestrian (V2P). Essentially, V2X is a collective term incorporating several communication modes that enable road elements to exchange information and coordinate to improve safety, efficiency, and mobility. A simplified representation of the V2X concept is presented in Figure 2.1.

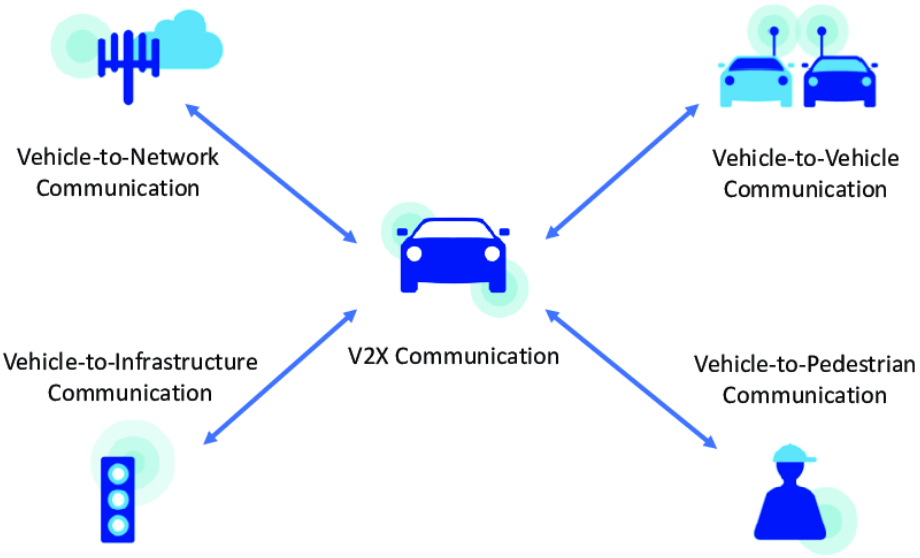


Figure 2.1: Representation of some communication modes incorporated by V2X (extracted from [30]).

V2X communication systems can either use a technology based on IEEE[1] 802.11p protocol[2] that operates in the 5.9 GHz frequency band or a cellular-based approach (C-V2X) [3], the latter it's not the purpose of this thesis. The scenario of interest in this work is the utilisation of the IEEE 802.11p protocol, which is an amendment to the IEEE 802.11 standard designed to standardise vehicular communication systems. 802.11.p is the basis of some standards for V2X communication [48], including:

- Dedicated Short-Range Communications (DSRC) with Wireless Access in Vehicular Environment (WAVE) as the upper layer, in the United States (U.S.).
- Intelligent Transport Systems G5 (ITS-G5) with C-ITS as the upper layer, in Europe.

This dissertation focuses on the use of ITS-G5 for V2V and V2I communications. C-ITS refers to the integration of communication and information technologies with the support of transport infrastructures to provide an improvement in terms of traffic safety

[1] Institute of Electrical and Electronics Engineers (IEEE)
[2] https://en.wikipedia.org/wiki/IEEE_802.11p, accessed on: 2023-04-16

(e.g., collision avoidance applications); mobility (e.g., traffic jam reporting); and sustainability (e.g., reducing harmful gas emissions). Thus, leading to a more efficient and safer transportation [6, 29].

2.1.2 Architecture

C-ITS is composed by multiple sub-systems [13], which include:

- Personal ITS sub-systems in hand-held devices such as smartphones;
- Central ITS sub-system part of an ITS central system;
- Vehicle ITS sub-system present in cars, trucks, and other vehicles;
- Roadside ITS sub-system on traffic lights, poles, and other roadside infrastructure.

Although ETSI has defined handheld devices as one of the types of ITS station (ITS-S), subsequent ETSI specifications have only focused exclusively on a vehicle-centric view [48]. An ITS-S is a functional entity specified by the ITS-S reference architecture [13]. The reference architecture follows the principles of the Open Systems Interconnection (OSI) model for layered communication protocols, which is extended for the inclusion of ITS applications (as Figure 2.2 shows). Each ITS sub-system contains an ITS station, i.e., the functionality described by the ITS station reference architecture.

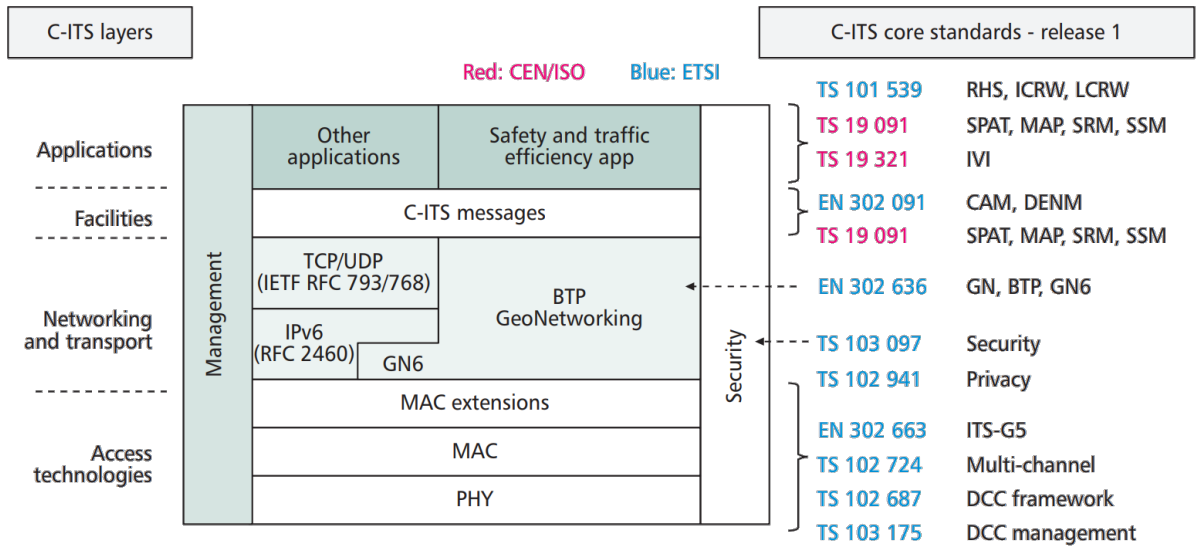


Figure 2.2: ITS-S protocol stack and standards for C-ITS (extracted from [22]).

The ITS-S architecture's protocol stack is organised into several layers (Figure 2.2), each of which serves a specific function in the communication process:

Access Layer — includes technologies such as ITS-G5 and cellular technologies. This thesis uses ITS-G5 for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) / Infrastructure-to-Vehicle (I2V) communications.

Networking and Transport Layer — with two main stacks, **(i)** GeoNetworking (GN) with Basic Transport Protocol (BTP) and **(ii)** IPv6 with UDP or TCP[3]. Which stack to use depends on the application itself. As stated at [22], generally, the IPv6 stack is used for communication with an IP-based infrastructure over a cellular network, while the GN stack is used for ad-hoc communication over ITS-G5 utilising the geo-addressing. Thus, it is designed to allow vehicles to exchange information with each other in a decentralised manner without relying on a central infrastructure.

Facilities Layer — enables application functionality, i.e., supports common message management for data exchange between ITS-S applications. The main messages are CAM and DENM (more detail in Section 2.1.3).

Applications — ITS applications are grouped into "Road Safety", "Traffic Efficiency", and "Other Applications".

Management — mainly managing communications within the ITS station.

Security — provides security services regarding the C-ITS security standard (more detail provided in Section 2.4).

2.1.3 Messages

Many ITS applications require one of two communication strategies [35] or a combination of both. Periodic status exchange — messages needed by apps to know the status of a vehicle or a roadside terminal. Event-driven information — messages about a specific event (mainly found in safety applications). Therefore, ETSI has defined two essential messaging services.

1. **Cooperative Awareness Basic Service** [18], which defines the Cooperative Awareness Message (CAM). This service provides a basic awareness of the surrounding environment by periodically sending status data to neighbouring nodes within a single hop distance. Through the reception of CAM messages, an ITS-S is made aware of other stations in its neighbourhood area, including their positions,

[3]Internet Protocol version 6 (IPv6), User Datagram Protocol (UDP), Transmission Control Protocol (TCP)

movements, and relevant characteristics. The receiver evaluates the relevance of the information contained within the message and acts accordingly.

For instance, CAM are used in safety applications, such as collision avoidance, to detect potential impacts [38].

2. **Decentralised Environmental Notification (DEN) Basic Service** [19], which defines the Decentralised Environmental Notification Message (DENM). It is designed to provide timely and relevant information about the driving environment and traffic events to drivers and other road users. The DENM includes information about the event type, location, and detection time, among other details. The DENM can then be transmitted multi-hop to cover a specific geographic dissemination area, alerting other road users and vehicles (in the region) of the event.

   Use case examples of DENM being transmitted are slippery pavement or post-collision warning [23].

There are other messages in C-ITS [12], but DENM and CAM are the most widely used.

2.1.4 Equipment

To enable communication between vehicles and infrastructure in ITS, two types of equipment are required: On-Board Units (OBUs) and Roadside Units (RSUs). OBUs are installed in vehicles and are mainly responsible for disseminating information related to the vehicle, which is encapsulated in a CAM. These CAMs are received by other OBUs in nearby vehicles (V2V) or RSUs (V2I). An RSU, on the other hand, is a static equipment located along the road or pedestrian passageway, whose primary function is to disseminate DENMs to share information about events on or near the road (I2V).

Nonetheless, the DEN basic service can be implemented in RSUs and OBUs [19]. The cooperative awareness service can also be implemented in both.

2.2 ITS Threat Analysis

ITS communication is inherently open and vulnerable to exploitation by attackers. A proactive approach is crucial to ensure trustworthiness, which involves identifying and mitigating potential threats and risks during development. This section analyses critical security threats and explores potential motivations behind breaching ITS security.

2.2.1 Attacker Motivation

In general, there are always motivations behind the attacks, so understanding them is crucial in security analysis. Based on the different types of attacks, motivation classes can be determined [32]:

- **Organised crime, vandalism, and physical harm** — e.g., Denial of Service (DoS), identity theft and causing an accident.
- **Financial** — e.g., steal private information, insurance fraud or vehicle tracing (privacy violation) to steal them or even abduct people for ransom.
- **Non-financial** – e.g., enhancement of own traffic conditions or to improve its own 'hacker' reputation.

2.2.2 Attack Variants

As found in the literature [27], there are different attack variants. For a better understanding of the attack's nature and better organisation of the threat analysis, they can be categorised as follows:

- **Active or Passive** — In active attacks, the attacker actively interacts with the system, e.g., DoS, False Data Injection (FDI). In contrast, passive attacks describe when the adversary does not interact directly with the system, e.g., eavesdrop on critical data or threaten a user's privacy by linking C-ITS messages.
- **Offline or Online** — Offline attacks are performed when the system is not operational (e.g., physical access to a device). In contrast, online attacks are executed by exploiting the system at runtime.
- **Internal or External** — In internal attacks, adversaries are legitimate ITS nodes and behave "accordingly" to the underlying protocol, e.g., sending false data. On the other hand, in external attacks, the attacker is not an ITS user, e.g., jamming or eavesdropping on the communication to gather private information.

Given this set of classifications, it would be possible to define strategies to address or mitigate certain attacks catalogued with a specific set of variants (e.g., active, online and internal).

2.2.3 Threat Identification

According to the ETSI report – Threat, Vulnerability and Risk Analysis (TVRA) [14], the analysis of ITS security has considered the following five threat categories: **(1)** Availability threats, **(2)** Integrity threats, **(3)** Authenticity threats, **(4)** Confidentiality threats and **(5)** Non-repudiation/Accountability threats.

These categories are almost identical to the categories of threats considered by the STRIDE[4] [1] methodology, which consists of an acronym for six classifications of threats to systems. We can observe (Table 2.1) that the properties compromised by each STRIDE category are coincidental with the aforementioned threats considered by ETSI, except for category 'E'.

Table 2.1: Represents what security property is compromised in each threat the STRIDE method considers.

STRIDE Category	Threat	Property Compromised
S	Spoofing	Authentication
T	Tampering	Integrity
R	Repudiation	Non-repudiation
I	Information Disclosure	Confidentiality
D	Denial of Service	Availability
E	Elevation of Privilege	Authorisation

Apart from the five categories outlined by ETSI, privacy threats will also be taken into account as a major threat category [10, 27] since if it is compromised, it will have a great negative impact on users' trust in C-ITS. Going a little deeper into the details of each of the threats considered, we have the following:

Privacy Threats — Privacy is critical due to the sensitive nature of the transmitted data. One of the most significant privacy threats in ITS is the risk of access to personal data, such as location and travel patterns. The consequences of privacy breaches in V2X / ITS can be severe [10], as they could destroy public trust in these ecosystems, leading to reluctance to use them.

Availability threats — Threats to the availability and continuous behaviour of an ITS system include DoS attacks, which aim to make the system unavailable to legitimate users. It can cause an ITS station to fail in receiving, responding, relaying, generating, and transmitting traffic safety messages, for example, maliciously generating a high volume of false messages.

[4]Spoofing, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privilege (STRIDE)

Integrity threats — Threats to the integrity of an ITS-S include manipulating and corrupting information.

Authenticity threats — All ITS stations can send, receive and replay messages. Thus, it is crucial to guarantee the authenticity of the information received by the ITS stations. Some examples of attacks are:

- *Masquerade*, where a malicious node inserts false messages into the network, pretending to be another node.
- *Sybil*, being considered [27] one of the main threats against the security of an ITS network. This attack consists of a vehicle pretending to have more than one identity (as exemplified in Figure 1.1). Sybil attackers may launch DoS attacks, waste network bandwidth, and destabilise the network.

Confidentiality threats — Threats to the confidentiality of information associated with an ITS station happen when sensitive information can be leaked/revealed. Confidentiality is only required for selected services, and there are two versions of the ETSI technical specification about *Confidentiality Services* [16, 17].
The latest version is from 2022 [17], however is still in development (has no information). The first version (2012) [16] indicates that for cooperative awareness basic service (CAM messages), no confidentiality services are needed. Although there is no specification about the DEN Basic Service (DENM messages), it is expected (as it is a multi-hop message) that confidentiality is also not needed.

Non-repudiation threats — ITS end-users may be able to avoid criminal prosecution by denying their involvement. For instance, by executing attacks on the ITS network or doing road traffic crimes and denying it (property of non-repudiation compromised).

2.3 Public Key Infrastructure

PKI plays a crucial role in ensuring the security of digital communications. In cryptography, a public key is used for asymmetric encryption. Nevertheless, the challenge lies in verifying that the public key belongs to the intended recipient, as there is always a risk of a Man-in-the-middle (MITM) attack. As depicted in Figure 2.3, a MITM attack occurs when a malicious third party (e.g., Mallory) intercepts communication between two parties (e.g., Alice and Bob) and masquerades itself as the intended recipient, potentially stealing sensitive information.

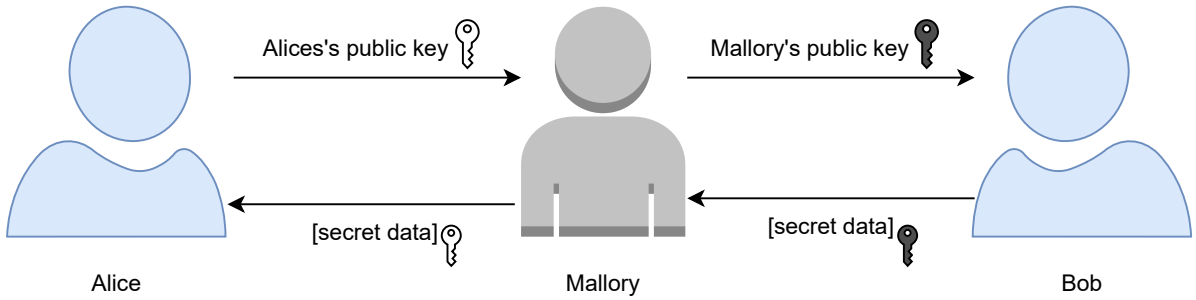


Figure 2.3: Example of a MITM attack on the exchange of a public key (based on [37]).

To prevent this, PKI provides a solution using digital certificates, which assign identities to keys and verify the ownership of public keys while preventing MITM attacks. By using PKI, organisations and individuals can have trust in the authenticity and security of their digital communications.

There are two main elements in a PKI general architecture.

- **Certificate Authority (CA)**: entities responsible for verifying the identity of users and issuing signed digital certificates. These duties are carried out by organisations and companies that have established themselves as a trustworthy certificate authority.
- **Digital Certificates**: digital documents issued to people and systems to uniquely identify them in digital communications. It proves the ownership of the public key mentioned in the certificate, i.e., the receiver can verify the signature to ensure the certificate's integrity because they are signed by CAs (the security of digital certificates relies on the trust placed on CAs). The generic constitution of a digital certificate is as follows:
  - **Subject**: Identifies the entity to which the certificate belongs.
  - **Public Key**: The public key of the entity that corresponds to a private key held by the entity.
  - **Issuer**: Identifies the CA that issued the certificate.
  - **Signature**: A digital signature from the issuer (CA) confirms the certificate's authenticity and the binding of the public key to the entity.
  - And others.

There are several CAs worldwide, and they are organised in a hierarchical structure (Figure 2.4). CAs at the top of the hierarchy are called root CAs. They can issue certificates directly for end clients or intermediate certificates to be used by intermediate

CAs (subordinates). Intermediate CAs may also apply this logic to their subordinates, forming a chain of trust.

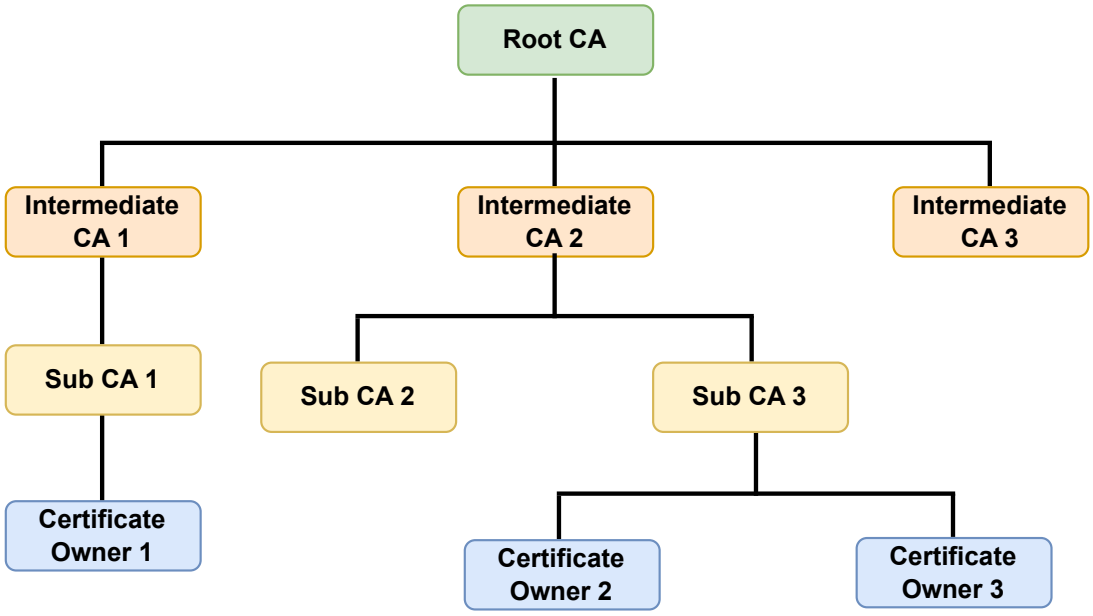


Figure 2.4: Hierarchy of certificates authorities (inspired on [44]).

