future internet

MDPI

Article

Protecting Hybrid ITS Networks: A Comprehensive Security Approach

Ricardo Severino [1,*], José Simão [1,2,*], Nuno Datia [1,3] and António Serrador [1]

[1] Lisbon School of Engineering (ISEL), Polytechnic University of Lisbon (IPL), 1549-020 Lisboa, Portugal; datia@isel.ipl.pt (N.D.); antonio.serrador@isel.pt (A.S.)
[2] Instituto de Engenharia de Sistemas e Computadores: Investigação e Desenvolvimento em Lisboa (INESC-ID), 1000-029 Lisboa, Portugal
[3] NOVA LINCS, NOVA School of Science and Technology, 2829-516 Monte da Caparica, Portugal
* Correspondence: A45245@alunos.isel.pt (R.S.); jose.simao@isel.pt (J.S.)

Abstract: Cooperative intelligent transport systems (C-ITS) continue to be developed to enhance transportation safety and sustainability. However, the communication of vehicle-to-everything (V2X) 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. Therefore, when developing C-ITS systems, it is important to broaden the focus beyond connected vehicles to include soft-mobility users and legacy vehicles. This work presents a new approach developed in the context of emerging hybrid networks, combining intelligent transport systems operating in 5.9 GHz (ITS-G5) and radio-mobile cellular technologies. Two protocols 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. Commercial equipment was used for this goal, including on-board and roadside units. Computational, transmission and end-to-end latency were used to assess the system's performance. Implemented protocols introduce an additional 11% end-to-end latency in hybrid communications. Moreover, workflows employing hybrid communications impose, on average, an extra 28.29 ms of end-to-end latency. 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; cellular network; hybrid network; security; privacy; integrity; V2X

Citation: Severino, R.; Simão, J.; Datia, N.; Serrador, A. Protecting Hybrid ITS Networks: A Comprehensive Security Approach. *Future Internet* **2023**, *15*, 388. https://doi.org/10.3390/fi15120388

Academic Editor: Naoki Shibata

Received: 7 November 2023
Revised: 24 November 2023
Accepted: 27 November 2023
Published: 30 November 2023

1. Introduction

Transportation has always been crucial in human society. It connects people, allows access to essential services and promotes prosperity. However, the growing number of vehicles [1] has led to concerns about road traffic and safety. Despite stricter European road safety regulations [2], accidents persist, leading to fatalities. Moreover, increased road traffic has resulted in congestion, higher gas emissions and decreased air quality [3]. The 2018 global status report on road safety [1] from the World Health Organisation (WHO) alerts that the number of road traffic deaths worldwide remains unacceptably high, with 1.35 million people dying each year—the eighth leading cause of death for people of all ages and the number one cause for children and young adults.

Considering these circumstances, 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) [4], particularly in the architecture of solutions that enable vehicles to exchange information with each other (V2V), the road infrastructure (V2I), and with pedestrians (V2P), being therefore known as vehicle-to-everything (V2X). The main C-ITS goal is to enable communication and information exchange among road elements, providing cooperation and, thus, increasing safety, mobility and sustainability [5].

Despite the potential benefits, C-ITS/V2X communications are inherently open. This openness creates vulnerabilities [6] that attackers can exploit, representing a significant threat to all road users, as security failures can lead to privacy violations or even fatalities. These security and privacy challenges must be addressed to ensure that road safety is not compromised [7]. According to Serban et al. [8], "Security plays a crucial role in cooperative applications because a security breach can easily lead to human casualties". Moreover, C-ITS relies heavily on communication between vehicles that have the necessary equipment installed. This issue is also raised by Yoshizawa et al. [9], where it is referred that, although in the European Norm (EN) 302 665 (V1.1.1) [10], the European Telecommunications Standards Institute (ETSI) has defined handheld devices as one of the types of ITS stations, 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 [1]. Therefore, in the development of C-ITS-based systems, it is essential to broaden the perspective beyond connected vehicles, also considering the needs of soft-mobility users (e.g., cyclists) and legacy vehicles that do not have the required equipment—the on-board units (OBUs).

To illustrate these issues, some possible scenarios are described. An attacker could perform a Sybil attack, as illustrated in Figure 1, where he claims the existence of non-existing vehicles at multiple locations [7], creating confusion and disrupting communication. This could lead to inaccurate traffic information (e.g., fake congestion), which can cause misinformed decision-making by drivers.

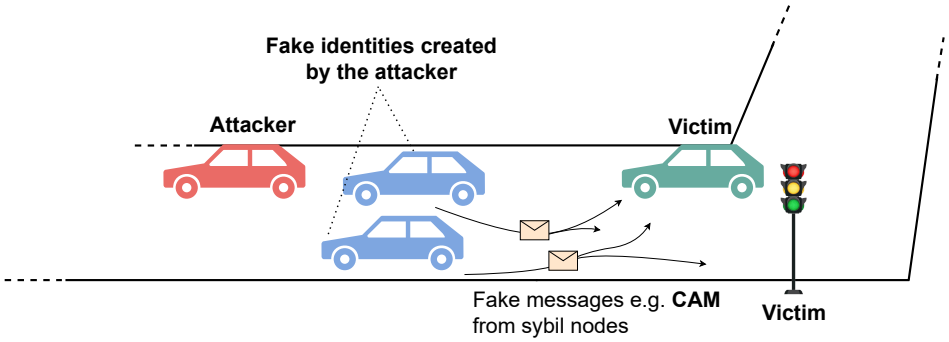


Figure 1. Sybil attack where the attacker claims his existence at multiple locations.

The situation depicted in Figure 2 exemplifies why it is beneficial to include soft-mobility users and legacy vehicles in the C-ITS ecosystem. In this scenario, an accident occurred, and vehicles (legacy and G5-connected) were approaching. The lack of information about the situation makes it unpredictable and unsafe. The cyclist and the pedestrians present could securely notify ITS central systems. Information about this event could then be disseminated to connected vehicles.