Chain of Trust Validation

A recursive method is applied to verify each certificate in the chain. First, to validate the leaf certificate's public key, we validate its certificate by using the issuer's public key (verifying the digital signature). This method is applied until the stop condition is met, a self-signed certificate — trust anchor. Lastly, it is successfully verified if the thrust anchor is on the system's trusted CA list (as depicted in Figure 2.5).

Finally, it is worth mentioning that certificates only keep the public key. The private key is "associated" with the certificate in its storage. One example is the Portuguese citizen's card, where the certificate can be exported, but the private key cannot be extracted from the card's memory.

2.4 PKI as an Architecture for Securing ITS Communications

The preceding section provided a general overview of how PKI operates. Hereupon, PKI is also a building block for Cooperative-ITS security, being also referred to as — *C-ITS Communications Security Architecture and Security Management* — in the ETSI Technical Specification (TS) 102 940 [15]. This section describes its global architecture, major elements and roles, as well as highlights its strengths and limitations.

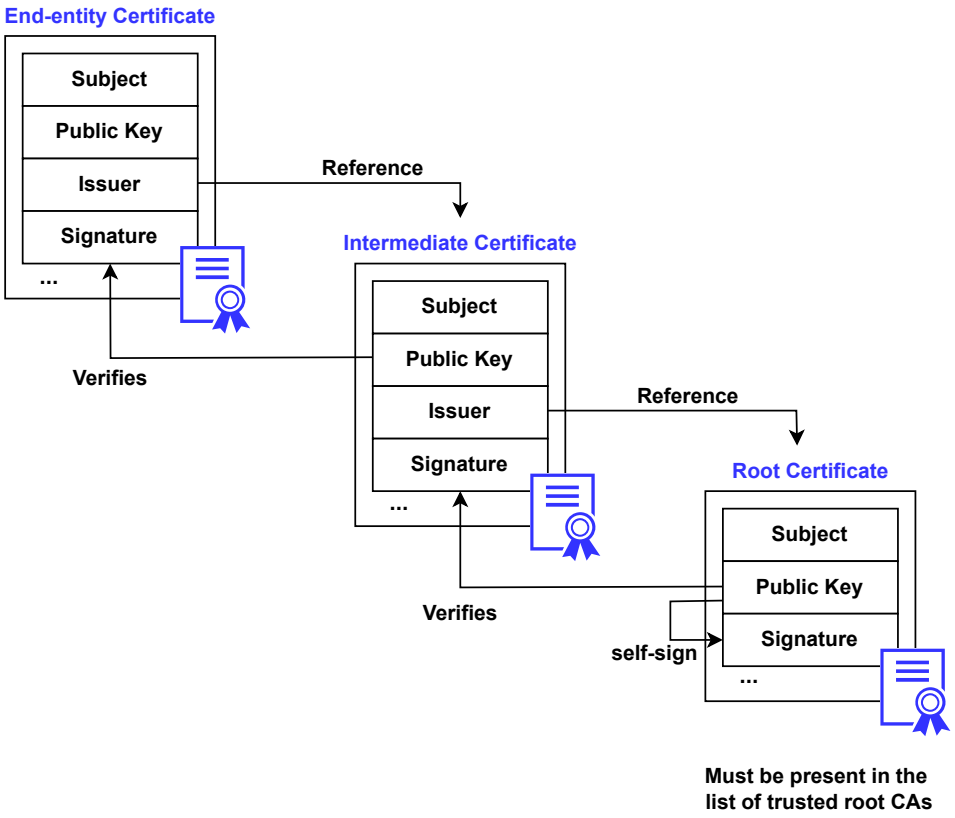


Figure 2.5: Schema representing the chain of trust validation logic.

2.4.1 Architecture

The most relevant elements in the architecture are as follows:

- **ITS Station (ITS-S)** – This entity is the end device. For instance, an OBU in the vehicle and an RSU in road infrastructure.
- **Root CA** – Trust anchor in the certification hierarchy. In C-ITS, it provides Enrolment Authority (EA) and Authorisation Authority (AA) certificates to authorise them to issue, respectively, Enrolment Credentials (ECs) and Authorisation Tickets (ATs) to ITS-S.
- **Enrolment Authority (EA)** — Entity responsible for management of ECs. It issues one EC per ITS-S to authenticate and grant it access to ITS communications. An EC is a long-term certificate and is considered a proof of identity, thus used to identify and authenticate an ITS-S within the PKI [25].
- **Authorisation Authority (AA)** – Entity responsible for issuing ATs, allowing the ITS-S to use specific ITS services. ATs, also known as pseudonym certificates, are

short-term certificates that mask the identity of an ITS-S while simultaneously proving it is authenticated and authorised to access communication resources and services.

- **Misbehaviour Authority (MA)** – Responsible for processing misbehaviour reports and can detect and revoke misbehaving ITS stations, excluding them from the C-ITS trust domain.

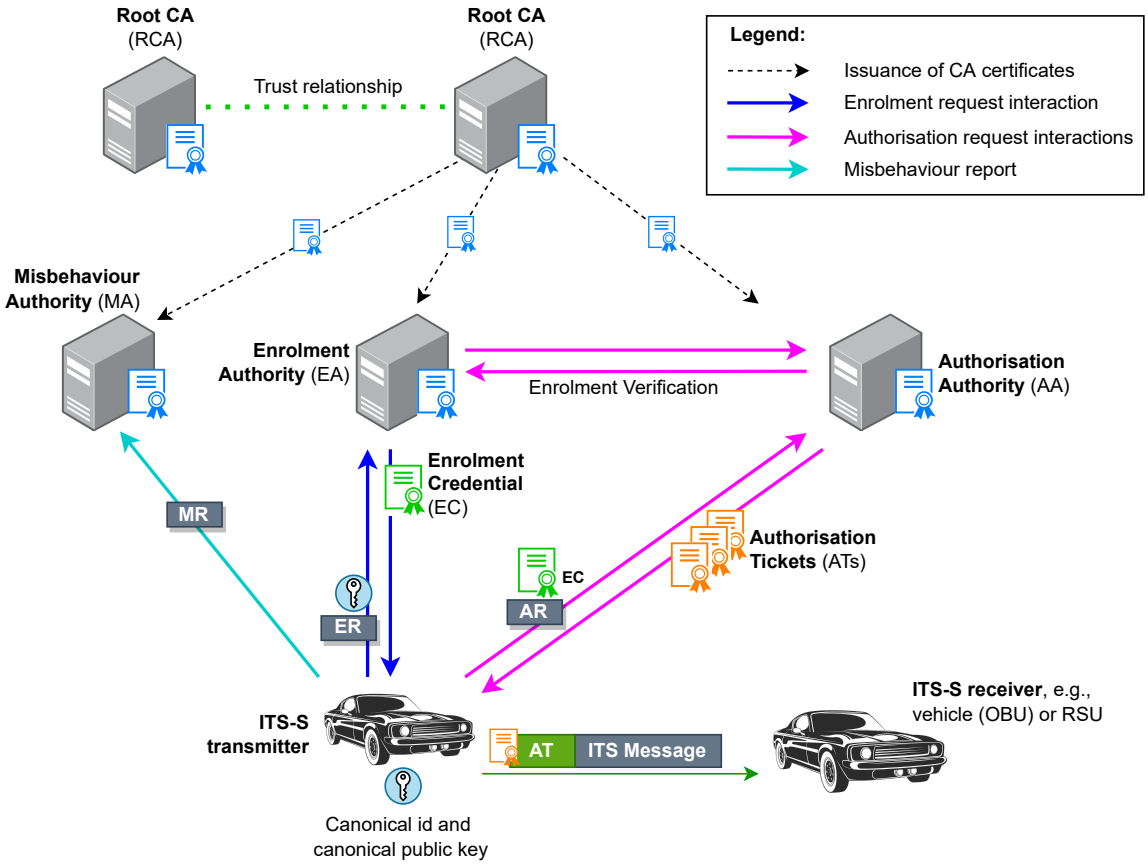


Figure 2.6: PKI architecture in C-ITS (based on [15]).

The sequence of steps that occurs from when a vehicle intends to enter the C-ITS network until it is enabled to send messages to another ITS-S (as exemplified in Figure 2.6) is as follows.

1. As each ITS-S needs to be registered at one EA. The sequence starts with the ITS-S requesting the EA for permission to access ITS communications – *enrolment*.

   The request message carries the ITS-S station's unique identifier, provisioned with associated credentials during the initialisation process. These identify the ITS-S by the PKI and include at least a public and private key pair. If the request

is successful, EA issues an enrolment credential for the ITS station, which is then used to validate its identity when accessing ITS communications.

2. ITS-S requests AA authorisation to invoke ITS services (e.g., to send CAM and DENM).

   When the AA receives this request, it sends a request to EA to validate the ATs request made by the ITS-S. If the answer is positive, the AA sends authorisation tickets to the ITS-S.

3. The ITS-S transmitter can now send secured messages (e.g., CAM) to other ITS-S.

This sequence is depicted in the sequence diagram in Figure 2.7.

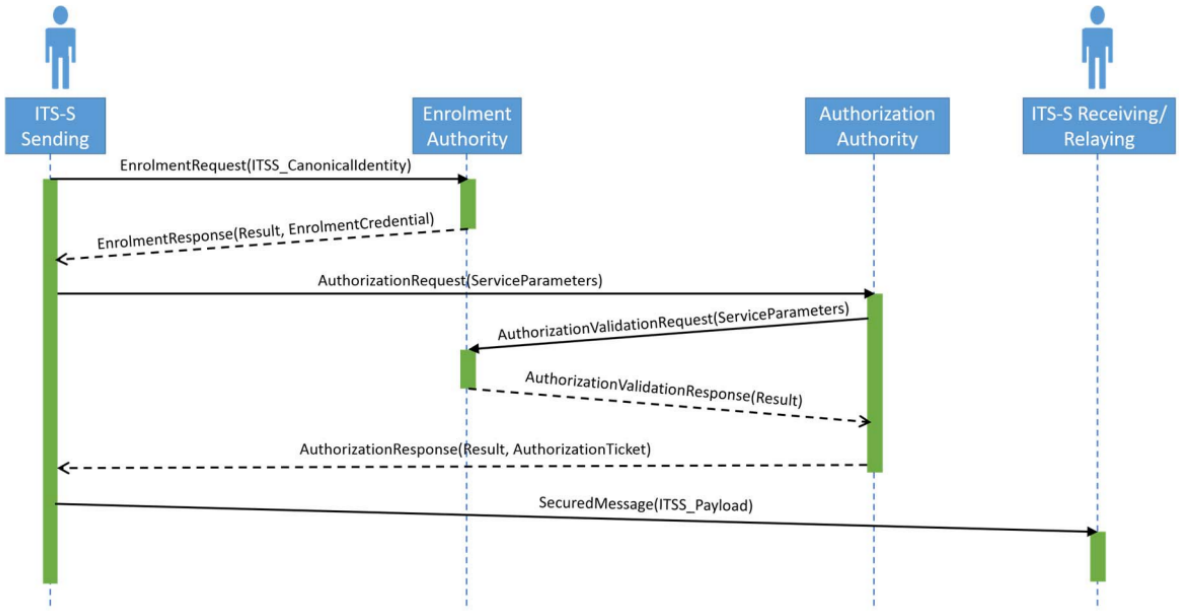


Figure 2.7: Sequence to achieve secure message exchange between ITS stations using C-ITS PKI (extracted from [10]).

Aside from the elements referred to, others play a role in this PKI system, such as the Manufacturer, Operator, and optionally, the Distribution Centre (DC). However, since they are not essential to understanding the overall architecture or the sequence of interactions, they have not been included to avoid overloading the explanation or the diagram (Figure 2.6).

2.4.2 Certificate Trust List

Returning to the architecture illustrated in Figure 2.6, we can observe that multiple Root CAs can exist and work together within the C-ITS trust domain. In this case, the

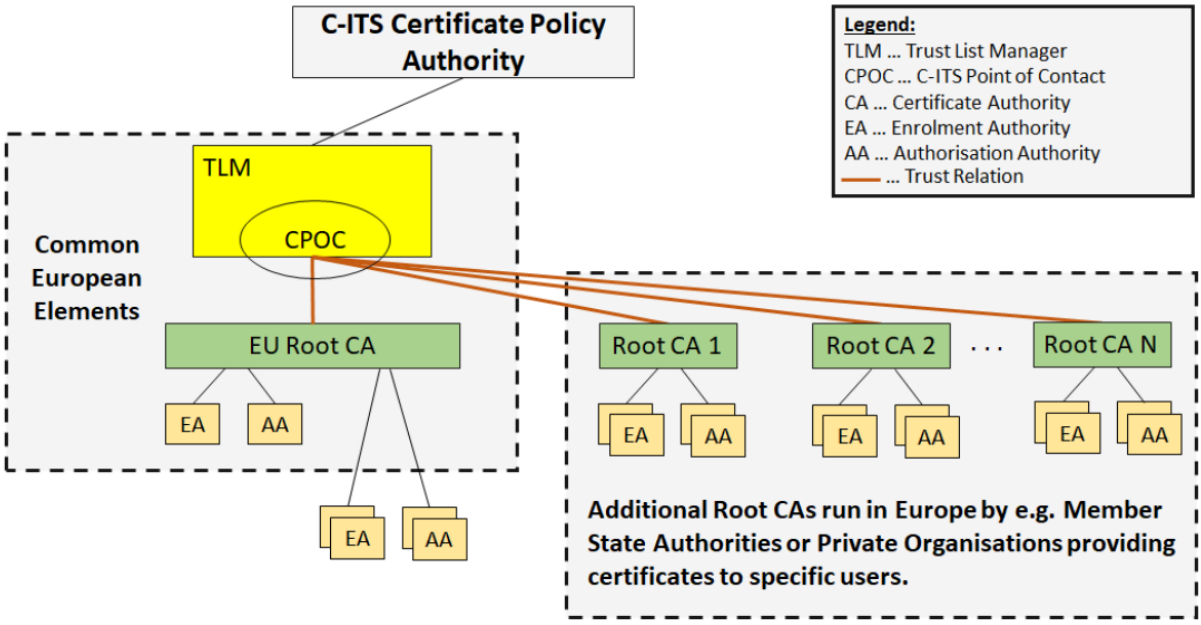


Figure 2.8: C-ITS trust model architecture (extracted from [21]).

C-ITS PKI uses a Certificate Trust List (CTL) approach as depicted in Figure 2.8.

This approach complements the C-ITS PKI architecture of Figure 2.6, now adding the following elements:

- **Certificate Policy Authority (CPA)** is the highest level role in the C-ITS trust system. It comprises representatives from public and private stakeholders (e.g., road operators, vehicle manufacturers, authorities, etc.) who participate in the C-ITS trust model. The CPA is responsible for designating and authorising the Trust List Manager (TLM) and the Central Point Of Contact (CPOC) to operate in the C-ITS trust system. Additionally, it has the authority to determine root CAs' trustworthiness and approve or revoke their operation in the C-ITS trust domain. The CPA communicates its decisions on approved or revoked Root CAs certificates to the TLM.

- **Central Point Of Contact (CPOC)** is an optional entity designated by the CPA. It is responsible for establishing and ensuring communication between Root CAs. It collects the Root CA certificates and provides them to the TLM.

- **Trust List Manager (TLM)** is appointed by the CPA and operates under its authority. The TLM's primary function is to ensure that all ITS environment communication is secure and authenticated. To achieve this, the TLM compiles a — *Certificate Trust List* — which contains the certificates from all the root CAs and

subordinate CAs authorised to issue certificates within the C-ITS PKI. This list is periodically updated and distributed to all ITS stations, enabling them to verify the authenticity of received messages and the validity of any associated certificates. This list is updated after CPA approval, e.g., when a Root CA certificate is created, renewed or revoked.

2.4.3 Strengths and Limitations

Strengths

Handling privacy issues — V2X messages include several identifiers (such as MAC address and station ID) which can be used to track ITS users and cause privacy violations (e.g., by eavesdropping on its CAM). As a countermeasure to this issue, users' privacy is protected by a pseudonym scheme, i.e., frequently changing the ATs (pseudonym certificates) used to authenticate messages. Each time an ITS-S changes its AT, it also changes all its identifiers.

To guarantee ITS-S privacy even from the entity's (EA and AA) point-of-view, the architecture separates the EA and the AA into two distinct entities. EA knows the real identity of the ITS-S but does not know its AT (or pseudonym), whereas AA knows the ITS-S AT but not its real identity.

Message Integrity, Authenticity and Non-repudiation — These three properties can be ensured by creating a digital signature (using an AT) over the message payload. The ITS-S transmitter can use its AT to generate a digital signature for an outgoing message. When a signed message is received, the receiver can verify the digital signature by performing a chain of trust validation as seen in Section 2.3. Successful digital signature verification implies that the content of the message is not altered, and only the transmitter ITS-S can generate that specific message and signature, thus achieving authentication/integrity and non-repudiation.

Confidentiality — Can be ensured by encrypting the packets with a key shared with the ITS-S receiver [32].

Limitations

Despite the high-security level of the C-ITS PKI solution, it also has its limitations. The major drawbacks of PKI are due to processing delay and communication overhead [26].

The first one — *processing delay* — is because PKI uses asymmetric cryptography to sign and verify each message, which is quite computationally demanding. According to ETSI [18] the maximum generation rate for CAM messages is 10 Hz. Each vehicle may

receive many signed messages every 100 ms in such a scenario. The ability for each vehicle to check its Certificate Revocation List (CRL) for a large number of certificates and verify the senders' signatures on the received messages in a timely manner forms an inevitable challenge to C-ITS efficiency requirements [43].

Lastly, C-ITS PKI experiences — *high communication overhead* — because the sent certificate (AT) allows the receiver to verify the message. Compared with the original message, the large size of the attached certificate implies a non-negligible overhead [2], which is not ideal in ITS communications, where the lowest possible delay is desirable.

2.5 Publish/Subscribe Communication Pattern

The publish/subscribe communication pattern (also known as pub/sub) comes with the need for an application to announce events to multiple interested consumers asynchronously, decoupling the senders from receivers, thus offering a different approach than the conventional client-server architecture. As pub/sub will be used in the proposed approach, it will be briefly described.

In a typical client-server model, a client initiates the communication, and the server provides the requested service (direct interaction). With pub/sub, the client(s) sending messages (referred to as publishers) and the client(s) receiving them (subscribers) are decoupled and don't interact directly with each other. Subscribers and publishers are not even aware of each other's existence. The intermediary component (broker) manages the connection between them. In other words, the pub/sub pattern consists of three main components: publishers, subscribers and the message broker.

- **Publishers** produces messages (of a certain topic) and publishes them in the message broker.
- **Subscribers** receive messages about a previously subscribed topic (this interest was registered with the broker).
- A **broker** acts as an intermediary, filtering and forwarding messages (received from publishers) to subscribers.