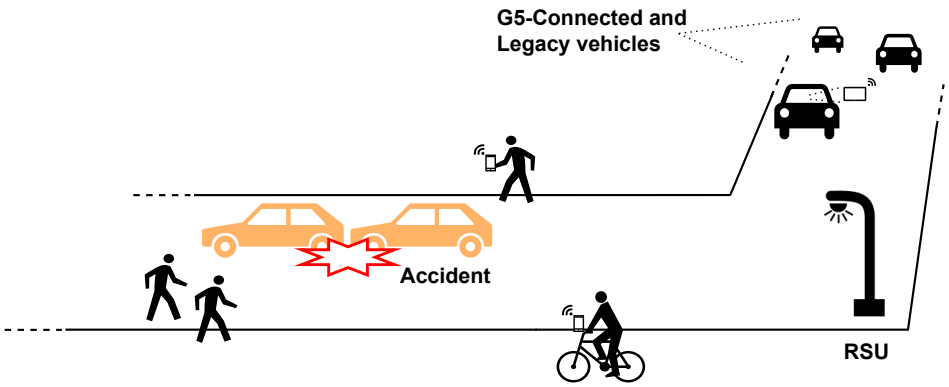


Figure 2. Accident where cooperation between soft-mobility users and the G5 network would be beneficial.

To address previous issues, this work develops a proof-of-concept approach that introduces security guarantees within a C-ITS ecosystem while accommodating soft-mobility users and legacy vehicles. Additionally, this research intends to evaluate and compare the security protocols proposed in the literature using real equipment. Emphasis is also given to assessing how the security protocols affect performance. Lastly, this study measures the performance cost of incorporating soft-mobility users and legacy vehicles within a realistic testing environment. To accomplish these goals, this work builds and assesses a system that employs a security protocol in a C-ITS environment while operating within a hybrid network, combining intelligent transport systems operating in 5.9 GHz (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, such as the fifth- and sixth-generation (5G and 6G) [11,12]. Two security protocols, DLAPP [13] and MFSPV [14], were implemented using hardware equipment—OBUs, roadside units (RSUs) and smartphones. An application was developed for each of these computing environments. These applications allow sending and receiving/verifying protected messages using a protocol.

Viewing through a high-level model, the proposed approach and subsequent experimental environment are embedded in the context of edge computing (EC) and fog computing (FC) paradigms [15]. Devices, such as smartphones, OBUs found in vehicles and RSUs along the road infrastructure, act as communication devices [16] that operate at the edge of the hybrid network. They process data in their local applications, providing low latency and faster response [17]. Lastly, computational, transmission and end-to-end latency were used to assess the system's performance. Our primary contributions are:

- Development and assessment of a novel approach that employs a security protocol in a C-ITS hybrid environment by combining ITS-G5 and radio-mobile networks;
- Extend the literature by going beyond the traditional focus on connected vehicles to include soft mobility users and legacy vehicles in C-ITS;
- Assessing the effectiveness of security protocols (DLAPP [13] and MFSPV [14]), thus bridging the gap between theory/simulation and real-world implementations;
- Enrichment of the literature's information regarding the implementation of security protocols in real ITS equipment (OBUs and RSUs).

This paper is organised as follows. First, Section 2 reviews the existing concepts and research related to this problem, including the implemented security protocols. Section 3 presents the proposed approach and its development. Section 4 analyses and reports the conducted experimental evaluation. Finally, Section 5 concludes the document by summarising the developed work, main remarks and possible future work.

2. Background and Related Work

This section covers important background information, standards and a literature review. Section 2.1 addresses C-ITS technology. Public key infrastructure (PKI) for C-ITS is presented in Section 2.2. Section 2.3 presents the existing research on integrating hybrid networks in ITS. Finally, Section 2.4 showcases the study conducted on the proposed protocols that ensure secure ITS communications.

2.1. C-ITS

A general overview of V2X systems will be given for a better understanding of C-ITS technology. V2X is a collective term incorporating several communication modes that enable communication among road elements. V2X communication systems can use technologies based on the IEEE 802.11p protocol that operates in the 5.9 GHz frequency band, having been designed to standardise vehicular communication systems. The IEEE 802.11p protocol is the basis of some standards for V2X communication [9], including:

- Dedicated short-range communications (DSRC) with wireless access in vehicular environments (WAVE) as the upper layer in the United States of America.

- ITS-G5 with C-ITS as the upper layer in Europe.

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, mobility and sustainability, thus leading to more efficient and safer transportation [5,18]. Moreover, C-ITS is composed of multiple sub-systems [10], such as handheld ITS sub-systems (such as smartphones); the central ITS sub-system; vehicle ITS sub-system present in vehicles; and roadside ITS sub-systems to be on traffic lights and other roadside infrastructures. An ITS-S is a functional entity specified by the ITS-S reference architecture [10]. The reference architecture follows the principles of the open systems interconnection (OSI) model for layered communication protocols [19].

Many ITS applications require one of two communication strategies [20] or a combination of both: Periodic status exchange, where messages are needed by apps to know about the status of a vehicle or a roadside terminal, and event-driven information—messages informing about a specific event. Therefore, ETSI has defined two essential messaging services:

- A cooperative awareness message (CAM) [21] provides awareness of the surrounding environment by periodically sending status data to nodes within a single-hop distance.
- Decentralised environmental notification message (DENM) [22] provides timely and relevant information about the driving environment and traffic events via multi-hop transmission to cover a specific geographic dissemination area.

There are other messages in C-ITS [23], but DENM and CAM are the most widely used.

2.2. PKI as an Architecture for Securing ITS Communications

Public key infrastructure (PKI) plays a crucial role in ensuring the security of digital communications [24]. Particularly, PKI is a building block for C-ITS security, also referred to as C-ITS communications security architecture and security management in the ETSI specification [25]. The most relevant elements are the ITS-S, the root certificate authority (CA), which serves as a trust anchor and provides certificates to the enrolment authority (EA) and authorisation authority (AA). EA manages enrolment credentials (EC), which are long-term certificates used for authentication and access to ITS communications. AA issues authorisation tickets (AT), also known as pseudonym certificates, which are short-term certificates that allow ITS-S to access specific ITS services while masking their identity. The sequence of interactions that occurs from the moment a vehicle intends to enter the C-ITS network and send messages to another ITS-S is depicted in Figure 3.