The broker is vital in pub/sub as it filters incoming messages and distributes them to the appropriate subscribers. The broker can have multiple options for this "filtering process" (e.g., topic/-based filtering), where messages can be filtered based on the

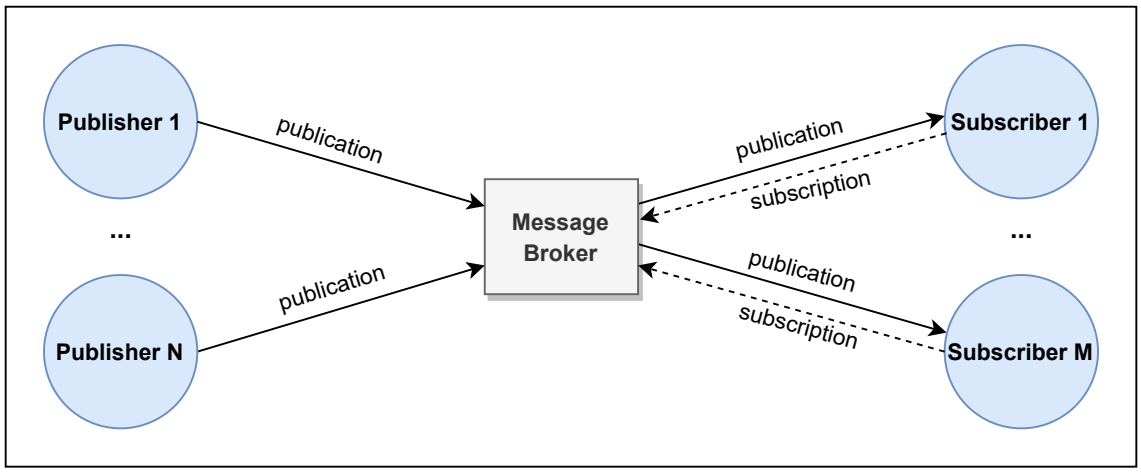


Figure 2.9: Publish / Subscribe communication pattern.

topic or subject to which they belong. This allows the broker to route messages only to subscribers interested in a specific topic or subject.

Given this brief introduction, pub/sub has several benefits, mainly because it proposes an indirect interaction between publishers and subscribers. These characteristics make it possible to achieve decoupling in multiple dimensions – space, time and synchronisation [31].

- **Space decoupling** — Clients (publishers and subscribers) do not need to know each other's location.
- **Time decoupling** — Publishers and subscribers do not need to be executed simultaneously.
- **Synchronisation decoupling** — publisher's and subscriber's operations do not need to be synchronised.

3

Securing Classic and Hybrid ITS Networks

This chapter reviews the research carried out in the literature related to the dissertation topic. Firstly, Section 3.1 reports the existing investigation on the integration of hybrid networks in ITS. Section 3.2 showcases the study conducted on protocols that introduce secure guarantees in ITS communications.

3.1 Hybrid ITS Networks

This section discusses research works on hybrid network integration in ITS. Some studies focus solely on hybrid networks, while others include security aspects.

Recalling, as a high fatality rate is correlated with soft mobility road users and as C-ITS is built heavily around communication between only vehicles with OBUs, this work approach includes development in the context of emerging hybrid networks, combining ITS-G5 and cellular technologies.

3.1.1 Proposed Approaches

Gonçalves *et al.* [24] aimed to develop a system capable of enhancing users' awareness regarding potentially dangerous situations around them. They also highlight the relevance of hybrid networks in ITS, stating that among all types of road users who travel

and move daily, those using soft mobility transportation forms are the most vulnerable. The proposed solution uses hybrid networks (G5 and IP), allowing ITS equipment to communicate with other devices over Wi-Fi or cellular networks. For this purpose, the proposed architecture is divided into three main components. **(1)** Wi-Fi and Cellular Network, **(2)** ITS Centre and **(3)** G5 Network.

The ITS centre is the intermediary between the mobile application and the connected elements via the G5 network. Furthermore, the ITS centre contains an MQTT broker and a server communicating directly with RSUs in the G5 network. Lastly, the G5 network includes connected vehicles and infrastructure, such as OBUs and RSUs. Data transmission is bidirectional, enabling the elements on the Wi-Fi and cellular network, as well as those on the G5 network, to send and receive information to and from one another.

Bissmeyer *et al.* [7] analyses PKI (Section 2.4) as a security concept to secure data in hybrid vehicular communications. However, the concept of hybrid networks differs from the one adopted in this study. In [7], hybrid communications are ideally used to support the reliability of communication by using redundant communication technologies. The security concept is described to secure these communications in the presence of multiple radio technologies (present in the "access technologies" layer in Figure 2.2) using different physical channels to transmit V2X messages. In summary, the paper analysed PKI (specified by ETSI [15]) to ensure security in communication between connected vehicles using different access technologies — LTE and ITS-G5.

Lastly, Scholliers *et al.* [36] has made performance measurements of communication between vehicles and infrastructure for ITS-G5 and LTE. The goal was to test the connectivity between different network technologies, allowing fast handovers (between ITS-G5 and LTE) to enable the system to quickly react to changing circumstances, thus exploiting multiple networks and prioritising them by preference, signal strength, among other criteria. Similarly to [7], the adopted hybrid network is also used in the context of using LTE as another access technology through an ITS station.

3.1.2 Summary

The conducted study provided a better understanding of the current knowledge and experiences within the field of hybrid networking in ITS. Nonetheless, the existing literature on this topic remains limited in its research of methodologies akin to the approach taken in this study.

As mentioned, each work has its motivations for the use of hybrid networks. Some approaches, such as the ones followed in [7, 36], focus on using hybrid networks to utilise different access technologies (G5 and LTE) within an ITS station. However, this thesis develops a C-ITS hybrid environment that integrates non-ITS station users through a smartphone application. The study in [24] comes closest to our goal, although it presents a scenario more focused on road safety. In this work, the objective is also to construct a hybrid network architecture and introduce security guarantees, forming a secure hybrid ITS network. Table 3.1 summarises four relevant aspects of each article, all pertinent to this work.

Table 3.1: Summary of articles studied in the context of hybrid networks in ITS. Indicates whether the study: addresses hybrid networks; considers ITS station that uses multiple access technologies; considers users not connected to the G5 network (without OBU); and finally, if security aspects are considered.

Reference	Year	ITS Hybrid Networks	ITS-S with multiple access technologies	Users not connected to the G5 network	Security aspects
[24]	2022	✓	✗	✓	✗
[7]	2019	✓	✓	✗	✓
[36]	2016	✓	✓	✗	✗

3.2 Architectures for Securing ITS Networks

This section presents and discusses proposed security protocols for ITS.

Recalling, the inherently open nature of ITS communication creates potential vulnerabilities and makes the network susceptible to attacks. Therefore, it is essential to ensure that ITS users exchange trustworthy information. The consequences of security breaches in V2X/ITS are severe, as they can compromise road safety.

Regarding architectures to introduce security guarantees in ITS, the security standard can be briefly revisited. The European and U.S. security architectures are based on a Public Key Infrastructure, and the European one was described in Section 2.4. Table 3.2 lists ETSI specifications related to our C-ITS PKI research. Despite the high level of security provided by the C-ITS PKI solution, it also has its limitations. As analysed previously (Subsection 2.4.3), the main drawback of PKI is the lack of efficiency. This limitation is also emphasised in the literature [5, 26, 27, 42]. These constraints are among the main reasons for the academic community to propose other security protocols that could help mitigate these limitations.

Table 3.2: Security specifications consulted for C-ITS PKI research (all extracted from the ETSI website [20]).

Reference	Document	Title	Version
[14]	TR 102 893	ITS Security - Threat, Vulnerability and Risk Analysis (TVRA)	v1.2.1 2017
[15]	TS 102 940	ITS Security - ITS communications Security Architecture and Security Management	v2.1.1 2021
[10]	TS 102 941	ITS Security - Trust and Privacy Management	v1.4.1 2021
[16]	TS 102 943	ITS Security - Confidentiality services	v1.1.1 2012
[11]	TS 103 097	ITS Security - Security Header and Certificate formats	v2.1.1 2021

Next, some protocols proposed in the literature are reviewed. For each one, we will use the terminology and symbols used in the respective paper.

3.2.1 DLAPP

Hakeem *et al.* [26] proposes a Decentralised Lightweight Authentication and Privacy Protocol (DLAPP) for vehicular networks that offer authentication and privacy protection. The protocol utilises two security hardware devices:

- **Biometric Device (BD)** responsible for driver identification and authentication, e.g., using unique characteristics such as fingerprints or facial recognition.
- **Tamper Proof Device (TPD)** responsible for secure cryptographic material storage. The protocol assumes that TPDs make it extremely difficult to compromise the information kept inside the hardware, stating that TPDs are generally presumed impossible to compromise.

The protocol decentralises the CA's tasks by allowing each vehicle to generate its own pseudo-identity and private keys locally, rather than relying on frequent communication with the CA. Thus, preserving privacy and authentication while reducing the communication workload on the CA. This is possible by using the TPD and BD as a local "CA" to provide (local) security services instead of direct communication with the CA. Furthermore, the paper suggests an authentication signature protocol that employs the concept of hash-chain key generation. Additionally, the protocol assumes synchronisation between the various vehicles and that each one is equipped with BD and TPD devices. Their results were obtained by implementing the proposed protocol with the Network Simulator 3 (NS3).

Message exchange

Two major tasks need to be performed before message transmission or reception, namely:

- **Pseudo-identity generation** — To provide anonymity, a group of pseudo-identities is generated (similar purpose to the authorisation tickets in PKI), preventing traceability. Each vehicle generates n pseudo-identities at initialisation to save the generation time.

  The paper claims that the structure of pseudo-identity is defined in a way that allows only the CA to retrieve the real identity.

- **Hash chain generation** — By applying a hash function iteratively, starting with K_s (secret system key) as a seed, is produced a set of keys. Each element of the chain is used later as a signing key for message authentication. The idea is that all legitimate vehicles that belong to the ITS network will have the same set of keys (because they all have the secret system key, K_s), being able to sign and verify other messages.

After these two steps, the vehicle can send and receive messages. To transmit a message, it calculates a MAC using a randomly selected chain key (*ki*) as shown in Equation 3.1. For this, it is also necessary to choose a pseudo-identity from the generated set and to extract the current timestamp.

$$Sig_{ki} = mac_{ki}(PID_i \,||\, m \,||\, T_s) \tag{3.1}$$

where
Sig_{ki} output of the MAC operation (truncated to 12 bytes),
mac_{ki} MAC value for message m using a signing key k_i,
PID_i pseudo-identity of the transmitter vehicle v_i,
m message to transmit (e.g., CAM or DENM),
T_s current timestamp.

The protocol's proposed message format attaches to the message to be transmitted (m), the chosen pseudo-identity, the calculated MAC value, the index of the selected key and the current timestamp (Figure 3.1).

PID_i (20 bytes)	Sig_{ki} (12 bytes)	m (variable)	k_{index} (4 bytes)	T_s (4 bytes)

Figure 3.1: DLAPP's proposed message format.

The receiver first validates the validity of the timestamp T_s. If the timestamp is valid, the receiver uses the received key index to extract the corresponding key from the

locally generated chain and verifies the received MAC. If the calculated MAC, Sig^*_{ki} (via Equation 3.2) and the received one (Sig_{ki}) don't match, the message is discarded.

$$Sig^*_{ki} = mac_{ki}(PID_i \,||\, m \,||\, T_s) \tag{3.2}$$

Results

According to the authors, their simulations, conducted using the NS-3 simulator, demonstrate that the DLAPP can sign 60000 messages per second, which is up to 55 times higher than other protocols that the paper compares itself to. Additionally, the authors state that their proposed protocol achieves a communication overhead reduction of 20% to 85% when compared to other protocols. They conclude that the DLAPP is well suited to time-critical applications such as large-scale V2X networks.

3.2.2 MFSPV

Alfadhli *et al.* [5] proposes a Multi-Factor Secured and lightweight Privacy-preserving authentication scheme for VANETs (MFSPV). The protocol employs a combination of Physically Unclonable Functions (PUF) [28] and one-time dynamic pseudo-identities as authentication factors. The protocol intends to mitigate the heavy dependency that other protocols [26, 42] have on the system key and long-term sensitive data stored in an ideal TPD. It also aims to decentralise the wide domain of the CA into regional domains by assigning an autonomous regional domain key for each region/domain instead of one key for the entire system.

The protocol assumes that every OBU is equipped with both a TPD and a PUF. In the event of any attempt by an adversary to compromise the PUF or remove it, the authors assume that the OBU stops working correctly and destroys its sensitive information.

Message exchange

The instant a message is ready to be transmitted, it is generated a pseudo-identity PID_v as shown in (3.3). Having generated the pseudo-identity, the message hash signature can be calculated as in Equation (3.4).

$$PID_v = h(API_{new}||V_{sk}||ID_v||k_{mbr}) \oplus h(API_{new}||t) \tag{3.3}$$

$$\phi_{vi} = h(PID_v \,||\, R_k \,||\, m \,||\, t) \tag{3.4}$$

where

PID_v	dynamic pseudo-identity,
API_{new}	authentication pseudo-identity,
V_{sk}	vehicle's secret key,
ID_v	vehicle's real identity,
k_{mbr}	PUF's key member,
t	current timestamp.
ϕ_{vi}	message hash signature,
R_k	regional key,
m	message to transmit (e.g., CAM or DENM).

Lastly, the protocol's proposed message format attaches to the message to be transmitted (m), the timestamp, the calculated pseudo-identity and the hash signature. The protocol's proposed message format attaches to the message to be sent (m), the chosen pseudo-identity, the calculated MAC value, the index of the selected key and the current timestamp to the message (Figure 3.2).

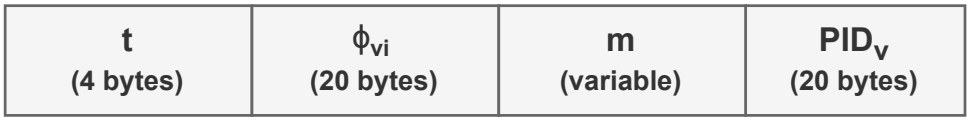


Figure 3.2: MFSPV's proposed message format.

Regarding message verification, when a surrounding ITS station receives the message, it checks the validity of the timestamp. If it is valid, it checks the signature as shown in Equation (3.5).

$$\phi'_{vi} = h(PID_v \,||\, R_k \,||\, m \,||\, t) \tag{3.5}$$

Results

According to the authors, the computation cost for one message verification using the MFSPV scheme is 0.006 ms, which offers from 64.0% to 99.9% lighter computation than the protocols that they compare MFSPV to. Additionally, the authors state that their proposed protocol achieves a communication overhead reduction of 6.4% to 89.2% when compared to other schemes. They conclude that the MFSPV offers superior performance and features over the existing and related schemes.

3.2.3 2FLIP

Wang *et al.* [42] proposed a Two-Factor Lightweight Privacy-preserving Authentication scheme (2FLIP) to improve the security of Vehicular Ad-hoc Network (VANET) communication. This scheme utilises a decentralised certificate authority (CA) and a biological-password-based Two-Factor Authentication (2FA) to achieve its goals. By using a decentralised CA, 2FLIP only requires hashing processes and a Message Authentication Code (MAC) operation for message signing and verification between vehicles. This paper was published in 2015 and used the Opportunistic Network Environment (ONE) simulator to perform the simulations.

The proposed protocol makes some assumptions. Firstly, each vehicle is bonded to one Telematics Device (TD), which is used in conjunction with biometric technology to provide two-factor authentication. This approach combines "something I have" (vehicle's TD) with "something I am" (driver's biometric data). It should be noted that the paper does not address the resilience of the biometric authentication mechanism to potential attacks. Another key assumption is the presence of a TPD embedded in an OBU to store cryptographic materials and perform cryptographic operations. Moreover, the TPD is assumed to be secure against any compromise attempt since it would trigger a self-destruct mechanism to protect the system (similar to [26]). Finally, time synchronisation between ITS-S is also presupposed.

Message exchange

When a $vehicle_i$ generates a new message (e.g., CAM), it calculates an instant pseudo-identity for privacy-preservation. The TPD calculates the MAC of the message, including on it: the pseudo-identity ($PID_{i,ts}$); the message (m) to be sent (e.g., CAM) and the current timestamp (ts). The key used is a system key (common to all vehicles). The receiver performs a MAC regeneration operation to authenticate the message. If equal, the message is accepted, otherwise it's rejected.

Results

According to the authors, the simulated performance results (using ONE) have shown that 2FLIP achieved nearly zero network delay and a 0% packet-loss ratio. Furthermore, compared with previous schemes (that the paper compares itself to), the 2FLIP protocol is said to significantly reduce computation cost by 100-1000 times and reduce communication overhead by 55.24%-77.52%. The results were obtained through simulation, so the protocol performance may vary with real scenarios and equipment.

3.2.4 Summary

As we have seen, despite the high level of security provided by the C-ITS PKI solution, it also has its limitations. The main drawback of PKI is the lack of efficiency due to the use of asymmetric cryptography and the large size of the attached certificate. The papers introduced previously [5, 26, 42] present lightweight protocols that attempt to mitigate these constraints. Then, in conclusion, Tables 3.3, 3.4 and 3.5 are presented summarising the results obtained in each protocol (according to their respective papers) in terms of security properties and protocol efficiency.

Table 3.3: Computation time comparison for message signature and verification in milliseconds (ms).

Max. messages per second	**Wang *et al.* [42] 2FLIP**	**Hakeem *et al.* [26] DLAPP**	**Alfadhli *et al.* [5] MFSVP**
Signature	0.058	0.0167	0.018
Verification	0.0167	0.0167	0.006

Table 3.4: Comparison of the communication overhead introduced by each protocol when sending a message (e.g., CAM or DENM).

Efficiency metric	**Wang *et al.* [42] 2FLIP**	**Hakeem *et al.* [26] DLAPP**	**Alfadhli *et al.* [5] MFSVP**
Overhead (bytes)	47	40	44

Table 3.5: Comparison of the security properties achieved by each protocol (according to the respective paper).

Major security properties	**Wang *et al.* [42] 2FLIP**	**Hakeem *et al.* [26] DLAPP**	**Alfadhli *et al.* [5] MFSVP**
Integrity & Authenticity	✓	✓	✓
Privacy	✓	✓	✓
Non-repudiation	✓	✓	✓
Resistance to DoS	✓	✓	✓
Resistance to message replay attack	✓	✓	✓
Secure System Key Update	✓	✓	✓
Traceability	✓	✓	✓

4

Proposed Approach

This chapter presents the approach proposed to achieve the objectives outlined in Section 1.3. First, a high-level overview of the approach is given. Section 4.1 shows greater detail of its architecture. Section 4.2 provides insight into the implementation.

Recalling, this thesis's main objective is to build and assess a proof-of-concept system that employs a security protocol in a C-ITS hybrid environment. Put simply, the system should enable G5-connected ITS stations to send protected messages that can be received and verified by other ITS stations and mobile applications, and vice versa. As for security, it is intended to implement, evaluate and compare DLAPP [26] and MFSPV [5] protocols using real equipment. To illustrate the approach proposed in this thesis to achieve these goals, a simplified depiction is provided in Figure 4.1. The illustration shows a scenario where all road users can securely communicate with each other within a C-ITS hybrid environment, combining ITS-G5 and cellular technologies. Therefore, the proposal enables communication between G5-connected ITS stations and mobile application users (such as soft-mobility and legacy vehicle drivers). This interaction is achieved by using the ITS Centre as an intermediary.