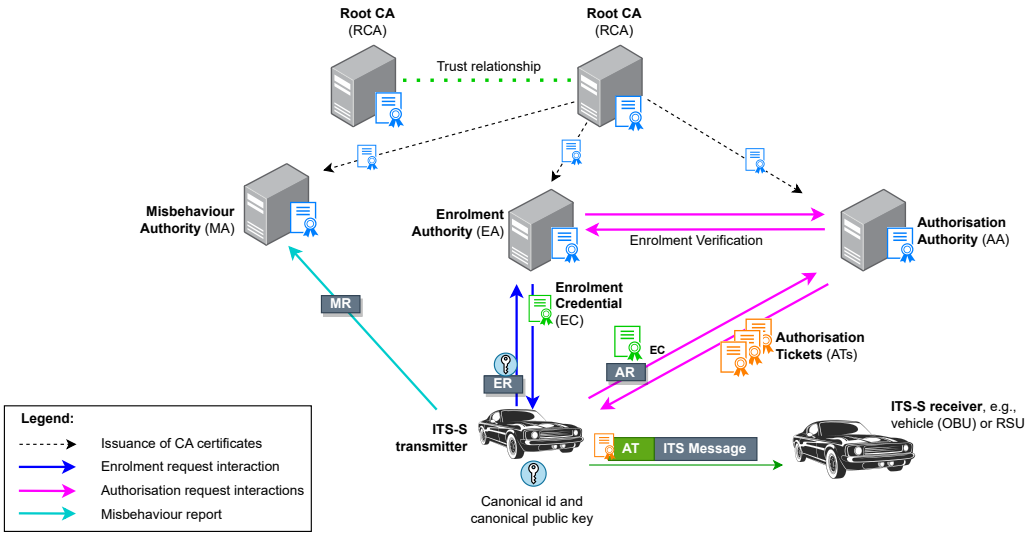


Figure 3. PKI architecture in C-ITS (adapted on [25]).

PKI for C-ITS achieves a high level of security. Regarding handling privacy issues, C-ITS messages include several identifiers that can be used for privacy violations. As a countermeasure, users' privacy is protected by a pseudonym scheme, i.e., changing the AT frequently, which also changes all its identifiers. Message integrity, authenticity and non-repudiation allow properties to be ensured by creating a digital signature (using an AT) over the message. The ITS-S transmitter can use its AT to generate a digital signature. When a signed message is received, the receiver can verify the digital signature. Confidentiality can be ensured by encrypting the packets with a key shared with the ITS-S receiver [26].

Despite the high-security level C-ITS PKI provides, it also has its limitations. The main drawbacks of PKI are due to processing and communication overhead latency [13]. The first one is due to the use of asymmetric cryptography to sign and verify each message, which is quite computationally demanding. The ability of each vehicle to check its certificate revocation list 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 [27]. Lastly, C-ITS PKI experiences high communication overheads because the certificate sent (AT) for message verification is large, causing inefficiency.

2.3. Hybrid ITS Networks Approaches

In 2022, Gonçalves et al. [28] aimed to develop a system capable of enhancing users' awareness regarding potentially dangerous situations around them. They also highlighted 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 cellular), allowing ITS equipment to communicate with other devices (not directly) via Wi-Fi or cellular networks.

Bissmeyer et al. [29] analyse PKI 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 [29], hybrid communications are ideally used to support the reliability of communication by using redundant communication technologies. The security concept is described as securing these communications in the presence of multiple radio technologies using different physical channels to transmit V2X messages.

Lastly, Scholliers et al. [30] conducted 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 to enable the system to react quickly, thus exploiting multiple networks and prioritising them by preference and signal strength, among other criteria. Similarly to [29], the adopted hybrid network is also applied in the context of using LTE as another access technology through an ITS-S.

Each study has its motivations for the use of hybrid networks. Some approaches, such as the ones followed in [29,30], focus on using hybrid networks to utilise different access technologies (G5 and LTE) within an ITS station. Nonetheless, this work develops a C-ITS hybrid environment that integrates non-ITS station users through a smartphone application. The study proposed in [28] comes closest to this work, although it presents a scenario more focused on road safety. In this work, the objective is also to propose a hybrid network architecture while introducing security guarantees, forming a secure hybrid ITS network. Table 1 summarises four relevant aspects of each article, all pertinent to this study.

Table 1. Summary of four relevant aspects in the context of hybrid networks in ITS. Indicates whether the paper: addresses hybrid networks, considers ITS station that uses multiple access technologies, considers users without an OBU and if security aspects are considered.

Reference	Year	ITS Hybrid Networks	ITS-S with Multiple Access Technologies	Consider Users without an OBU	Security
Gonçalves et al. [28]	2022	✓	✗	✓	✗
Bissmeyer et al. [29]	2019	✓	✓	✗	✓
Scholliers et al. [30]	2016	✓	✓	✗	✗

2.4. Architectures for Securing ITS Networks

This section introduces the security protocols implemented in this work. As seen, the standard approach relies on PKI in both European and USA security architectures. Table 2 lists the ETSI specifications related to our C-ITS PKI research. Meanwhile, the C-ITS PKI solution offers a high level of security, but it has certain limitations. As previously discussed, a major drawback of PKI is its inefficiency, a point that has also been emphasised in the literature [7,13,14,31]. These constraints have motivated the research community to explore alternative security protocols. The DLAPP [13] and MFSPV [14] protocols were chosen to be implemented in this study; therefore, they will be briefly described. Our selection criteria for these protocols include their significance within the research community.

Table 2. Security standards summary for C-ITS PKI.

Reference	Document	Title	Version
[32]	TR 102 893	Threat, Vulnerability and Risk Analysis (TVRA)	v1.2.1 2017
[25]	TS 102 940	ITS Communications Security Architecture	v2.1.1 2021
[33]	TS 102 941	Trust and Privacy Management	v1.4.1 2021
[34]	TS 102 943	Confidentiality Services	v1.1.1 2012
[35]	TS 103 097	Security Header and Certificate Formats	v2.1.1 2021

2.4.1. DLAPP

Hakeem et al. [13] proposed a decentralised, lightweight authentication and privacy protocol (DLAPP) that offers authentication and privacy protection. The protocol utilises a biometric device (BD) for driver identification and authentication and a tamper-proof device (TPD) for secure storage and processing. Its objective is to decentralise the CA's tasks by allowing each vehicle to locally generate its own pseudo-identity and private keys rather than relying on frequent communication with a central CA, thus preserving privacy and authentication while reducing the communication workload on the CA.

For message exchanges, it is necessary to perform the pseudo-identity and the hash chain generation. Then, to transmit a message (m), the transmitter calculates a message authentication code (MAC) using a randomly selected chain key (ki) identified by index k_{index}, as shown in Equation (1). For this, it is also necessary to choose a pseudo-identity (PID_i) from the generated set and to extract the current timestamp (T_s).

$$Sig_{ki} = mac_{ki}(PID_i \,||\, m \,||\, T_s) \tag{1}$$

The protocol's proposed message format attaches to the message, the pseudo-identity, the MAC value, the index of the selected key and the current timestamp (Figure 4).

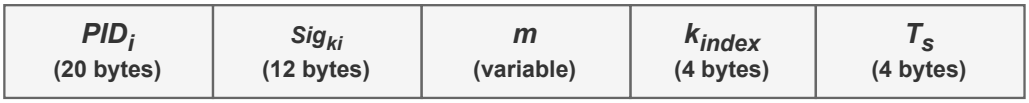


Figure 4. DLAPP proposed message format.

The receiver validates T_s. If valid, it uses the received ki to extract the corresponding key from the locally generated chain and verifies the received MAC. If the calculated MAC, Sig^*_{ki} (via Equation (2)) and the received one (Sig_{ki}) do not match, the message is discarded.

$$Sig^*_{ki} = mac_{ki}(PID_i \,||\, m \,||\, T_s) \tag{2}$$

According to the authors, simulations conducted using the NS-3 simulator demonstrate that DLAPP can sign 60,000 messages per second, up to 55 times higher than other protocols the paper compares itself to. The authors state that DLAPP achieves a communication overhead reduction of 20% to 85% compared to other protocols. They conclude that the DLAPP is well suited to time-critical applications such as large-scale V2X networks.

2.4.2. MFSPV

Alfadhli et al. [14] proposed a multi-factor secured and lightweight privacy-preserving authentication scheme for a vehicular ad hoc network (MFSPV), which employs a combination of physically unclonable functions (PUF) [36] and one-time dynamic pseudo-identities (PID_v) as authentication factors. The intent is to mitigate the heavy dependency that other protocols [13,31] have on the system's key and long-term sensitive data stored in an ideal TPD, which may not be realistic. It aims to decentralise the wide domain of the CA into regional domains by assigning an autonomous regional domain key (R_k) for each region.

The instant a message is ready to be transmitted, it is generated a PID_v, as shown in (3). The message hash signature (ϕ_{vi}) is then calculated, as in Equation (4).

$$PID_v = h(API_{new}||V_{sk}||ID_v||k_{mbr}) \oplus h(API_{new}||t) \quad (3)$$

$$\phi_{vi} = h(PID_v \,||\, R_k \,||\, m \,||\, t) \quad (4)$$

The protocol's proposed message format attaches the chosen pseudo-identity, hash signature and current timestamp (t) to the message m (Figure 5).

t (4 bytes)	ϕ_{vi} (20 bytes)	m (variable)	PID_V (20 bytes)

Figure 5. MFSPV proposed message format.

Regarding message verification, when a surrounding ITS station receives the message, it validates the timestamp. If it is valid, it checks the signature, as shown in Equation (5).

$$\phi'_{vi} = h(PID_v \,||\, R_k \,||\, m \,||\, t) \quad (5)$$

According to the authors, the computation cost for one message verification is 0.006 ms, which offers from 64.0% to 99.9% lighter computation than the protocols that they compare MFSPV to. 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.

2.4.3. Summary

Despite the high level of security provided by the C-ITS PKI solution, it has limitations. The main one is the lack of efficiency due to the use of asymmetric cryptography and the large size of the attached certificate. The previous papers [13,14] present lightweight protocols that attempt to mitigate these constraints. Tables 3–5 summarise some of the results claimed by the protocol's authors in terms of security properties and efficiency.

Table 3. Computation latency comparison for message signature and verification.

Operation	Hakeem et al. [13] DLAPP [ms]	Alfadhli et al. [14] MFSVP [ms]
Signature	0.0167	0.018
Verification	0.0167	0.006

Table 4. Comparison of the communication overhead introduced by each protocol when sending a message (e.g., CAM or DENM).

Efficiency Metric	Hakeem et al. [13] DLAPP	Alfadhli et al. [14] MFSVP
Overhead (bytes)	40	44

Table 5. Comparing some security properties attained by each protocol (according to the respective paper).

Security Properties	Hakeem et al. [13] DLAPP	Alfadhli et al. [14] MFSVP
Integrity and Authenticity	✓	✓
Privacy	✓	✓
Non-repudiation	✓	✓
Resistance to DoS	✓	✓
Resistance to message replay attack	✓	✓
Traceability	✓	✓

The DLAPP [13] and MFSPV [14] protocols have additional features. Nonetheless, the focus of this work centres around security guarantees during message exchanges. For more comprehensive details, please consult the respective papers.

3. Proposed Approach

This section presents the approach proposed to achieve the outlined objectives. First, a high-level overview of the approach is given. Section 3.1 shows greater detail of its architecture. Finally, Section 3.2 provides insight into the implementation.

This study aims to build and assess a proof-of-concept system that employs a security protocol in a C-ITS hybrid environment. Thus, 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 (apps) and vice versa. As for security, DLAPP [13] and MFSPV [14] protocols are implemented, evaluated and compared using real equipment. To illustrate the proposed approach to achieve these goals, a simplified depiction is provided in Figure 6.