For example, a soft-mobility user message is transmitted through a mobile application to the ITS Centre, which then relays it via an Ethernet connection to road infrastructure like semaphores equipped with RSUs. These RSUs disseminate the messages over the G5 network to vehicles equipped with OBUs as they pass by. Conversely, messages initially sent via G5 are routed through RSUs to the ITS Centre, which distributes these

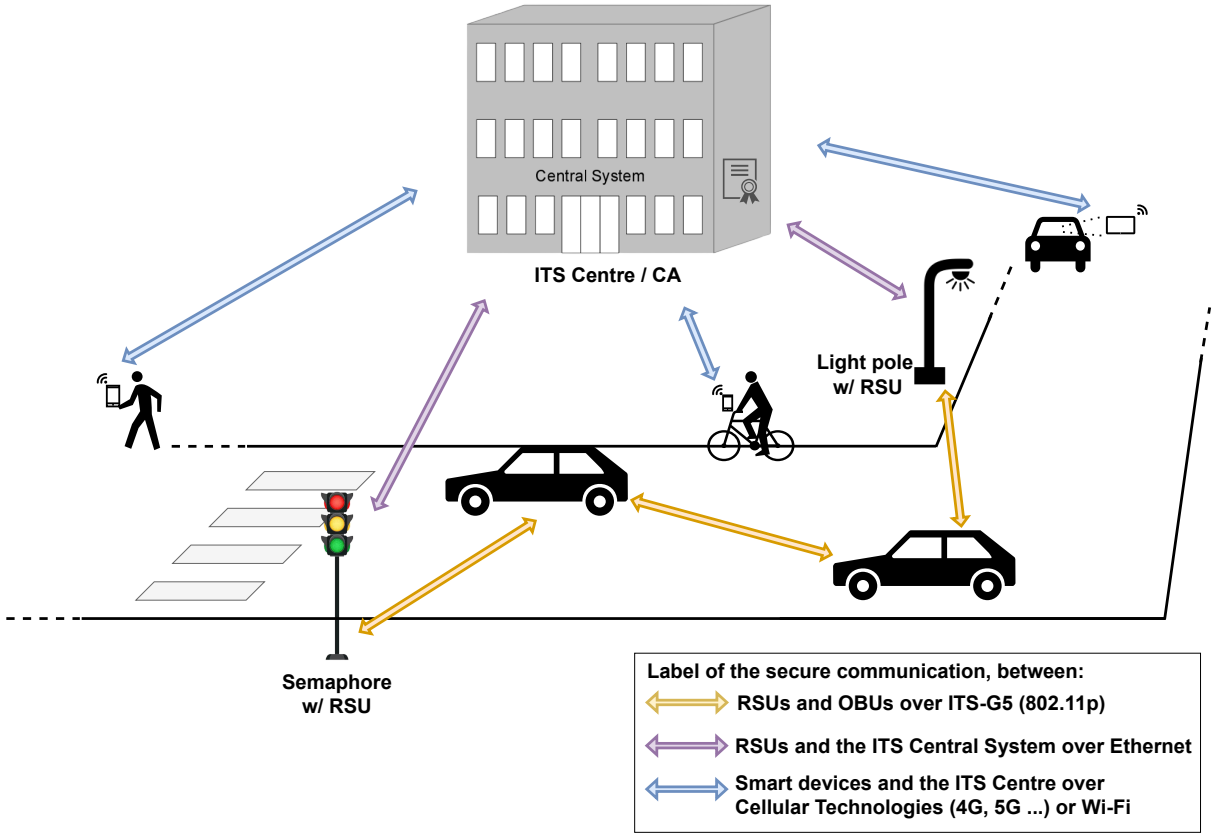


Figure 4.1: Simplified representation of the proposed approach.

messages to mobile applications. This approach establishes a bidirectional communication channel, bridging the G5 and cellular networks. Furthermore, within the hybrid C-ITS ecosystem, every type of node — Smartphones, OBUs, and RSUs — is required to implement the security protocols MFSPV [5] and DLAPP [26], so one of them can be used. Thus enabling information sharing among all road users with security guarantees.

4.1 Architecture

This section describes in greater detail the architecture of the proposed approach and the elements that constitute it.

4.1.1 Domains and Entities

Figure 4.2 presents a more detailed perspective of Figure 4.1. The proposed approach can be separated into three domains: cellular network, ITS Centre and ITS-G5 network.

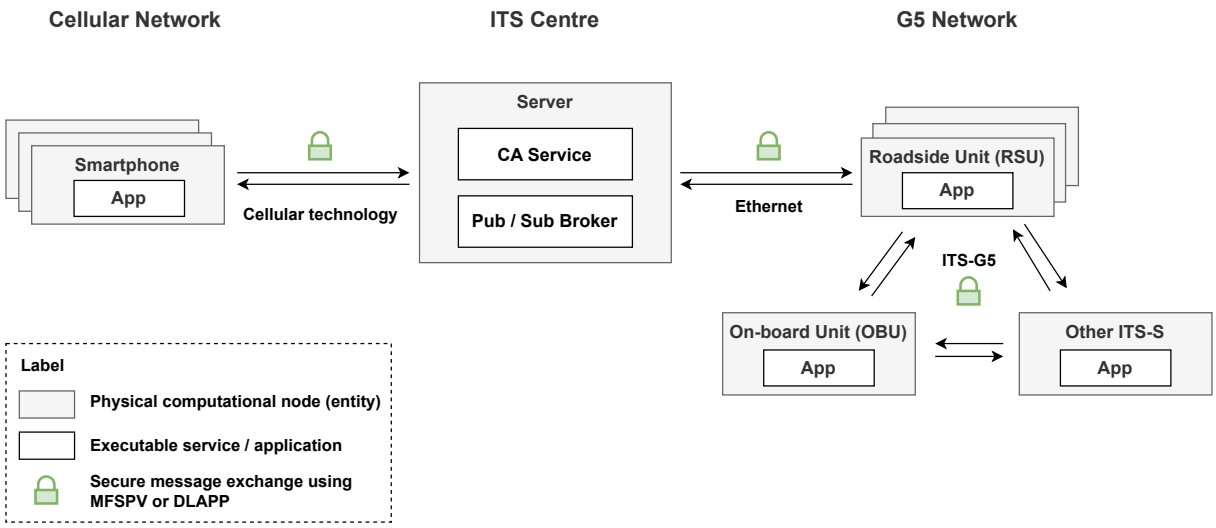


Figure 4.2: Architecture of the proposed approach.

Cellular network — In this domain, the entities are the smartphones used by soft-mobility users and legacy vehicle drivers. The mobile application can receive and verify messages as well as disseminate protected messages.

ITS Centre — Represents an ITS central system. It is composed of a "server" entity that hosts two main services:

- CA Service — As seen in the studied protocols, the CA entity is crucial for user registration, cryptographic material exchange, security updates, among others.
- Pub/Sub broker — As previously described (Section 2.5), the pub/sub communication pattern suits well into a hybrid C-ITS environment since decoupled and asynchronous communications are desired, and multiple producers and consumers exist [31]. Therefore, a broker is critical to enable the ITS Centre to flow data bi-directionally.

ITS-G5 network — The entities (physical computational nodes) present in this domain are the ITS stations, i.e., OBUs in vehicles and RSUs in road infrastructure.

- The OBU application allows them to send, via G5, protected messages (e.g., CAM) as well as receive and verify messages.
- RSUs have direct communication with the ITS-Centre. Application in these units have the capability to: **(i)** send protected messages via G5 and the broker; **(ii)** receive messages via G5, verify them and then send them to the ITS-Centre broker through Ethernet; **(iii)** receive messages from smartphones through the broker, verify and disseminate them into G5 network.

The padlock label in Figure 4.2 indicates secure exchange of messages using MFSPV [5] or DLAPP [26]. Therefore, each application must implement these protocols as they will introduce security guarantees. Note that during message exchanges only one protocol is used. The configuration of which protocol to use is occurs at the initialisation of each application.

4.1.2 Message Exchange Scenarios

The scenarios contemplated for exchanging messages (e.g. CAM, DENM) with security guarantees are illustrated in Figures 4.3, 4.4 and 4.5. These message exchanges assume the configuration of a security protocol (MFSPV or DLAPP) to protect and deprotect (verify) a message.

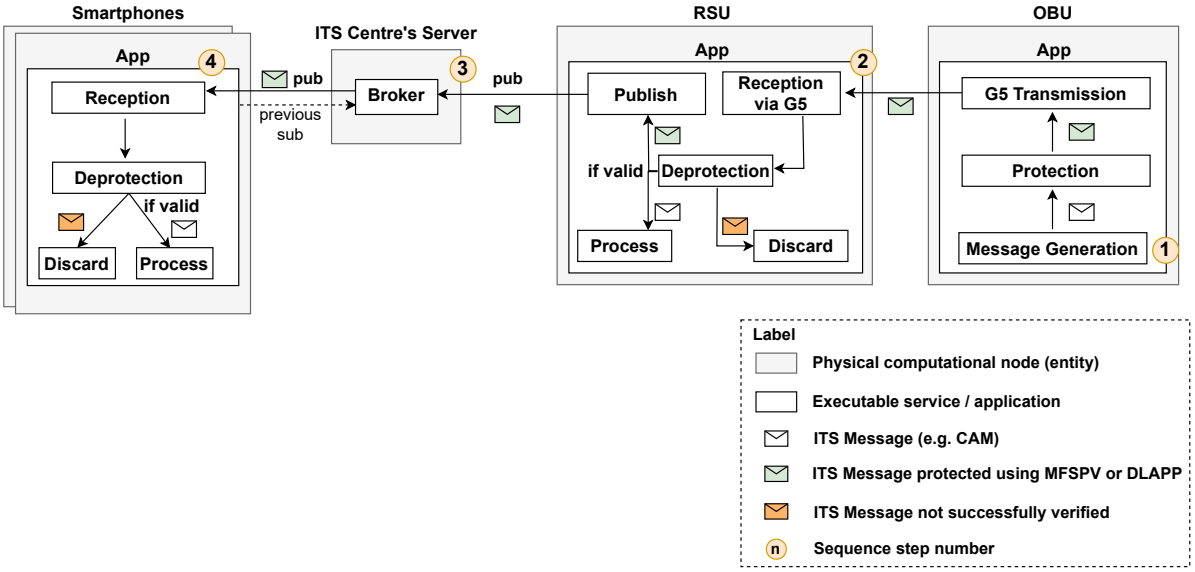


Figure 4.3: Flow diagram when an OBU generates a message.

Firstly, the sequential flow when the OBU generates a message in the G5 network is described in Figure 4.3. The step-by-step is as follows:

1. The OBU generates the ITS message, protects it with the configured security protocol and transmits it via G5.

2. An RSU receives the message and validates it. If the message is valid, the RSU processes and publishes it in the broker. Otherwise, it discards it.

3. The broker receives the message and sends it to all interested consumers.

4. Each interested smartphone receives and verifies the message. If it is successfully verified, then it is processed.

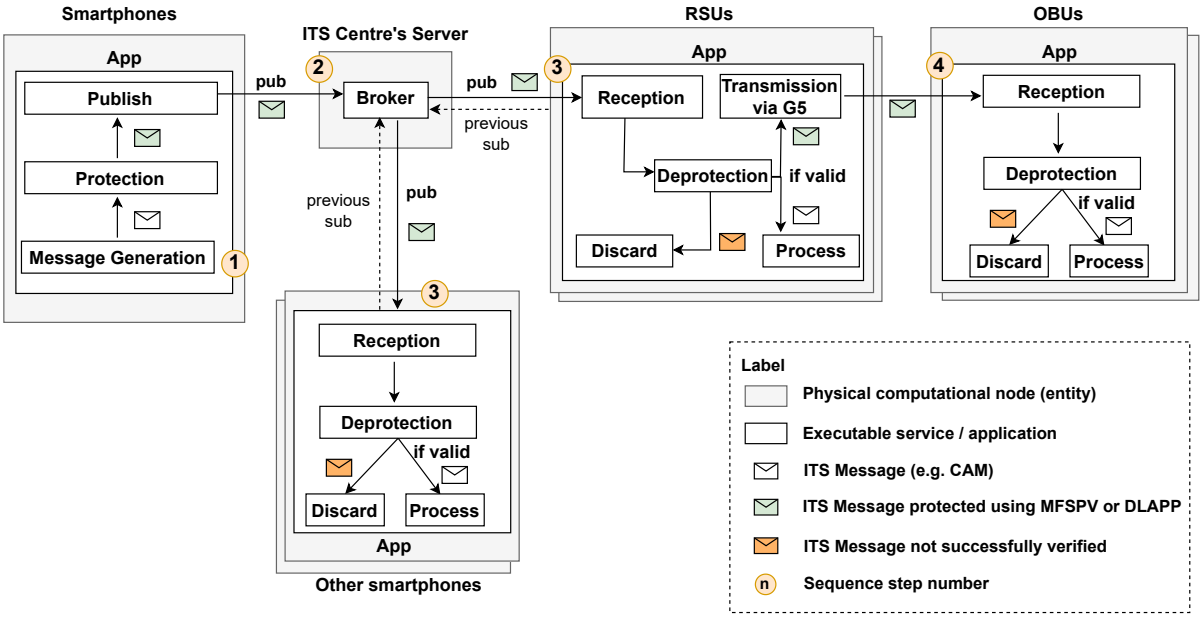


Figure 4.4: Flow diagram when a smartphone generates a message.

Figure 4.4 represents the inverse, i.e., the sequential flow when a smartphone generates a message in the cellular network. The step-by-step is as follows:

1. The smartphone that generates the ITS message protects with the configured protocol and publishes it.
2. The broker receives the message and sends it to all interested consumers.
3. When an RSU or another smartphone receives the message, it undergoes verification. If the validation is successful, the message is processed. RSUs also transmit it via G5 for reception by other ITS stations. In the case of validation failure, both RSUs and smartphones discard the message which is not propagated to the G5 network.
4. OBUs receive the message via G5 and verify it.

Finally, the sequential flow when the RSU generates a message is described in Figure 4.5. The step-by-step is as follows:

1. The RSU generates the ITS message, protects it and sends it to G5 and cellular networks.

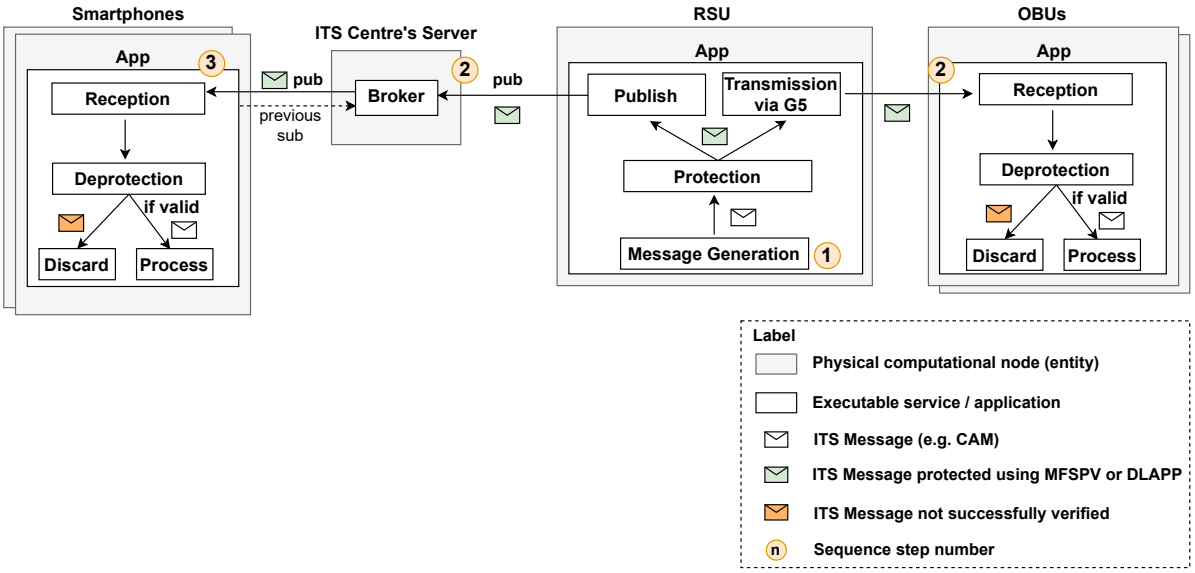


Figure 4.5: Flow diagram when the RSUs generate a message.

2. The broker and OBUs receive the message sent by the RSU. The broker sends it to all interested consumers. Conversely, the OBUs verify the message. If valid, they process it.

3. Each interested smartphone receives and verifies the message. If it is successfully verified, then it is processed.

Note that from the point of view of smartphones, their reception service is identical if the message is generated by the RSU or the OBUs. The same goes for the OBU; its reception logic is identical whether the messages were generated by the RSU or smartphones. This is no longer true for RSUs, as they act as an intermediary between the cellular and G5 networks. RSUs have a service to handle messages from each network. Moreover, it can be observed that there is no need to interact with the CA in any of the scenarios presented. This occurs because both protocols, MFSPV [5] and DLAPP [26], decentralise it so that there is no communication with the CA during the message exchange process.

4.2 Implementation

Priority was given to exchanging messages with security guarantees. Thus, the CA service (present in Figure 4.2) was not implemented. Despite its significance (e.g., in user

registration and cryptographic material exchange), given that this work is a proof-of-concept with time restrictions, it was prioritised implementing applications and functionalities that allowed to achieve the outlined goals and draw the most important conclusions for the study. The CA is not utilised in secure message exchange sequences, as observed in Section 4.1.2. Consequently, when evaluating the protocols and system's performance regarding secure message exchange and hybrid networking — both of which are objectives of this thesis — the presence or absence of the CA does not impact the contemplated use cases or objectives. Excluding the CA, the implementation followed the proposed approach depicted in Figure 4.2. Therefore, the implemented communication scenarios remain identical to those described in Section 4.1.2.

Each application implements and can be configured with one of three security approaches:

1. No security,
2. DLAPP [26] or,
3. MFSPV [5].

The software was modularised to be independent of the security approach in use. It expects an object representing the security protocol, with two methods: '*protection*' and '*deprotection*'. Protection involves applying a security protocol to a message and encapsulating it with the protocol bytes. Deprotection entails verifying a message according to the configured protocol. If the message is valid, the security bytes are then removed. The 'no security' approach was added so that it is possible to assess (experimental evaluation section) the security impact on the performance. In the no-security approach, the methods '*protection*' and '*deprotection*' are identity functions, returning the message passed to them as a parameter.

As the CA was not implemented, each application has the cryptographic material configured locally. Empirically, for each node application (OBU, RSU and smartphone), the DLAPP protocol was implemented with a secret system key k_s of 32 bytes. Each element of the hash chain was obtained using the SHA-256 hash function. Therefore, each key is 256 bits long. In MFSPV, 32-byte keys, such as the V_{sk} and the R_k, were also used.

Each entity's application implementation will be briefly described, highlighting important considerations related to their development.

4.2.1 OBU

The OBU equipment used was the Unex EVK-301E (Figure 4.6), and as previously mentioned, OBUs are the equipment present in vehicles.

Figure 4.6: ITS equipment: Unex OBU EVK-301E (extracted from [41]).

Two main difficulties were encountered during familiarisation with the equipment and, consequently, when implementing its application.

The initial challenge arose from the difference between the execution and development environments. The execution environment was the equipment itself (Figure 4.6), which has an *armv7-a* architecture and a *Linux Yocto*[1] Operating System (OS). On the other hand, the development environment was an *Ubuntu Linux 18.04 LTS 64-bit* OS, whose setup was configured in a virtual machine. Different compiling and running environments led to cross-compile and system library compatibility issues. In particular, problems occurred with the OpenSSL[2] library, whose some modules could not be used due to compatibility issues.

Another challenge was encountered during the application development, namely, how to send messages in the secure message format proposed by the protocols (for instance, as shown in Figure 3.1). First, extending the messages (e.g., CAM) by adding additional fields was attempted. However, ITS messages are structured according to Abstract Syntax Notation One (ASN.1) definitions, leading to strict payload verification. Thus, only valid messages can be encoded. This ASN.1 validation restriction also occurs in the RSU [47]. Given the impracticality of modifying the protocol stack within this work scope, the chosen approach involves directly incorporating the protocol's security bytes into the messages using optional fields. More precisely, the *'PathHistory'* field, which is defined in ITS ASN.1 (as depicted in Listing 4.1), was utilised

[1] https://www.yoctoproject.org/, accessed on: 2023-09-23
[2] https://www.openssl.org/, accessed on: 2023-09-24

within CAM messages. This field is defined in the context of an optional *'lowFrequency-Container'* field. This implementation detail directly influences the message transmission/reception logic on the OBU.

```
PathHistory::= SEQUENCE (SIZE(0..40)) OF PathPoint

PathPoint ::= SEQUENCE {
    pathPosition DeltaReferencePosition,
    pathDeltaTime PathDeltaTime OPTIONAL
}

DeltaReferencePosition ::= SEQUENCE {
    deltaLatitude DeltaLatitude,
    deltaLongitude DeltaLongitude,
    deltaAltitude DeltaAltitude
}
```

Listing 4.1: Excerpt from the ASN definition of the CAM message.