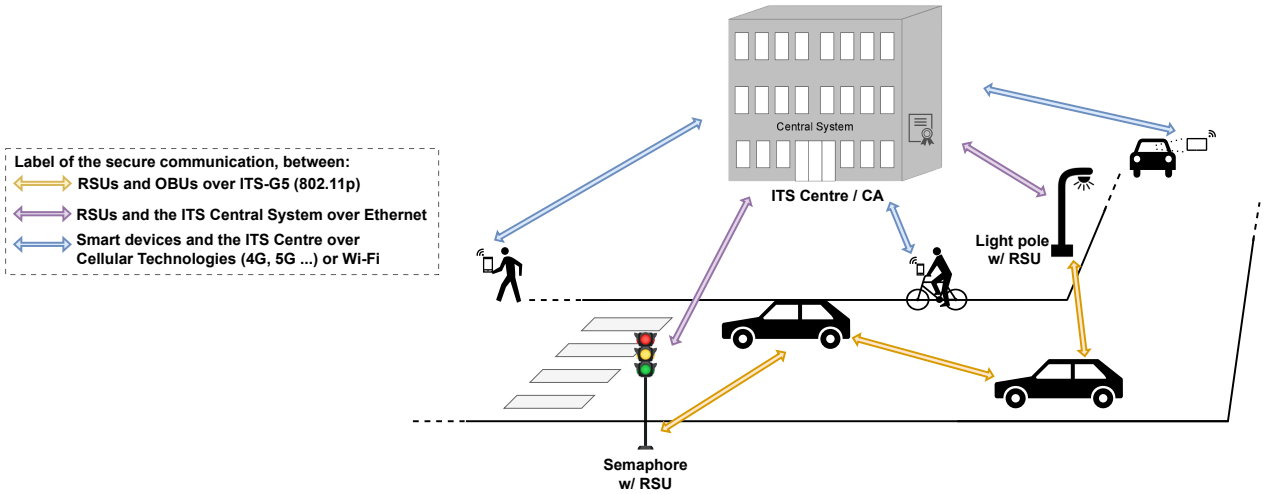


Figure 6. Simplified representation of the proposed approach.

A soft-mobility user message (e.g., CAM) is transmitted through a mobile app to the ITS centre, which then relays it to road infrastructures, such as 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 messages to mobile apps. This approach establishes a bi-directional 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 [14] and DLAPP [13], so one of them can be used, thus enabling information sharing among road users with security guarantees.

3.1. Proposed Architecture

In this section, the architecture of the proposed approach and the elements that constitute it will be described.

3.1.1. Domains and Entities

Figure 7 presents a more detailed perspective of Figure 6. The proposed approach can be separated into three domains: cellular network, ITS centre and ITS-G5 network.

Cellular network In this domain, the entities (physical computational nodes) are the smartphones used by soft-mobility users and legacy vehicle drivers. The mobile app can receive and verify messages as well as disseminate protected messages. Technologies such as 5G and 6G can be used to fulfil reliability and low-latency requirements on information exchange [11,37,38].

ITS centre Represents an ITS central system. It is composed of a "server" entity that hosts two main services: The CA service, which is crucial for user registration, cryptographic material exchange and security updates, among others, and the pub/sub broker, where the pub/sub communication pattern suits the hybrid C-ITS environment since decoupled and asynchronous communications are desired, and multiple producers/consumers exist [39]. Thus, a broker is critical to enable the ITS centre to flow data bi-directionally. Furthermore, the pub/sub model has been used as a powerful tool to develop many distributed applications; it suits the paradigm of edge computing and fog computing (in which this work is embedded), as we are dealing with latency-sensitive applications [17] and resource-constrained devices.

ITS-G5 network Its entities are the ITS stations. The applications of OBUs and RSUs allow them to communicate via G5. The RSUs have direct communication with the ITS centre, thus being the connection point between the cellular and G5 networks.

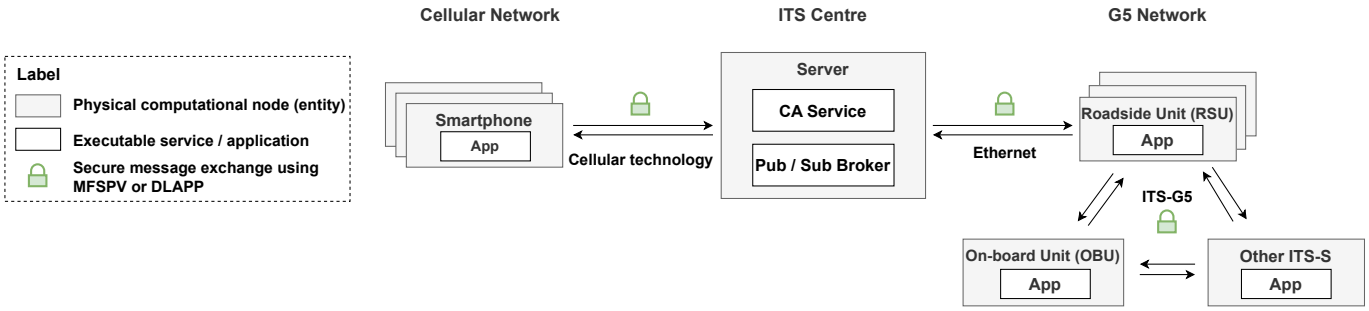


Figure 7. Architecture of the proposed approach.

The padlock label in Figure 7 indicates a secure exchange of messages using MFSPV [14] or DLAPP [13]. 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 occurs at the initialisation of each app.

3.1.2. Message Exchange Scenarios

The scenarios contemplated for exchanging messages with security guarantees are illustrated in Figures 8–10. These message exchanges assume the configuration of a security

protocol (MFSPV or DLAPP) to protect and deprotect (verify) a message. The numerical sequence in each figure represent the order of actions performed.

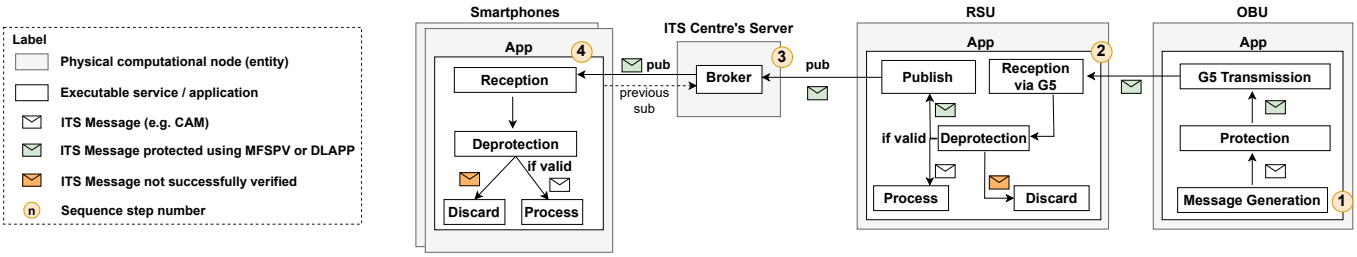


Figure 8. Flow diagram when an OBU generates a message, including the actions' order.

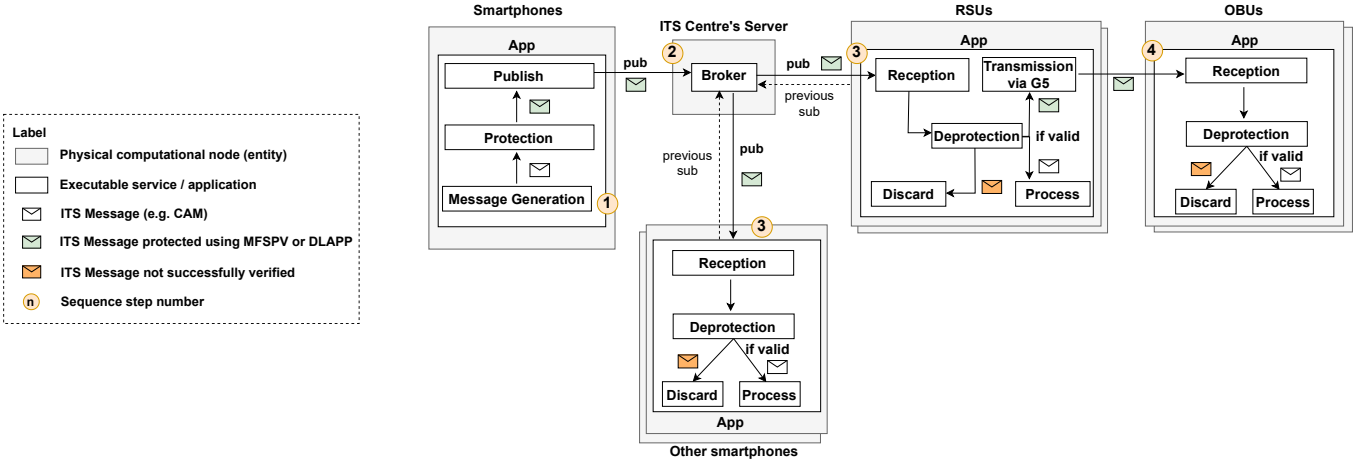


Figure 9. Flow diagram when a smartphone generates a message, including the actions' order.

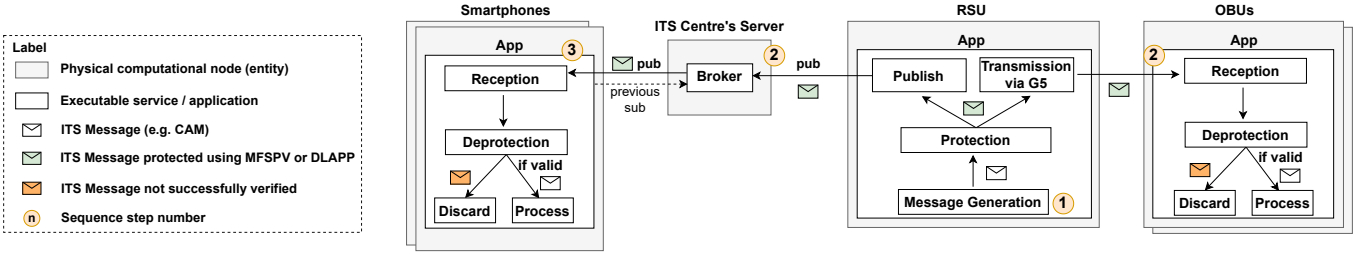


Figure 10. Flow diagram when the RSUs generate a message, including the actions' order.

Firstly, the sequential flow when the OBU generates a message in the G5 network is described in Figure 8. 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 9 represents the inverse, i.e., the sequential flow when a smartphone generates a message. 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-Ss. 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 10. The step-by-step is as follows: (1) The RSU generates the ITS message, protects it and sends it to G5 and cellular networks. (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 [14] and DLAPP [13], decentralise it so that there is no communication with the CA during the message exchange process.

3.2. Implementation

In this research, the CA service (Figure 7) was not implemented, as in this phase, the priority was to exchange messages with security guarantees, and as seen (Section 3.1.2), the CA is not utilised in that procedure. Consequently, when evaluating the protocols and system's performance regarding secure message exchanges and hybrid networking, the absence of the CA does not impact the contemplated use cases or objectives. Nevertheless, it is to emphasise the importance of its implementation in future iterations of this work to test the complete system.

Regarding the broker technology, message queuing telemetry transport (MQTT) was used in this implementation. As stated in [40], MQTT is lightweight and suitable for constrained environments. Thus, it is an approach for building event-driven solutions across edge and fog layers. In addition, its choice prevailed over other message-oriented middleware because when considering essential criteria such as latency, bandwidth/overhead and standardisation, MQTT stands out over other technologies, such as the advanced message queuing protocol (AMQP), Kafka and ZeroMQ [41]. In particular, compared to Kafka [42], MQTT is simpler in terms of implementation complexity [41]. Due to the greater complexity of Kafka, e.g., the ability to store events (which is not relevant to this scenario, as the goal is to process messages timely), there are occasions that Kafka demands a more resource-intensive and slower process [43]. These traits make Kafka less ideal than MQTT for our environment, where there is resource-constrained equipment in EC and FC.

Each node application implements and can be configured with one of three security approaches: No security, DLAPP [13] or MFSPV [14]. 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. 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 the security impact on the performance. 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 were also used, such as the V_{sk} and the R_k. Each entity's application implementation will be briefly described, highlighting important considerations related to their development.

3.2.1. OBU

The OBU equipment (present in vehicles) used was the Unex EVK-301E. Two main difficulties were encountered during its application development.

The initial challenge arose from the difference between the execution and development environments. The execution environment was the equipment itself, which has an armv7-a architecture and a Linux Yocto operating system (OS). On the other hand, the development environment was an Ubuntu Linux 18.04 LTS 64-bit OS. Different compiling and running environments led to cross-compile and system library compatibility issues.

The other challenge was how to send messages in the format proposed by the protocols (Figures 4 and 5). 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 also occurs in the RSU [44]). Therefore, the chosen approach involves incorporating the protocol's security bytes into the messages using optional fields. More precisely, the "PathHistory" field [21], was utilised within CAM messages. While the "PathHistory" field is not being employed with its intended semantics, it allowed the development of the application using real equipment (according to the established objectives), avoiding the need for protocol stack and software modifications on the equipment.