While the *'PathHistory'* field isn't being employed with its intended semantics, it allowed the development of the application according to the established objectives, avoiding the need for protocol stack modifications. While not the most versatile solution, it enabled the utilisation of existing equipment without software changes.

Regarding the development of the OBU's application, it was developed in the C programming language using the *V2Xcast*[3] Software Development Kit (SDK) available for the Unex OBU. As defined in the approach architecture (Section 4.1), the OBU's application has two main services. These are responsible for transmitting locally generated and receiving messages from the G5 network. These services are simplified in Algorithms 1 and 2. As it can be seen, both involve a conversion process. This is necessary due to the previous issue. During transmission, the security bytes are placed in the 'PathHistory' field before transmitting through G5. Conversely, the message arrives with security bytes in the 'PathHistory' field and must be transformed into the expected protocol format to verify it.

[3] https://unex.com.tw/en/v2xcast/, accessed on: 2023-09-29

Algorithm 1 OBU — Message transmission service (pseudo-algorithm)

Require: *security_protocol*: Security protocol object
1: **function** TRANSMIT_MESSAGE
2: *encoded_its_message* ← cam_message_generation()
3:
4: *secured_its_message* ← security_protocol.protection(encoded_its_message)
5: ▷ secured_its_message is in protocol's proposed message format
6:
7: *encoded_its_message_extra* ← transform_format(encoded_its_message)
8: ▷ Insert security bytes into PathHistory
9: ... transmit via G5 ...
10: **end function**

Algorithm 2 OBU — Receive message service (pseudo-algorithm)

Require: *security_protocol*: Security protocol object
1: **function** RECEIVE_MESSAGE(encoded_its_message) ▷ E.g. a received CAM
2: *secured_message* ← transform_to_protocol_format(encoded_its_message)
3: ▷ Convert to protocol's proposed message format
4:
5: *valid_its_message* ← security_protocol.deprotection(secure_message)
6: **if** *valid_its_message* = *None* **then**
7: ... invalid message, discard it ...
8: **else**
9: ... message successfully verified, continue processing it ...
10: **end if**
11: **end function**

4.2.2 RSU

The RSU equipment used in this work was the Siemens ESCoS RSU (Figure 4.7) and, as already mentioned, RSUs are the equipment present in road infrastructure (for instance, traffic lights), connecting this infrastructure to the G5 network.

Figure 4.7: ITS equipment: Siemens RSU (extracted from [40]).

After reading the equipment's documentation [34], no method was discovered to access the RSU and execute applications on it. Instead, only two options, with sufficient information were found to interact with the device.

1. Service Web Graphic User Interface (GUI) — allows the observation of the general status reporting of the RSU, its configuration, and application-specific configuration. The latter enables the sending of handcrafted messages for testing and validation purposes.

2. Interface XFER — WebSocket Secure (WSS) based interface. It provides bi-directional data exchange and device management functions [47]. It enables the issuance of commands to the RSU so that it has certain behaviours, for instance, echoing a received message.

The Web GUI serves a different purpose from application development, so the XFER interface was chosen to help implement the application. However, this approach requires the presence of a client. So, an intermediary component, Middleware RSU (M_RSU), was introduced and developed to address this, acting as an XFER client. For this reason, the Siemens RSU will be referred to as Physical RSU (P_RSU). In this implementation, the combination of these two components can be referred to as the

RSU (Figure 4.8), as both are essential for the expected behaviour of an RSU in the proposed approach. If it were possible to program the P_RSU, the computational logic of the M_RSU would be transferred to it.

In summary, P_RSU is responsible for communication via G5, whereas M_RSU is tasked with managing interactions with the broker. Additionally, M_RSU has most of the logic, including protocol implementations, within a Python application.

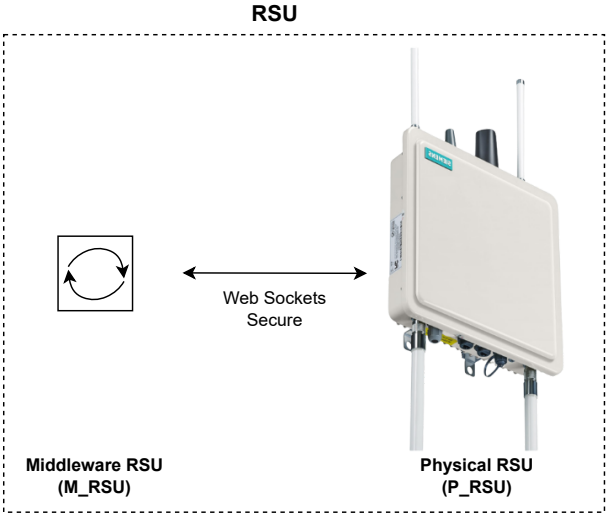


Figure 4.8: RSU entity composed of two elements that will cooperate to carry out the responsibilities of the RSU.

Generically, as seen before, the RSU handles three different message exchange scenarios. Thus, M_RSU and P_RSU act together to perform them. These are:

- Cellular network (Smartphone) → RSU
- G5 network (OBU) → RSU
- RSU → Cellular and G5 network

The behaviour of the M_RSU and P_RSU in each of the above scenarios is described below.

Cellular Network (Smartphone) → RSU
In this scenario, the M_RSU receives the secure message via the broker. Then, it validates the message (*deprotection*), and if it is valid, sends it to the P_RSU's XFER interface, which will forward the message to the G5 network. Before sending it to the P_RSU, the M_RSU first converts the message from the protocol format to include the security data within the message field (PathHistory), per the OBU's requirements.

G5 Network (OBU) → RSU
On M_RSU initiation, it uses the 'subscribe' command to instruct the RSU to forward upstream and downstream messages. Therefore, when the P_RSU receives a message from the OBU, it forwards it to the M_RSU, where it undergoes validation (*deprotection*). If the message passes validation, it is transformed into the format proposed by the protocol in use and published in the broker for smartphones to receive.

RSU → G5 and Cellular network
In this scenario, the messages are being generated by the RSU. Two possibilities were analysed and implemented.

1. Messages generated by the P_RSU, that are forwarded to the M_RSU (by subscription). M_RSU will protect them and send them to the cellular network (via the broker) and the G5 network (via the P_RSU).

2. Messages generated by the M_RSU, that follows a behaviour and sequence of actions similar to the previous case, but instead of initially receiving messages from the P_RSU, the M_RSU generates the messages itself.

4.2.3 Smartphone

Lastly, the smartphone application was implemented as an Android app. Mirroring the dual functionality in the OBU application, it primarily focuses on two services: transmitting locally generated messages and receiving messages from the broker. These services are simply described in Algorithm 3 and 4. Unlike the OBU, where it is necessary to introduce security bytes in the 'PathHistory' field, the Android app sends the bytes in the proposed format.

Algorithm 3 Android app — Transmit message service (pseudo-algorithm)

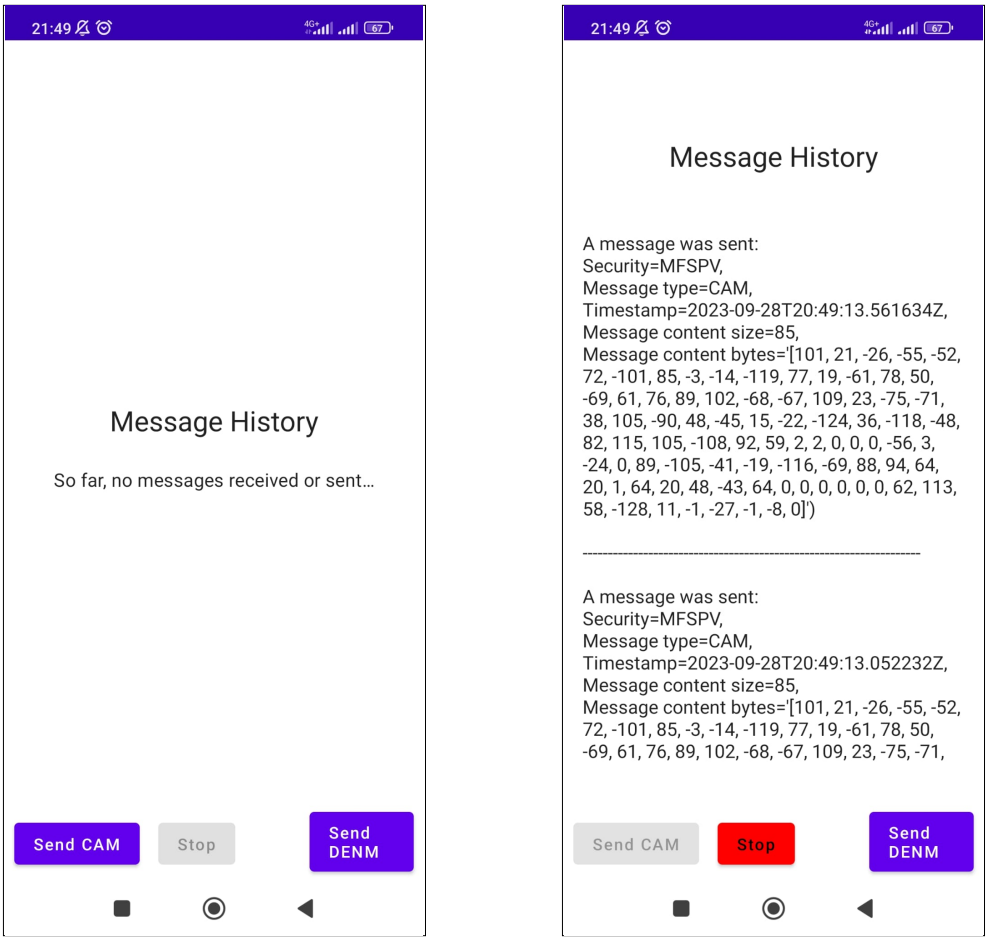
Require: *security_protocol*: Security protocol object
1: **function** TRANSMIT_MESSAGE(encoded_its_message) ▷ E.g. a generated CAM
2: *secured_its_message* ← security_protocol.protection(encoded_its_message)
3: ▷ secured_its_message is in protocol's proposed message format
4: ... publish message ...
5: **end function**

While developing the Android app, functionality was prioritised over user interface design, resulting in the development of a single activity app. This app enables message transmission via buttons (for development and testing purposes only, not intended

Algorithm 4 Android app — Receive message service (pseudo-algorithm)

Require: *security_protocol*: Security protocol object
1: **function** RECEIVE_MESSAGE(secure_message) ▷ E.g. a Broker-received CAM
2: *valid_its_message* ← security_protocol.deprotection(secure_message)
3: **if** *valid_its_message* = *None* **then**
4: ... invalid message, discard it ...
5: **else**
6: ... message successfully verified, continue processing it ...
7: **end if**
8: **end function**

for real-world use). Additionally, it displays received messages, indicates the configured protocol, and provides other meta-information strictly for testing purposes (Figure 4.9).



(a) (b)

Figure 4.9: Android mobile application screenshots.

5

Experimental Evaluation

This chapter outlines the experimental evaluation process of the developed prototype. First, the experimental environment is described. Section 5.1 reports and analyses each node's local computation performance results. The latency measurements for each network are presented and discussed in Section 5.2. Finally, Section 5.3 shows and discusses the end-to-end (E2E) results achieved in each workflow.

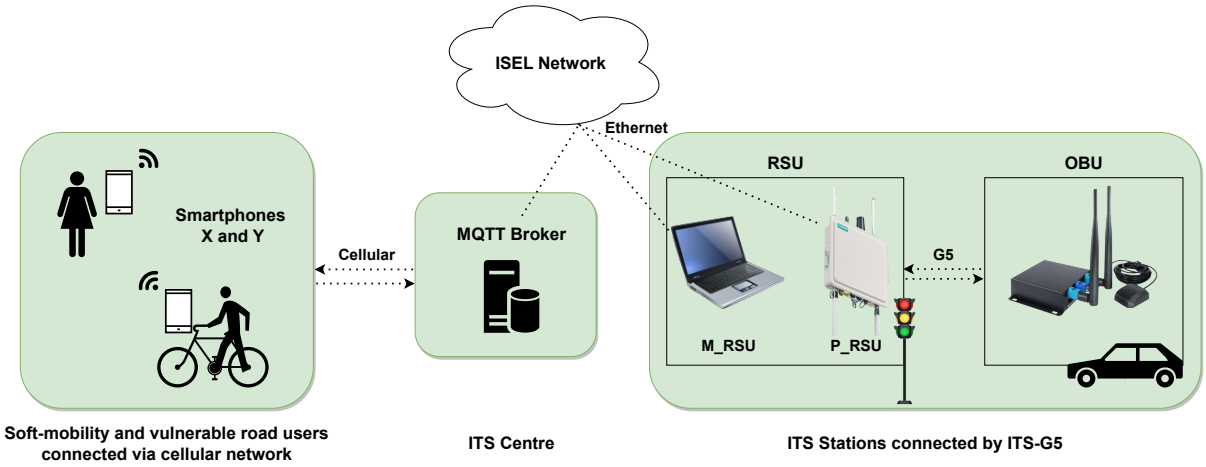


Figure 5.1: Testing environment representation.

The testing environment is depicted in Figure 5.1, and a concise description follows:

1. On the left is a representation of the cellular network serving soft-mobility and vulnerable road users. In the evaluation scenario, two smartphones are used (referred to as X and Y).

2. In the middle, the ITS Centre is hosting the MQTT broker connected to the ISEL network. This is the university's network where all the tests were conducted. Using a real network brings the evaluation closer to a real-world scenario.

3. On the right, and also connected to the ISEL's network via Ethernet, is the laptop (acting as the M_RSU) and the Siemens P_RSU. This one communicates with the OBU, both representing ITS Stations (e.g., a traffic light and a car) connected by ITS-G5.

Having described the evaluation setup, Table 5.1 reports each equipment computational environment.

Table 5.1: Characteristics of the computational environment where the prototype was tested.

Equipment	**Specifications**
Laptop	Windows 10 Processor Intel core i7-4710HQ CPU @ 2.50GHz 16 GB RAM
Unex OBU	Linux Yocto Dual 600MHz ARM Cortex-A7 32-bit CPU cores 128MB RAM
Siemens RSU	Linux Dual-Core ARM-Cortex A9 @800MHz 1 GB RAM
Smartphone X	Android 13 CPU Octa-core Max 2.96GHz 8GB RAM
Smartphone Y	Android 8.0 Qualcomm Snapdragon 425 2GB RAM

The developed work is assessed in different aspects:

Firstly, the **local computation** time of each execution environment's application (RSU, OBU and smartphone) is measured. In each app, the tests are carried out by varying the service to be executed (transmission/reception) and the security approach. These results are studied at various levels: analysing each protocol's performance individually; comparing some results with the ones shown in the respective proposal; how each security approach performs with real hardware and with a slightly different context of mobility; comparing both protocols' performance; and discuss their impact on computing performance.

Next, **communication latency** between nodes is also measured. Each network's (G5 and cellular) latency results are presented, discussed, and compared. The same three security approaches are used throughout these tests to analyse their impact on network latency.

Finally, **end-to-end times** are calculated. With these, the overhead introduced by each communication flow per network segment is compared. In addition, the overall impact of using the DLAPP and MFSPV protocols is discussed. Lastly, it is possible to draw insights into whether the E2E results suit the intended ITS use cases.

5.1 Computation Time

This section will perform a local computation performance comparison and analysis in each node using three security approaches — *No security*, *DLAPP* and *MFSPV*. This is important because it allowed to draw insights into how each security approach performs with real hardware in different execution environments and a slightly different context of mobility (with smartphones).

To achieve this, the evaluation procedure consists of two modes.

- **Total Computation Time** – Measures all the local computation time (CT) from the beginning of a transmission or reception processing until completion. This evaluation may be used with any security approach.
- **Security Computation Time** – Measures the CT for security protocol protection and deprotection. This mode must be used with a security protocol (DLAPP or MFSPV).

Protection involves applying a security protocol to a message and encapsulating it with the protocol bytes. Deprotection entails verifying a message according to the configured protocol. If the message is valid, the security bytes are then removed. Summarising, this evaluation objective is to measure the total and security computation time in each computational node without considering the network latency, only local computing.

The extraction of the necessary timestamps in each mode is illustrated in Figure 5.2. These are used in the computation time calculation, as given by the equations (5.1) and (5.2) for the transmission scenario. In the reception, the equations used are identical.

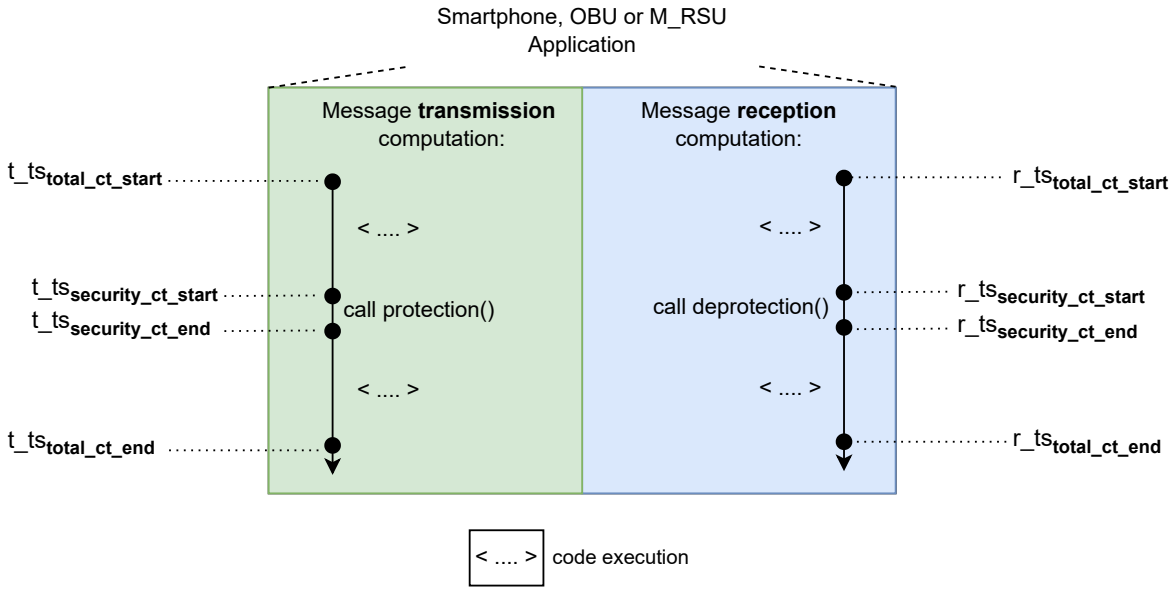


Figure 5.2: Total and security computation times extraction representation (for reception and transmission).

$$transmission\_total\_ct = t\_ts_{total\_ct\_end} - t\_ts_{total\_ct\_start} \tag{5.1}$$

$$transmission\_security\_ct = t\_ts_{total\_security\_start} - t\_ts_{total\_security\_end} \tag{5.2}$$

where
$t\_ts_{total\_ct\_start}$ represents the initial timestamp,
$t\_ts_{total\_ct\_end}$ is the final timestamp,
$t\_ts_{total\_security\_start}$ is the security CT initial timestamp, and
$t\_ts_{total\_security\_end}$ represents the security CT final timestamp.

Note that the processing results obtained by the RSU will be less emphasised as it does not fully represent the actual RSU execution environment. So, regarding the computation performance evaluation, OBU and smartphone results are more relevant.

From the set of all assessment combinations – *computing node*, *evaluation mode*, and *security approach* – more than 2000 computing times were extracted to make the obtained values more accurate. Firstly, the results will be presented and discussed for each protocol individually (security CT results). Then, these two will be compared, and their impact on the system's performance will be discussed (using the total CT results).

5.1.1 Performance Analysis: DLAPP

According to the DLAPP's proposal [26], its signature and verification simulation took 0.0167 ms (each operation). It is worth noting that the study only measured the time of the HMAC cryptographic operation, thus being a theoretical estimation. When implementing the protocol, there are more things to consider than just the cryptographic operation, e.g. validating the timestamp and checking if the HMAC matches the one received. It is essential to include all the computation associated with protecting and deprotecting a message to have a realistic measure of the computation time. The experimental performance results of the DLAPP protocol are shown in Table 5.2. In total, the security CT was measured at ~500 messages (one per second) that used the DLAPP protocol.

Table 5.2: Median security CT values [ms] utilising DLAPP in each node.

Node	Protection	Deprotection
Smartphone X	0.158	0.162
OBU	0.366	0.327
RSU	0.127	0.084

Based on this data, it can be concluded that, when using hardware (OBU and smartphones), the actual performance falls short (~90 – 95%) of what was initially projected in the protocol proposal. This can be attributed to the initial projections being based on simulations and not considering the entire protection and deprotection process.

Calculating the total operations per second as the DLAPP's proposal [26] does, we can discern that a smartphone can protect up to 6311 and deprotect 6165 messages per second. Conversely, the OBU has a lower capacity, protecting 2733 and deprotecting 3058 messages per second. This performance difference (~54%) may be attributed to the inherent limitations of OBUs as a resource-constrained device [5], as can be seen by its specifications in Table 5.1. In contrast, the smartphone, benefiting from highly-capable hardware, can deliver superior performance. Furthermore, the relation between protection and deprotection times exhibits similarity across all nodes. This is because the primary time-consuming factor is the HMAC, which is common in both operations.

These results can be seen in Figure 5.3. Assuming a similar high-vehicle-density scenario as the paper, i.e., 180 vehicles within communication range, sending a packet every 100 ms. This would result in 1800 messages needing to be verified per second. Based on the results (Figure 5.3), the DLAPP protocol is computationally light enough to manage such type of high node density scenario, both on OBU and smartphone.

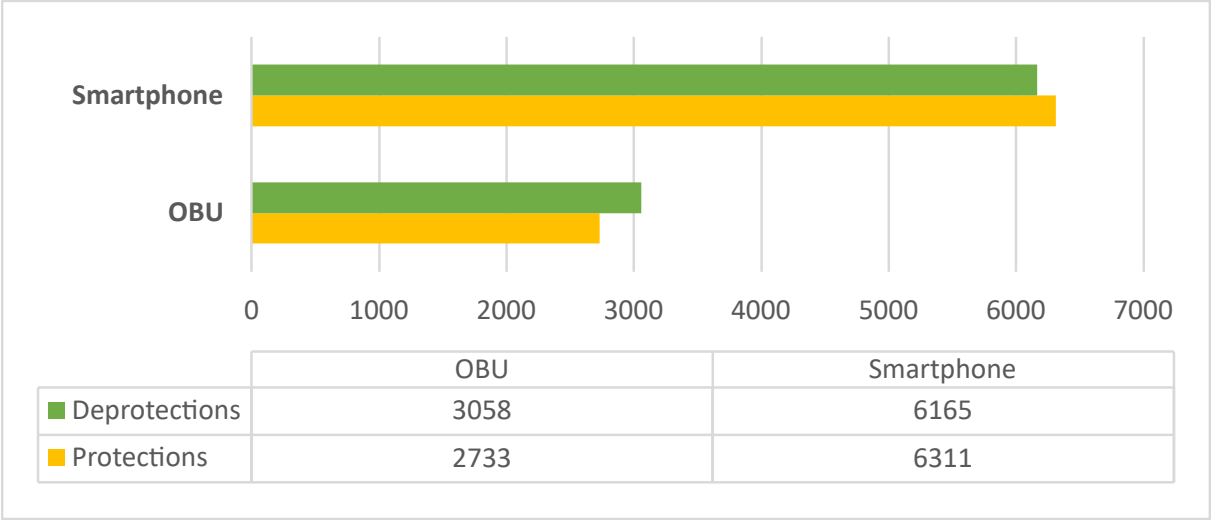


Figure 5.3: Total DLAPP operations (per second) in the developed applications for OBU and smartphone X.

5.1.2 Performance Analysis: MFSPV

Similar to the DLAPP proposal [26], MFSPV's authors [5] only consider the SHA-256 cryptographic operation to calculate the generation and verification times of a message. According to them, the MFSPV's protection takes 0.018 ms and the deprotection 0.006 ms. The MFSPV's performance results of this study are shown in Table 5.3. In total, the security CT was measured at ~500 messages (one per second) that used the MFSPV protocol.

Table 5.3: Median security CT results [ms] utilising MFSPV in each node.

Node	Protection	Deprotection
Smartphone X	0.136	0.107
OBU	0.167	0.153
RSU	0.138	0.064

By analysing the computation measurement results and as concluded in the DLAPP's performance analysis, the estimations provided in the proposal protocol [5] are higher than those observed in a real scenario for the same reasons. Furthermore, the smartphone X can protect up to 9343 messages per second and deprotect 7327 messages. The OBU, as before, presents a lower performance than the smartphone. Protecting 5981 and deprotecting 6519 messages per second. Unlike the DLAPP protocol, the protection and deprotection operations in MFSPV are not so similar (Section 3.2). Deprotection exhibits lower computation time (~8% to 53%) on all nodes. This difference

exists because the protocol performs more hash operations in protection than in deprotection.

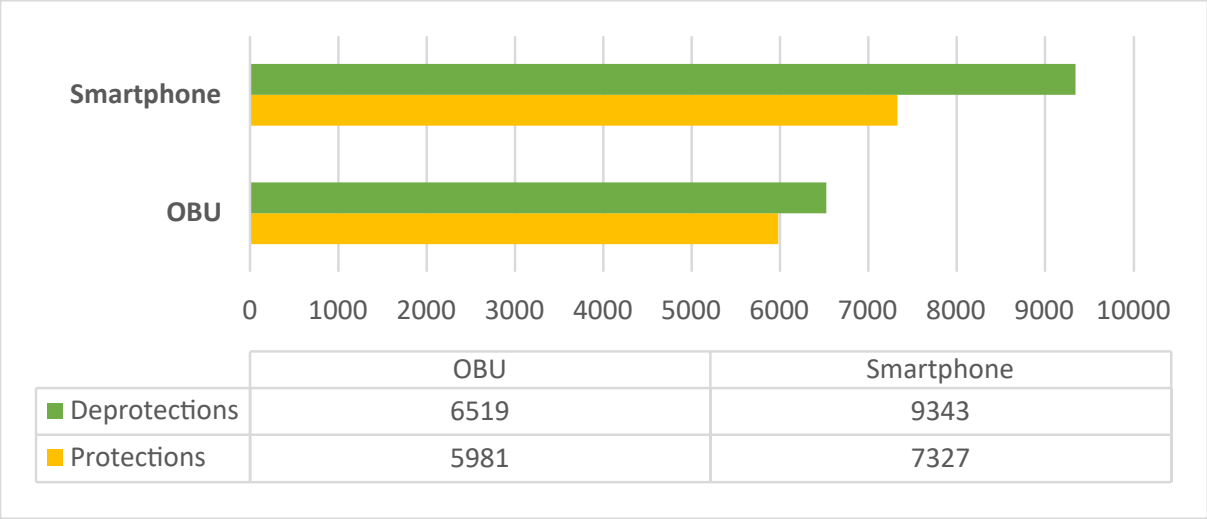
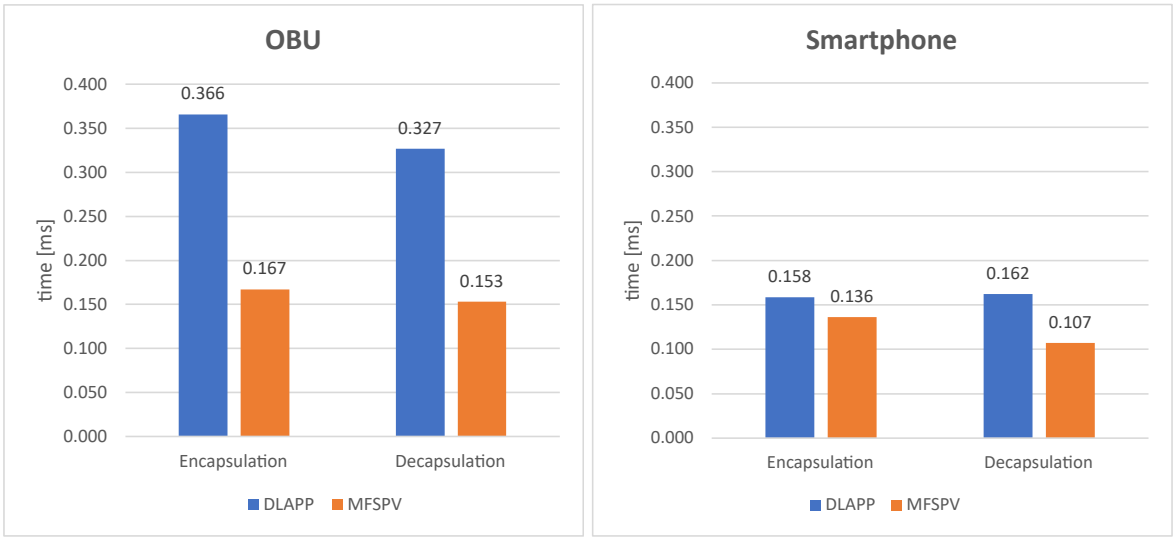


Figure 5.4: Total MFSPV operations (per second) in the developed applications for OBU and smartphone X.

These results can be seen in Figure 5.4. Assuming the previous high-vehicle-density scenario, i.e., 1800 messages needing to be verified per second. The MFSPV also showed to be computationally light enough to manage this high node density scenario on OBUs and smartphones.

5.1.3 Performance Analysis Comparison

After analysing the performance of each protocol individually, they are compared. Figure 5.5 reports the median security CT results for DLAPP and MFSPV in OBU and smartphone X. MFSPV outperforms DLAPP in both nodes, being more evident on the OBU. Analysing this difference from the perspective of operations per second, MFSPV allows protection of 1016 and 3248 more messages on the OBU and smartphone, respectively. Plus 3461 and 3178 message deprotections. This translates into a performance increase between ~**16% to 113%**, depending on the node and type of operation. MFSPV achieves this performance advantage due to the exclusive use of hashes, which are computationally lighter than the HMAC operation. Despite this, as both protocols were designed to be lightweight, the magnitude of the times involved is minimal, in the order of **tenths of milliseconds**.



(a) (b)

Figure 5.5: Median security CT results [ms] for DLAPP and MFSPV in (a) OBU and (b) Smartphone X (in a total of ∼400 measurements).

5.1.4 Security Impact on Performance

The performance results obtained in total CT mode are reported in Figure 5.2. This assessment encompasses the measurement of both transmission and reception computation times across all three security approaches. As a result, it provides insights into the impact of utilising the MFSPV and DLAPP on the application's performance.

Table 5.4: Total CT results [ms] measured on OBU, using different security approaches (in a total of ∼300 measurements).

Security approach	Transmission	Reception
No security	1.188	0.266
DLAPP	1.605	0.675
MFSPV	1.439	0.513

Table 5.5: Total CT results [ms] measured on a smartphone X, using different security approaches (in a total of ∼300 measurements).

Security approach	Transmission	Reception
No security	0.840	0.137
DLAPP	1.213	0.340
MFSPV	0.872	0.324

Table 5.4 and 5.5 present the experimental results from both the OBU and the smartphone X. The obtained results with and without security do not vary substantially. Figure 5.6 presents the total CT results but expresses each time as a relative ratio of the reference task (*baseline*), which is the non-use of security, making it easier to assess the security impact in performance.

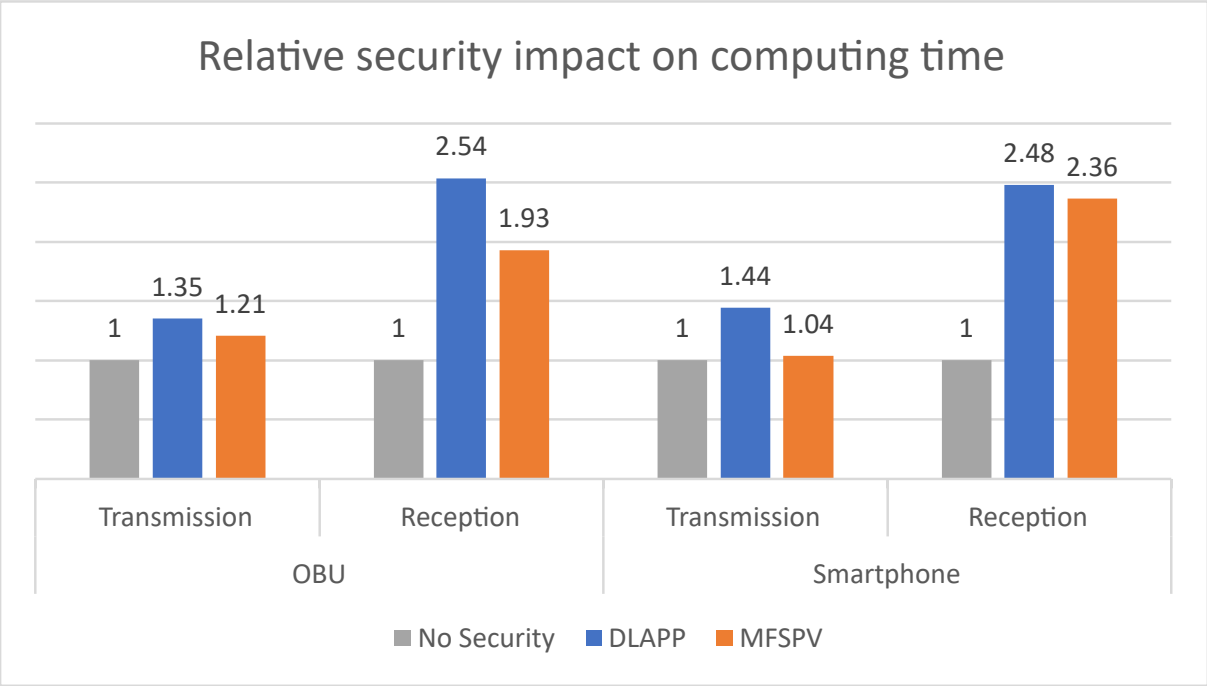


Figure 5.6: Performance impact of security protocols (DLAPP and MFSPV). The computation times are relative ratios to the respective 'No Security' task.

In transmissions, DLAPP increases the computational delay by **35%** on the OBU and **44%** on the smartphone (when compared to 'no security' scenarios). In comparison, MFSPV increases it by **21%** on the OBU and **4%** on the smartphone. DLAPP has a more significant impact on computing time than MFSPV, as expected according to previous analyses. The same applies to reception times, but greater relative increases can be seen in this case. This difference is understandable since reception times are lower than transmission times (the order of magnitude is smaller), even in the absence of security measures (as reported in Table 5.4 and 5.5). Consequently, even minor increases in reception times result in more pronounced relative changes. Nevertheless, the impact of protocols on reception is still low, increasing it in tenths of a millisecond.

5.2 Network Latency

Latency is an important performance indicator in communication. For this reason, this section assesses the latency of the developed hybrid network. In particular, measuring the latency in communications between nodes. Each network segment's latency results are presented, discussed, and compared. Moreover, three security approaches are used throughout the tests to analyse their impact on network latency.

To perform these tests, there were two options for measuring the latency. The first was to achieve synchronisation between the clocks, but guaranteeing this would be complex and more time-consuming. The other hypothesis, and the one applied to all the latency tests, was via the Round-Trip Time (RTT), which generally is the time it takes a system to get a response after initiating a request to another computational node. In this case, messages will be sent between the nodes whose latency needs to be measured. The receiving node's application was configured to echo the message as soon as they received it. Then, the transmitter divides the RTT by two, approximating the latency in this communication. By carrying out this process several times, we converge on an increasingly accurate approximation of what an average latency time would be when transmitting from node A to node B. Figure 5.7 describes the methodology adopted to obtain the timestamps needed for RTT calculation. In the exemplified scenario, the smartphone Y is the receiver, echoing the received message. The smartphone X calculates the RTT, and through this, the latency.

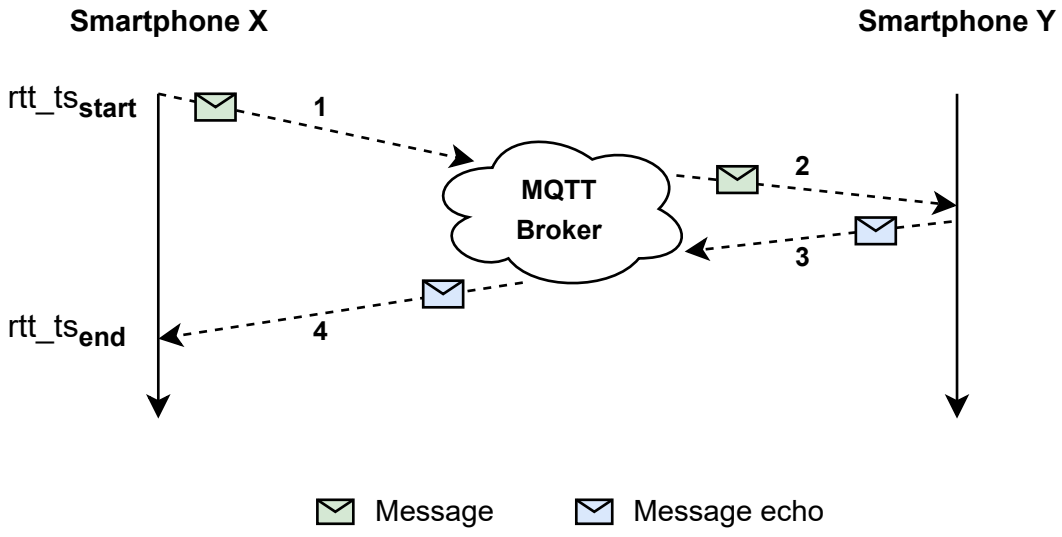


Figure 5.7: Methodology for calculating the RTT in communications involving the cellular network.

This methodology was used to calculate the communications latencies involving the cellular network. The timestamps (rtt_ts_{start} and rtt_ts_{end}) are used to calculate the transmission latency time, as given by equation (5.3).

$$latency = \frac{rtt\_ts_{end} - rtt\_ts_{start}}{2} \tag{5.3}$$

where
rtt_ts_{start} is the RTT's initial timestamp,
rtt_ts_{end} is the RTT's final timestamp.

The methodology illustrated in Figure 5.7 was used in the calculation of latency of the following communication flows:
(i) *Smartphone X* → *Smartphone Y*,
(ii) *Smartphone X* → *RSU*,
(iii) *M_RSU* → *Smartphone X*.