Regarding the development of the OBU application, it was developed in the C programming language using the V2Xcast software development kit (SDK) available for the Unex OBU. As defined in the approach architecture (Section 3.1), 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 can be seen, both involve a conversion process that is necessary due to the previous issue. OpenSSL and [45] were used as cryptographic libraries.

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

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

Algorithm 2 OBU app—Message receiving 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: *valid_its_message* ← security_protocol.deprotection(secure_message)
5: **if** *valid_its_message* = *None* **then**
6: . . . invalid message, discard it . . .
7: **else**
8: . . . message successfully verified, continue processing it . . .
9: **end if**
10: **end function**

3.2.2. RSU

The RSU equipment used in this work was the Siemens ESCoS RSU. RSUs are present in road infrastructure, connecting it to the G5 network.

In the equipment documentation [46], the interface XFER is described. It is an RSU interface based on WebSocket Secure (WSS). It provides bi-directional data exchange and device management functions [44]. It also enables the issuance of commands to the RSU

so that it has certain behaviours; for instance, echoing a received message. Therefore, this interface was chosen to help implement the RSU application. However, this approach requires the presence of a client. An intermediary component, Middleware RSU (M_RSU), was developed to address this, acting as an XFER client. For this reason, the Siemens RSU will be referred to as Physical RSU (P_RSU). The combination of these two components is referred to as the RSU, as both combined have the expected behaviour of an RSU in the proposed approach. The M_RSU application was developed in Python and used the standard library modules for cryptographic operations.

As seen before, the RSU handles three different message exchange scenarios. Thus, M_RSU and P_RSU act together to perform them. Each one of them is described below:

Cellular Network (Smartphone) → RSU First, 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 the format that the OBU requires.

G5 Network (OBU) → RSU As an XFER client, the M_RSU uses the "subscribe" command to instruct the P_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 published for smartphones to receive.

RSU → G5 and Cellular network In this scenario, the messages are being generated by the RSU. M_RSU will protect them and send them to the cellular network (via the broker) and the G5 network (via the P_RSU).

3.2.3. Smartphone

Lastly, the smartphone application was implemented as an Android app using the Java Cryptography Architecture (JCA) as the cryptographic library. Mirroring the dual functionality in the OBU application, it primarily focuses on two services: transmitting locally generated messages and receiving messages from the broker.

4. Experimental Evaluation and Results Analysis

This section outlines the experimental evaluation process and analyses the obtained results. All latency results depend on the used equipment, namely the node computational capacity. Therefore, nodes with high computational capacity will decrease the latency overheads presented. The testing environment is depicted in Figure 11.

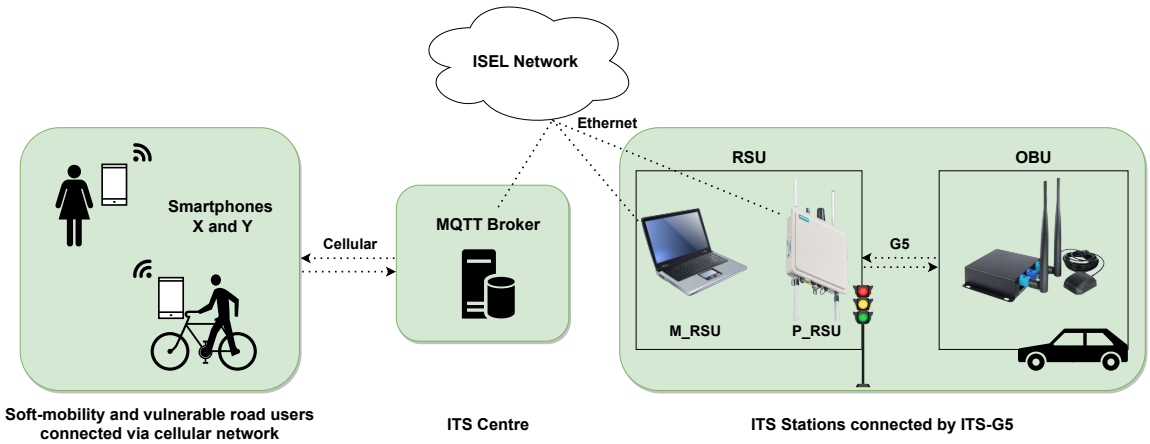


Figure 11. Representation of the testing environment.

On the left is a representation of the cellular network (e.g., 5G) serving soft-mobility users with two smartphones (X and Y). 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, thus bringing the evaluation closer to a real-world scenario. On the right, and also connected to ISEL's network, are the laptop (M_RSU) and the Siemens P_RSU. This one communicates with the OBU, both representing the ITS stations (e.g., a traffic light and a car) connected by ITS-G5.

In this experimental setup, the environment is structured in the context of the EC and FC models [15,47]. Smartphones X and Y, along with the Unex OBU and the RSU, operate at the edge of the hybrid network and process data within their respective local application. In addition, the ITS centre acts as an intermediary in the fog layer due to its proximity to the edge [17,48]. Table 6 describes each equipment computational environment.

Table 6. Characteristics of the computational environment where the prototype was tested.

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

The developed work is assessed in different aspects. Firstly, in Section 4.1, the local computation time of each execution environment application (RSU, OBU and smartphone) is measured. Next, in Section 4.2, communication latency is measured for three security approaches. Finally, in Section 4.3, the end-to-end times of each workflow are calculated.

4.1. Computation Time

This section performs a local computation performance comparison and analysis in each node using three security approaches—No security, DLAPP [13] and MFSPV [14]. This allows 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 times (CTs) 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 CTs 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.

In summary, the evaluation objective is to measure the total and security CTs in each computing node without considering the network latency, only the local computing.

The extraction of the necessary timestamps in each mode is illustrated in Figure 12. These are used in the computation time calculation, as given by Equations (6) and (7).

$$t = TF - TI \tag{6}$$

$$s = SF - SI \tag{7}$$

where:

t—total CT measurement;
TI—total CT initial timestamp;
TF—total CT final timestamp;
s—security CT measurement;
SI—security CT initial timestamp;
SF—security CT final timestamp.