Therefore, latency was calculated for all communications involving the cellular network segment. A similar but slightly different strategy was adopted to calculate communications latency in G5 (between the P_RSU and OBU). Figure 5.8 illustrates the methodology used while the equation (5.4) shows the latency calculation.

$$g5\_latency = \frac{rtt\_ts_{end} - rtt\_ts_{start}}{2} - wss\_latency \tag{5.4}$$

where
rtt_ts_{start} is the RTT's initial timestamp,
rtt_ts_{end} is the RTT's final timestamp,
avg_wss_latency represents an average of WSS communication latency between the M_RSU and the P_RSU. This corresponds to the transmission time of flow 1 or 4 in Figure 5.8. It is calculated using the same strategy as in Figure 5.7.

This strategy was chosen due to time constraints and the difficulty of programming the P_RSU. So, the M_RSU was used to help extract G5 latency measurements. By eliminating the latency linked to WSS communication, the G5 transmission latency is calculated. Alternatively, the G5 latency could have been measured without using M_RSU by programming the OBU to also act as a client of the P_RSU's XFER interface. However, this would require the OBU to instruct the P_RSU to send echoes, essentially making the OBU a dedicated XFER client for this purpose only. This approach would complicate the process, particularly in C, and consume a significant amount of time. Therefore, the presented procedure was chosen.

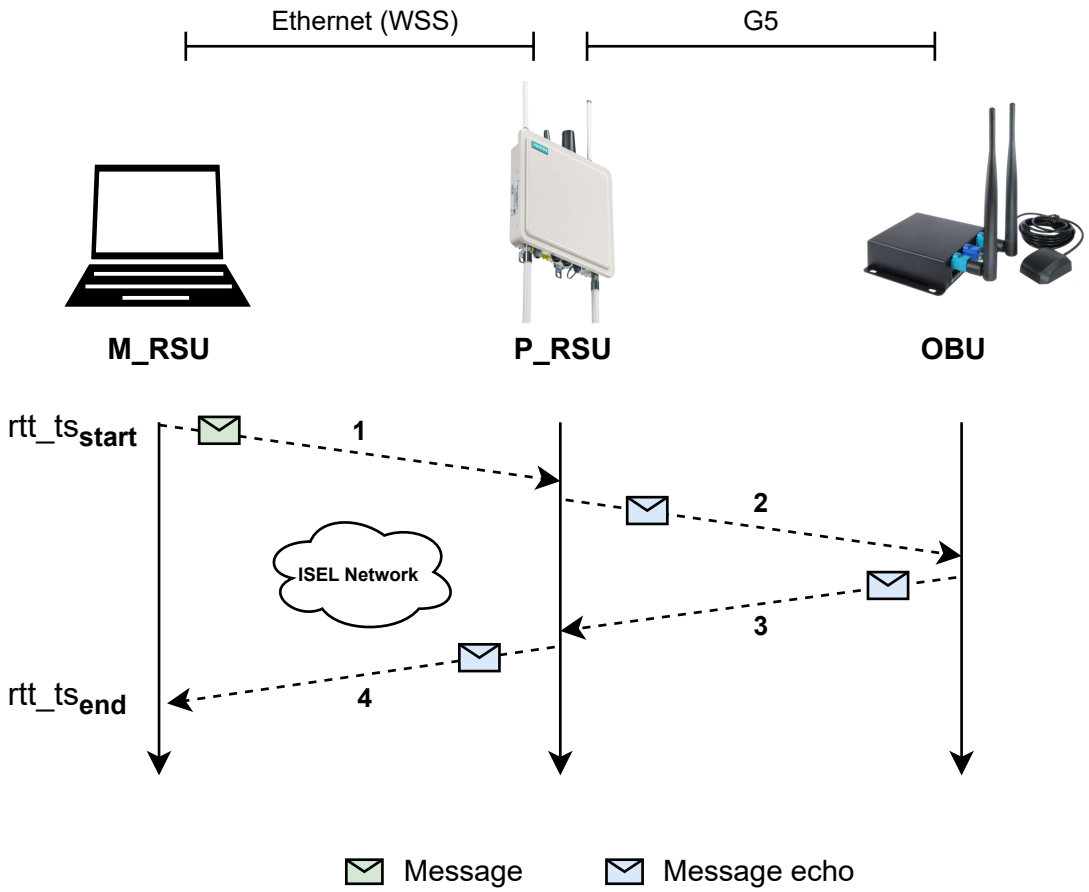


Figure 5.8: Methodology for calculating latency in G5 communications.

In all conducted tests, ~2000 latency samples were extracted, with a message transmitted once per second. Due to the occurrence of outliers, as illustrated in Figure 5.9, the median values will be used.

5.2.1 Latency Measurements Analysis: Cellular Network

The latency values involving the cellular network are shown in Table 5.6. Some patterns can be observed by analysing the measurements obtained for each communication flow.

The *M_RSU → Smartphone X* flow shows better results than the flows in which smartphones are the source. This may happen because, as shown in the testing environment (Figure 5.1), the M_RSU is in a privileged position as it is connected via ethernet to the ISEL network. The MQTT broker at the ITS Centre is also connected to this network. So, M_RSU publishing the message achieves lower latency than when smartphones publish it and reach the MQTT server via the cellular network.

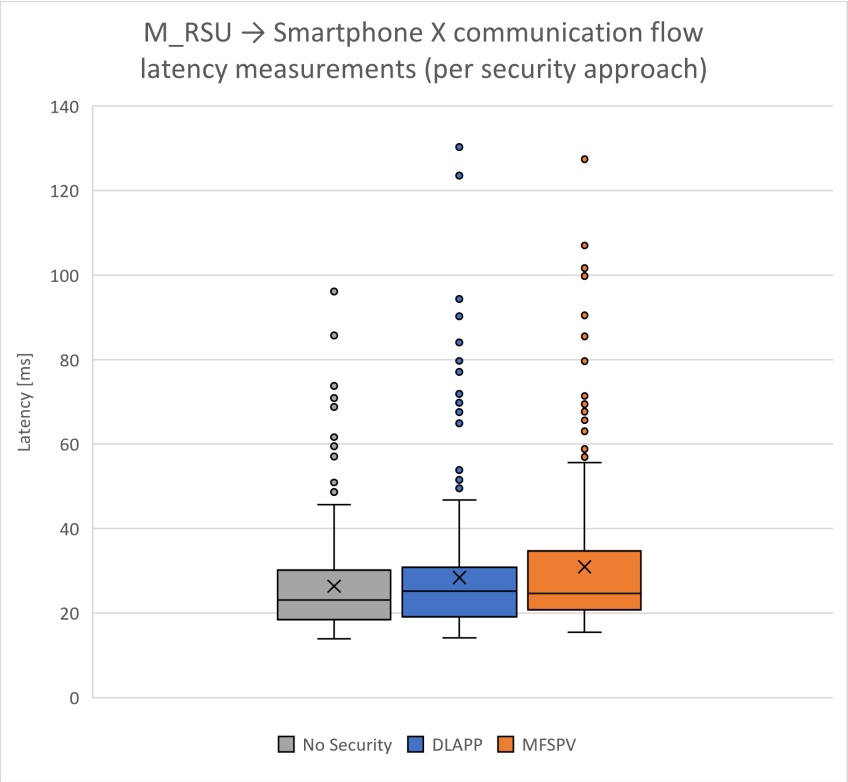


Figure 5.9: Box plot of latency measurements in *M_RSU → Smartphone X* comunication flow.

Table 5.6: Latency measurements [ms] of communications that involve the cellular network, using different security approaches.

Communication Flow			No security	DLAPP	MFSPV
M_RSU	→	Smartphone X	23.08	25.23	24.66
Smartphone X	→	M_RSU	29.74	31.90	32.32
Smartphone X	→	Smartphone Y	31.78	33.22	33.97

In addition, there are also higher latencies in communications between smartphones, which is justified by the fact that both are on the mobile network, which contributes to higher latencies.

Upon individual analysis of each communication flow, it becomes evident that the omission of a security protocol results in the most favourable latency measurements. This can be attributed to the fact that using security measures introduces an additional message payload overhead. Among the results of each protocol, DLAPP, with four fewer bytes of overhead than MFSPV, exhibits better performance on two occasions when compared to MFSPV. On only one occasion, it demonstrates slightly greater latency. These results indicate that the difference of 4 bytes does not significantly influence the use of one protocol over the other.

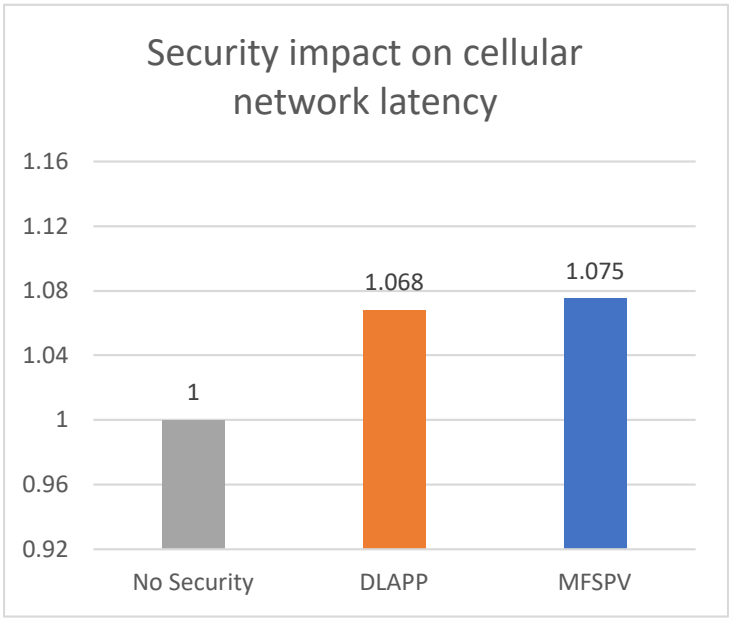


Figure 5.10: Latency performance impact of security protocols (DLAPP and MFSPV) expressed in relative ratios to the 'No Security'.

Figure 5.10 presents the average latency measurements (as detailed in Table 5.6) for each security approach, but expressing each delay as a relative ratio of the reference (*baseline*), which corresponds to the non-use of security. This representation simplifies the assessment of how security affects latency in cellular communications. It provides insights into the impact of utilising the DLAPP and MFSPV protocols on the cellular networks's overall performance. Specifically, when compared to scenarios where no security is used, DLAPP increases cellular network latency by 6.8%, while MFSPV increases it by 7.5%.

5.2.2 Latency Measurements Analysis: G5 Network

The latency measurements of G5 communications (RSU and OBU) are present in Table 5.7. DLAPP increases the latency by 6% and the MFSPV by 10%. The impact of the protocols on the G5 network is not very noticeable.

Table 5.7: Latency measurements [ms] of G5 communications between RSU and OBU, using different security approaches.

No security	DLAPP	MFSPV
10.196	10.792	11.251

5.2.3 Latency Measurements Analysis Comparison

The comparison of latency measurements of the cellular network and the G5 is illustrated in Figure 5.11.

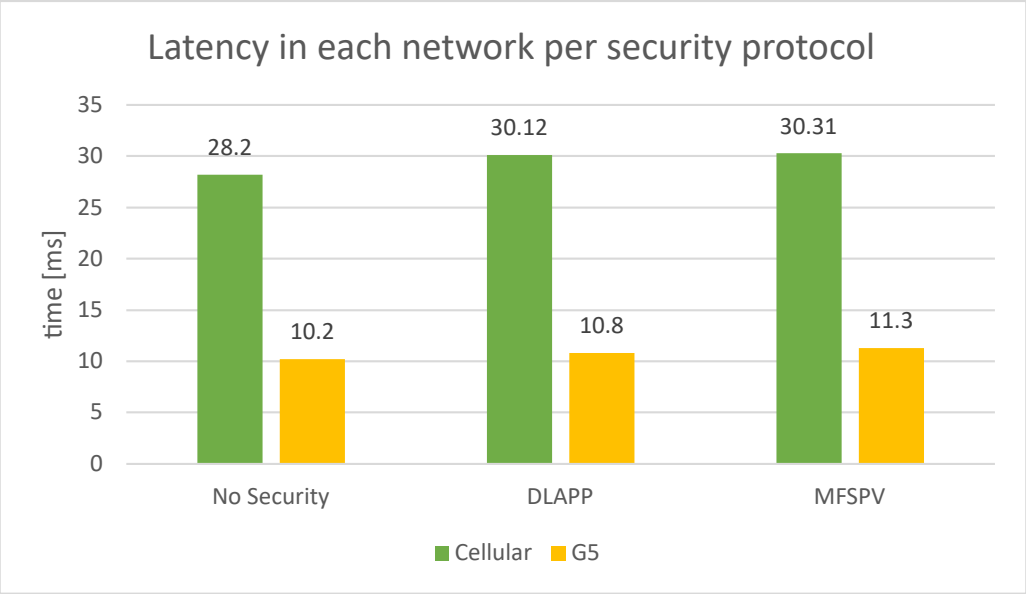


Figure 5.11: Cellular and G5 latency measurements comparison, in each security approach. The cellular latencies are the average of the ones reported in Table 5.6.

The G5 network, on average, has **63.6%** lower latency compared to the cellular network [24]. As shown in Figure 5.11, the G5 network achieves a considerably shorter transmission time across all security approaches. This difference is justified by the transmission in G5 being direct (ad-hoc), without needing a broker, thus being more efficient.

The impact of security protocols on network latency is, on average, **7.1%** on the cellular network and **8%** on the G5 network. Comparing both protocols, DLAPP is slightly more efficient. However, the measurements obtained in both protocols are similar, which is justified by the fact that there is only a 4-byte difference in the payload.

5.3 End-to-End Assessment

E2E is an important indicator when developing a system, as it is crucial to know how long the system takes to perform a job, from the start of a workflow to the end. Therefore, the E2E time will be calculated for each communication flow of the developed prototype.

All the measurements collected in the computation time and latency section will be used to obtain approximations of the E2E, i.e., it will be calculated according to the existing processing time and latency in each communication flow. It brings together the median values obtained in computing and networking latency measurements. In total, approximately 4000 measurements were collected across all the conducted assessments. The calculated E2E times for each combination between nodes are reported in Table 5.8.

Table 5.8: E2E times [ms] for the various flows of the prototype, with different security approaches. Communication flows are divided according to the network segment they use.

Network Segment	Communication Flow			No Security	DLAPP	MFSPV
G5	OBU	→	RSU	**11.63**	13.24	13.55
	RSU	→	OBU	12.24	13.61	13.97
Cellular	RSU	→	Smartphone X	**24.59**	27.71	27.19
	Smartphone X	→	RSU	31.72	34.99	34.98
	Smartphone X	→	Smartphone Y	32.76	34.77	35.16
Hybrid	Smartphone X	→	OBU	42.18	46.46	46.75
	OBU	→	Smartphone X	**34.94**	38.81	38.53

The results will be analysed from two perspectives: network segment and security approach. Following this analysis, the applicability of the developed proof-of-concept is briefly discussed based on the results obtained.

5.3.1 Analysis per network segment

The most time-consuming E2E communication flows are seen in the hybrid network communication flows, where messages are generated in the OBU and propagated until the smartphone and vice versa. In particular, the greatest median E2E time is observed in the flow *Smartphone → OBU* using the MFSPV protocol, **46.75 ms**. Based on the analysis conducted so far, this outcome was anticipated. The hybrid network shows an average E2E time of **41.26 ms**.

Conversely, the E2E times achieved by G5 exclusive communication flows are the lowest, namely in the *OBU → RSU* flow, the E2E time is just **11.63 ms**, without the use of security. The G5 network shows an average E2E time of **12.97 ms**.

From these E2E results (Table 5.8), it can be concluded that hybrid communication flows impose an extra **28.29 ms** of E2E time, which translates into an increase of **218%** compared to G5 only communication flows. Figure 5.12 illustrates the average E2E results obtained in each network segment.

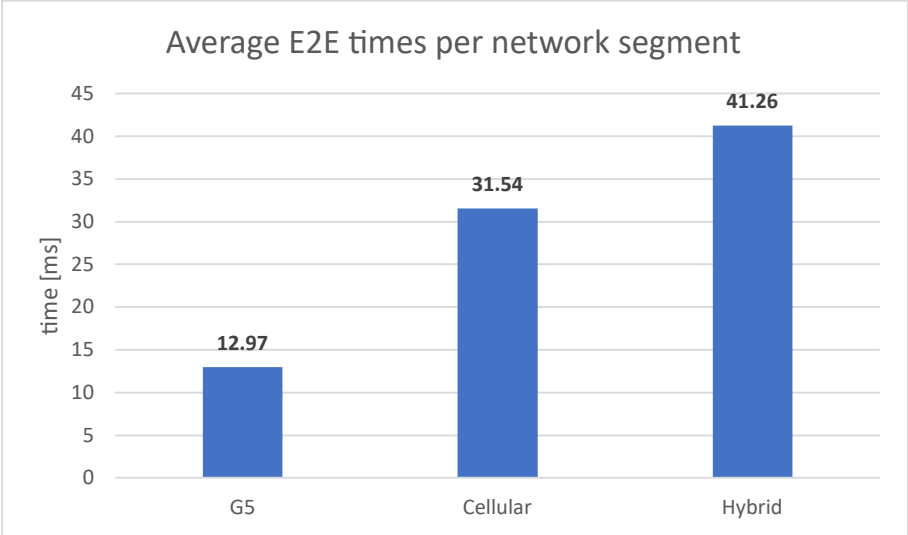


Figure 5.12: Average E2E times of communication flows associated with each network segment.

5.3.2 Analysis per security approach

Figure 5.13 summarises the end-to-end (E2E) results, focusing on the analysis of the protocols' impact.

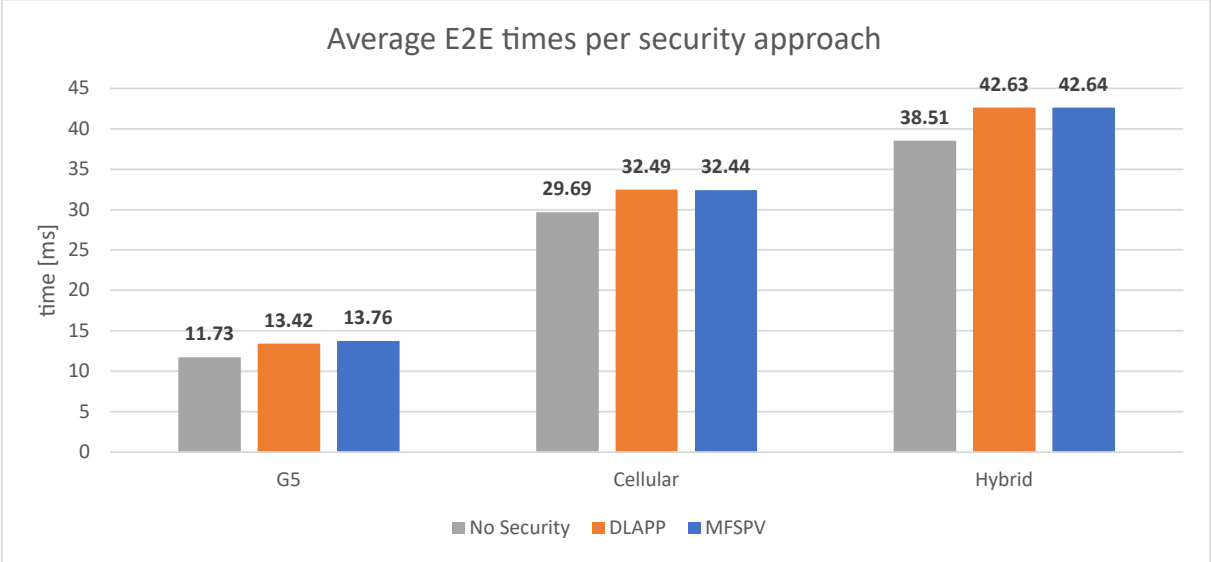


Figure 5.13: Average E2E times of communication per security approach. The study is also divided according to the network segment.

As observed in the experiments conducted so far, it is apparent that security protocols have a relatively low impact compared, for instance, to the extension to hybrid networks. In E2E results, the same thing happens. For example, the additional E2E delay imposed by the protocols in the hybrid segment workflows is approximately **11%** in

both protocols. MFSPV proves to be more efficient in local processing, and DLAPP achieves slightly better latencies. Nonetheless, looking at the big picture (Figure 5.13), both impose a very similar additional E2E time.

5.3.3 Applicability Considerations

On a final note, as referred by Castañeda *et al.* [8], various use cases have defined specific requirements for maximum latencies. The most stringent among them are emergency services, such as pre-crash warnings, which require a 50 ms maximum latency. In comparison, most other use cases require a maximum latency of 100 ms. With this understanding and examining the obtained results, significant conclusions can be drawn.

The median E2E values, as shown in Table 5.8, do not surpass ∼47 ms. This observation indicates that the developed approach aligns well with the requirements of many use cases. However, focusing solely on median values does not provide a complete picture. Therefore, the highest E2E time was also calculated, representing the worst-case scenario regarding latency and computational measurements.

The highest E2E time was encountered in the communication flow *Smartphone X* → *OBU* using the DLAPP protocol, reaching an E2E time of approximately 190 ms. Nonetheless, it's important to note that these values are considered outliers. Outliers were identified using the Interquartile Range (IQR) method, specifically, values above $Q3 + 1.5\ x\ IQR$ or below $Q1 - 1.5\ x\ IQR$, as illustrated in the box plot in Figure 5.9.

Finally, the same analysis was repeated, i.e., considering maximum values but now excluding outliers. In this case, the maximum E2E time observed was 86 milliseconds in the *Smartphone X* → *OBU* communication flow using the MFSPV protocol. This means that, when excluding outliers and assuming the worst-case scenario, the results obtained in this study still remain 14% below the maximum latency requirements for the majority of use cases.

6

Conclusions

This chapter concludes this document by offering an overview of the main considerations and findings that emerged during the thesis development. It also discusses the challenges encountered and provides recommendations for future work.

6.1 Main Considerations and Findings

The communication of V2X systems is inherently open, which leads to vulnerabilities that attackers exploit. This represents a threat to all road users, as security failures could lead to privacy violations or even fatalities. Moreover, a high fatality rate is correlated with soft mobility road users. So, while developing C-ITS-based systems, we must broaden our perspective beyond just vehicles, considering the needs of soft mobility transportation users.

In this study, an approach was proposed, and experiments were performed to explore the effectiveness of a system that employs lightweight security protocols in a C-ITS framework, operating within a hybrid network that integrates connected ITS stations via G5 and soft-mobility users connected through their smartphones via cellular networks.

To accomplish the main objectives, two lightweight security protocols — DLAPP and MFSPV — were implemented. This study employed real equipment, including OBUs and RSUs, and extended the protocols to smartphone applications. A hybrid environment was developed to allow soft mobility users and ITS stations to communicate.

On the one hand, soft-mobility users, via a mobile application and the cellular network, publish and receive messages via an MQTT broker, allegorically hosted in an ITS Centre. On the other hand, ITS Stations (OBU and RSU) were connected through ITS-G5. The integration of these two networks occurred through the ITS-Centre's broker. Besides managing smartphone messages, it utilised the RSU as a connection point for the G5 network. This architecture enabled messages between smartphones and the RSU/OBU, ensuring bidirectional communication as long as all nodes were configured with the same security policy.

The **computing** experimental results emphasise the importance of implementing security protocols in hardware instead of relying solely on simulation. Regarding the protocols' performance, MFSPV outperformed DLAPP, exhibiting a 16% to 113% efficiency improvement, depending on the specific computational node and the operation (protection or verification).
When considering their impact on overall computation time, for instance, in transmissions, DLAPP increased, on average, the computation delay by 39.5% in the OBU and smartphone. In comparison, MFSPV increases it by 12.5%.
Despite this, as both protocols were designed to be lightweight, the magnitude of the times involved is very small, in the order of tenths of milliseconds. Therefore, although MFSPV has proven more efficient, this difference is not noticeable in the E2E times. Nonetheless, both presented a relatively low impact on local computing time compared to situations where security was not used.

As for **network latency**, experimental measurements have shown that DLAPP is slightly more efficient as it increases G5 and cellular network latency by 6.5%, whereas MFSPV results in 8.3% degradation. Moreover, the G5 network, on average, has 63.6% lower latency times compared to the cellular network across all security approaches tested.

Regarding **end-to-end** assessment, the most time-consuming E2E communication flows are seen on the hybrid network communication flows, which is expected since messages travel via G5 and cellular networks. In particular, the highest E2E time was 46.75 ms. Conversely, the E2E times achieved by G5 exclusive communication flows are the lowest, 11.63ms. It was also observed that the extension for hybrid communication imposes, on average, an extra 28.29 ms of E2E time. Furthermore, the additional E2E delay imposed by using security is approximately 11% in both protocols.

Finally, there are some final remarks. DLAPP and MFSPV protocols imposed a similar additional E2E time. Therefore, choosing one over the other in terms of efficiency is not straightforward. The choice should depend on the specific priorities of the application. For example, if the application's priority is to put less stress on the equipment,

MFSPV may be the most suitable option. On the other hand, if reducing network latencies as much as possible is essential, DLAPP may be a viable option. Furthermore, operating a C-ITS system within a hybrid network raises challenges due to the increase in latency imposed by cellular networks. Despite this, the mobile application achieved good performance levels.

The suitability of the presented approach should be contingent on the specific nature of the ITS applications it will incorporate. That is, different ITS use cases have distinct maximum latency demands, with the most stringent ones, such as emergency services, requiring a 50 ms latency and most others allowing up to 100 ms. This study's median E2E values do not surpass ~47 ms, aligning well with the requirements of most use cases, especially those with 100 ms as requirements. In a worst-case analysis, E2E time reached around 190 ms. However, it represents a very unusual scenario. Therefore, outliers were isolated using the IQR method. When outliers were excluded, the worst-case E2E latency observed remained at 86 ms, 14% below the maximum latency requirements for most use cases. It can be concluded that this proof-of-concept's results align well with the requirements for most use cases. For instance, consider applications that rely on timely and punctual DENM messages to alert users about road events, such as lane closures. In such cases, this could be an interesting use case. The system would gain advantages like enhanced information and awareness for soft-mobility users, all while upholding important security attributes like privacy and integrity.

6.2 Future Work

Upon conducting a self-assessment of the work that has been developed and the results achieved, some opportunities for future enhancements and refinements were identified. These include:

- Acquire greater proficiency in interacting with ITS equipment, as it has been noted that this is a non-trivial task.
- Conduct experiments involving OBUs and RSUs from various manufacturers. This would enable a comparison of results and the validation of certain conclusions for different equipment, thus reinforcing its applicability.
- Develop the CA service.
- Perform the evaluation experiments under conditions of greater stress/overload, encompassing both computational and network aspects. This approach would

allow to analyse this proof-of-concept system's response to real-world extreme scenarios.

References

[1] Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer, "Stride-based threat modeling for cyber-physical systems", in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 2017, pages 1–6. DOI: 10.1109/ISGTEurope.2017.8260283.

[2] Farah Haidar, Arnaud Kaiser, and Brigitte Lonc, "On the performance evaluation of vehicular pki protocol for v2x communications security", in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pages 1–5. DOI: 10.1109/VTCFall.2017.8288286.

[3] Hamidreza Bagheri, Md Noor-A-Rahim, Zilong Liu, Haeyoung Lee, Dirk Pesch, Klaus Moessner, and Pei Xiao, "5g nr-v2x: Toward connected and cooperative autonomous driving", *IEEE Communications Standards Magazine*, vol. 5, no. 1, pages 48–54, 2021. DOI: 10.1109/MCOMSTD.001.2000069.

[4] European Comission Adina-Ioana Vălean, "Eu road safety policy framework 2021 - 2030, next steps towards 'vision zero'", Tech. Rep., Feb. 2020. [Online]. Available: https://visaozero2030.pt/wp-content/uploads/EU_Road_Safety_Policy_Framework_2021-2030_Next_Steps_towards_Vision_Zero.pdf.

[5] Saad Ali Alfadhli, Songfeng Lu, Kai Chen, and Meriem Sebai, "Mfspv: A multi-factor secured and lightweight privacy-preserving authentication scheme for vanets", *IEEE Access*, vol. 8, pages 142 858–142 874, 2020. DOI: 10.1109/ACCESS.2020.3014038.

[6] Ariane Debysern, "Road safety in the eu - european parliamentary research service", Tech. Rep., Feb. 2019. [Online]. Available: https://www.europarl.europa.eu/RegData/etudes/BRIE/2019/635540/EPRS_BRI(2019)635540_EN.pdf.

[7] Jan-Felix van Dam, Norbert Bißmeyer, Christian Zimmermann, and Kurt Eckert, "Security in hybrid vehicular communication based on its g5, lte-v, and mobile edge computing", in *Fahrerassistenzsysteme 2018*, Torsten Bertram, Ed., Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pages 80–91, ISBN: 978-3-658-23751-6. DOI: https://doi.org/10.1007/978-3-658-23751-6_8.

[8] Oscar Castañeda, Janie Baños, Antonio J Garrido, Carlos Cárdenas, Carlos Mendes, Antonio Serrador, Nuno Cota, Nuno Datia, and Nuno Cruz, "Latency assessment for cam services over 5g", 2021. [Online]. Available: https://5g-mobix.com/assets/files/5G-MOBIX_Latency-Assessment-for-CAM-services-over-5G.pdf.

[9] European Comission, "Final report of the single platform for open road testing and pre-deployment of cooperative, connected and automated and autonomous mobility platform (ccam platform)", Tech. Rep., Jul. 2021. [Online]. Available: https://transport.ec.europa.eu/system/files/2021-11/Final%20Report-CCAM%20Platform.pdf.

[10] ETSI, "ITS Security - Trust and Privacy Management", Technical Specification 102 941 v1.4.1, 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/102900_102999/102941/01.04.01_60/ts_102941v010401p.pdf.

[11] ETSI, "ITS Security - Security Header and Certificate formats", Technical Specification 103 097 v2.1.1, 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/103000_103099/103097/02.01.01_60/ts_103097v020101p.pdf.

[12] ETSI, "ITS Basic Set of Applications - Facilities layer protocols and communication requirements for infrastructure services", Technical Specification 103 301 v1.3.1, 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/103300_103399/103301/01.03.01_60/ts_103301v010301p.pdf.

[13] ETSI, "ITS Security - Communications Architecture", European Standard 302 665 v1.1.1, 2010. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/302665/01.01.01_60/en_302665v010101p.pdf.

[14] ETSI, "ITS Security - Threat, Vulnerability and Risk Analysis (TVRA)", Technical Report 102893 v1.2.1, 2017. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/102800_102899/102893/01.02.01_60/tr_102893v010201p.pdf.

[15] ETSI, "ITS Security - ITS communications Security Architecture and Security Management", Technical Specification 102 940 v2.1.1, 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/102900_102999/102940/02.01.01_60/ts_102940v020101p.pdf.

[16] ETSI, "ITS Security - Confidentiality services", Technical Specification 102943 v1.1.1, 2012. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/102900_102999/102943/01.01.01_60/ts_102943v010101p.pdf.

[17] ETSI, "ITS Security - Confidentiality services", Technical Specification 102943 v2.0.0, 2022. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/102900_102999/102943/02.00.00_60/ts_102943v020000p.pdf.

[18] ETSI, "ITS Vehicular Communications Basic Set of Applications Part 2: Specification of Cooperative Awareness Basic Service", European Standard 302 637-2 V1.4.1, 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf.

[19] ETSI, "ITS Vehicular Communications Basic Set of Applications Part 3: Specification of Decentralised Environmental Notification Basic Service", European Standard 302 637-3 V1.3.1, 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.03.01_60/en_30263703v010301p.pdf.

[20] ETSI, *Etsi - welcome to the world of standards*, https://www.etsi.org/, (Accessed on 2023-09-25).

[21] European Commission, "Certificate policy for deployment and operation of european cooperative intelligent transport systems (c-its)", Tech. Rep., Jun. 2018. [Online]. Available: https://transport.ec.europa.eu/system/files/2018-05/c-its_certificate_policy-v1.1.pdf.

[22] Andreas Festag, "Cooperative intelligent transport systems standards in europe", *IEEE Communications Magazine*, vol. 52, no. 12, pages 166–172, 2014. DOI: 10.1109/MCOM.2014.6979970.

[23] Dries Naudts, Vasilis Maglogiannis, Seilendria Hadiwardoyo, Daniel van den Akker, Simon Vanneste, Siegfried Mercelis, Peter Hellinckx, Bart Lannoo, Johann Marquez-Barja, and Ingrid Moerman, "Vehicular communication management framework: A flexible hybrid connectivity platform for ccam services", *Future Internet*, vol. 13, no. 3, 2021, ISSN: 1999-5903. DOI: 10.3390/fi13030081. [Online]. Available: https://www.mdpi.com/1999-5903/13/3/81.

[24] Mafalda Gonçalves, Nuno Datia, and António Serrador, "A safety perspective for soft mobility in the its ecosystem", in *Inforum 22 — Atas do 13o Simpósio de Informática*, 2022, pages 330–341. [Online]. Available: https://inforum.org.pt/2022/sites/default/files/2022-09/Actas_INForum.pdf.

[25] Farah Haidar, Arnaud Kaiser, and Brigitte Lonc, "On the performance evaluation of vehicular pki protocol for v2x communications security", in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pages 1–5. DOI: 10.1109/VTCFall.2017.8288286.

[26] Shimaa A. Abdel Hakeem, Mohamed A. Abd El-Gawad, and HyungWon Kim, "A decentralized lightweight authentication and privacy protocol for vehicular networks", *IEEE Access*, vol. 7, pages 119 689–119 705, 2019. DOI: 10.1109/ACCESS.2019.2937182.

[27] Monowar Hasan, Sibin Mohan, Takayuki Shimizu, and Hongsheng Lu, "Securing vehicle-to-everything (v2x) communication platforms", *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pages 693–713, 2020. DOI: 10.1109/TIV.2020.2987430.

[28] Matthias Hiller, "Key derivation with physical unclonable functions", Ph.D. dissertation, Technische Universität München, 2016.

[29] International Transport Forum, "New directions for data-driven transport safety corporate partnership board report", Tech. Rep., May 2019. [Online]. Available: https://www.itf-oecd.org/sites/default/files/docs/new-directions-data-driven-transport-safety_0.pdf.

[30] Adnan Mahmood, Wei Emma Zhang, and Quan Z. Sheng, "Software-defined heterogeneous vehicular networking: The architectural design and open challenges", *Future Internet*, vol. 11, no. 3, 2019, ISSN: 1999-5903. DOI: 10.3390/fi11030070. [Online]. Available: https://www.mdpi.com/1999-5903/11/3/70.

[31] HiveMQ Team, *MQTT Essentials*. HiveMQ, 2015.

[32] PRESERVE Project, "Security Requirements of Vehicle Security Architecture", Tech. Rep., Jun. 2011. [Online]. Available: https://trimis.ec.europa.eu/sites/default/files/project/documents/20121025_114452_74602_PRESERVE-D1.1-Security_Requirements_of_Vehicle_Security_Architecture.pdf.

[33] Ricardo Severino. "Security in hybrid its networks". (Accessed on: 2023-09-28). (2023), [Online]. Available: https://github.com/RicardoFilipe99/Security-in-Hybrid-ITS-Networks.

[34] Jiri Ohnheiser, *Escos roadside unit user manual etsi*, For RSU version 1.4.25, Siemens, 2019.

[35] José Santa, Fernando Pereñíguez, Antonio Moragón, and Antonio F. Skarmeta, "Experimental evaluation of cam and denm messaging services in vehicular communications", *Transportation Research Part C: Emerging Technologies*, vol. 46, pages 98–120, 2014, ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2014.05.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X14001193.

[36] Johan Scholliers, Mirjami Jutila, Mikko Valta, Kimmo Kauvo, Ari Virtanen, and Pasi Pyykönen, "Co-operative traffic solutions for hybrid communication environments", *Transportation Research Procedia*, vol. 14, pages 4542–4551, 2016, Transport Research Arena TRA2016, ISSN: 2352-1465. DOI: https://doi.org/10.1016/j.trpro.2016.05.377. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146516303830.

[37] Wenliang Du, *Public-Key Infrastructure (PKI) Lab*, Website, (Accessed on: 2023-04-10), 2018. [Online]. Available: https://seedsecuritylabs.org/Labs_16.04/PDF/Crypto_PKI.pdf.

[38] Hyeonji Seon, Hojeong Lee, and Hyogon Kim, "Predicting cam generation times through machine learning for cellular v2x communication", *ICT Express*, 2022, ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2022.08.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959522001151.

[39] Alexandru Constantin Serban, Erik Poll, and Joost Visser, "A security analysis of the etsi its vehicular communications", in *Computer Safety, Reliability, and Security*, Barbara Gallina, Amund Skavhaug, Erwin Schoitsch, and Friedemann Bitsch, Eds., Cham: Springer International Publishing, 2018, pages 365–373, ISBN: 978-3-319-99229-7. DOI: https://doi.org/10.1007/978-3-319-99229-7_31.

[40] Andrew Macleod, *Autonomous driving, smart cities, and the new mobility future*, Website, (Accessed on: 2023-02-16), Jul. 2019. [Online]. Available: https://www.techbriefs.com/autonomous-driving-smart-cities-and-the-new-mobility-future/.

[41] Nidor Huang, *Quick Start Guide for OBU-300 Family*, Unex-QSG-21-001, Unex, 2019.

[42] Fei Wang, Yongjun Xu, Hanwen Zhang, Yujun Zhang, and Liehuang Zhu, "2flip: A two-factor lightweight privacy-preserving authentication scheme for vanet", *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pages 896–911, 2016. DOI: 10.1109/TVT.2015.2402166.

[43] Albert Wasef, Rongxing Lu, Xiaodong Lin, and Xuemin Shen, "Complementing public key infrastructure to secure vehicular ad hoc networks [security and privacy in emerging wireless networks]", *IEEE Wireless Communications*, vol. 17, no. 5, pages 22–28, 2010. DOI: 10.1109/MWC.2010.5601954.

[44] Wenliang Du, *Computer & Internet Security: A Hands-On Approach*. Independently, 2017.

[45] World Health Organization, "European regional status report on road safety", Tech. Rep., Nov. 2019. [Online]. Available: https://visaozero2030.pt/wp-content/uploads/European_Regional_Status_Report_Road_Safety_2019.pdf.

[46] World Health Organization, "Global status report on road safety", Tech. Rep., Jun. 2018. [Online]. Available: https://www.who.int/publications/i/item/9789241565684.

[47] Filip Machovec, *Escos roadside unit its xfer gateway interface specification*, For RSU version 1.2.2, Siemens, 2019.

[48] Takahito Yoshizawa, Dave Singelée, Jan Tobias Muehlberg, Stephane Delbruel, Amir Taherkordi, Danny Hughes, and Bart Preneel, "A survey of security and privacy issues in v2x communication systems", *ACM Comput. Surv.*, vol. 55, no. 9, 2023, ISSN: 0360-0300. DOI: 10.1145/3558052. [Online]. Available: https://doi.org/10.1145/3558052.