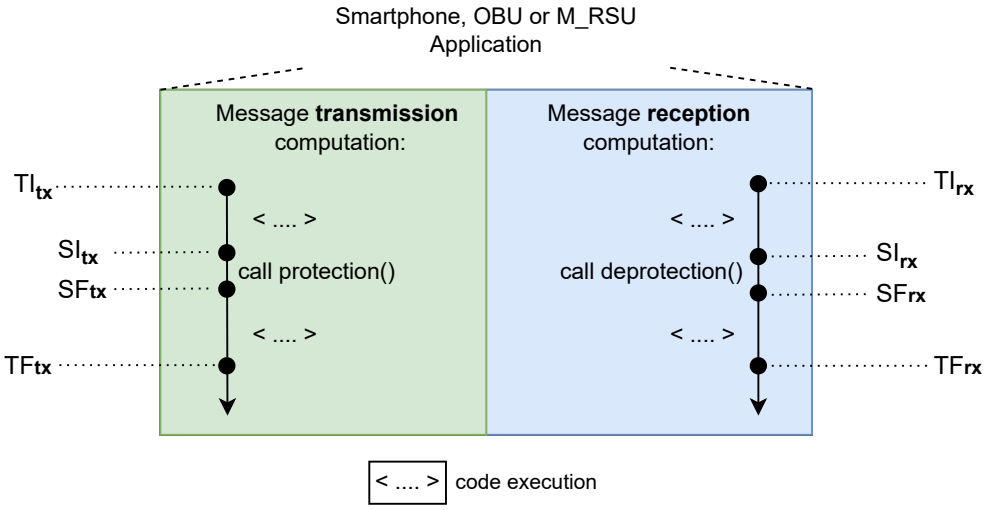


Figure 12. Total and security computation times extraction representation for transmission (tx) and reception (rx).

Note that the processing results obtained by the RSU will be given less emphasis as it does not fully represent the actual RSU execution environment. Therefore, regarding the computation performance evaluation, the OBU and smartphone results are more relevant. From the set of all combinations—computing node, evaluation mode, and security approach—about 2000 samples were extracted to make the obtained values more accurate.

4.1.1. Performance Analysis: DLAPP

According to the DLAPP's proposal [13], its signature and verification simulation took 0.0167 ms (each operation). However, the study only measured the time of the HMAC cryptographic operation, thus being a theoretical estimation. 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 7 (from a total of ∼500 messages, one per second).

Table 7. Median security CT latency using DLAPP in each node.

Node	Protection Latency [ms]	Deprotection Latency [ms]
Smartphone X	0.158	0.162
OBU	0.366	0.327
RSU	0.127	0.084

It can be concluded that, when using hardware (OBU and smartphones), the actual performance falls short (∼90 to 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 [13] does, the smartphone can protect up to 6311 and deprotect 6165 messages per second (Figure 13). 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 [14], as can be seen by its specifications in Table 6. 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.

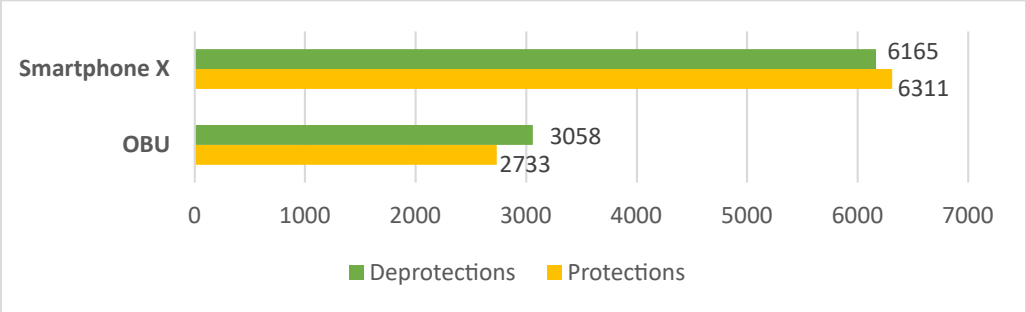


Figure 13. Total DLAPP operations (per second) in the developed applications for OBU and smartphone X.

Assuming a similar high-vehicle-density scenario as [13], 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 13), the DLAPP protocol is computationally light enough to manage such a type of high-node-density scenario.

4.1.2. Performance Analysis: MFSPV

Similar to the DLAPP proposal [13], MFSPV's authors [14] only considered the SHA-256 cryptographic operation to calculate the generation and verification times. They claim that MFSPV's protection takes 0.018 ms and that deprotection takes 0.006 ms. The MFSPV's performance results of this study are shown in Table 8 (from a total of ∼500 messages).

Table 8. Median security CT ltency results using MFSPV in each node.

Node	Protection Latency [ms]	Deprotection Latency [ms]
Smartphone X	0.136	0.107
OBU	0.167	0.153
RSU	0.138	0.064

By analysing the results, and as concluded in the DLAPP's performance analysis, the estimations provided in the proposal protocol [14] are higher than the ones obtained (for the same reasons). Moreover, smartphone X can protect up to 9343 and deprotect 7327 messages per second (Figure 14). 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. Deprotection exhibits lower computation times (∼8% to 53%) on all nodes. This occurs because the protocol performs more hash operations in protection than in deprotection.

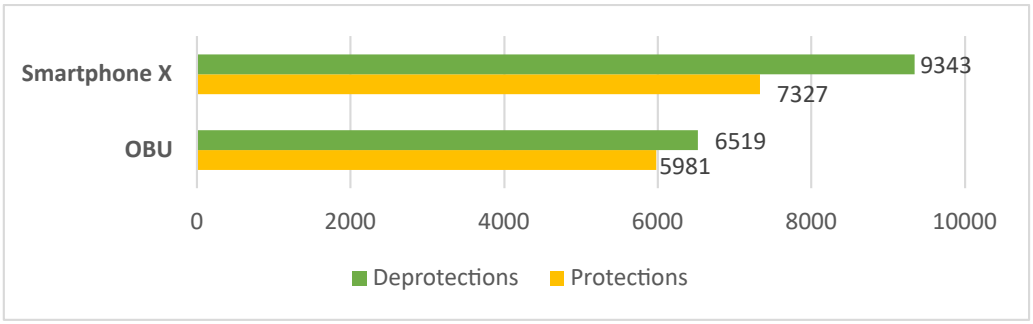
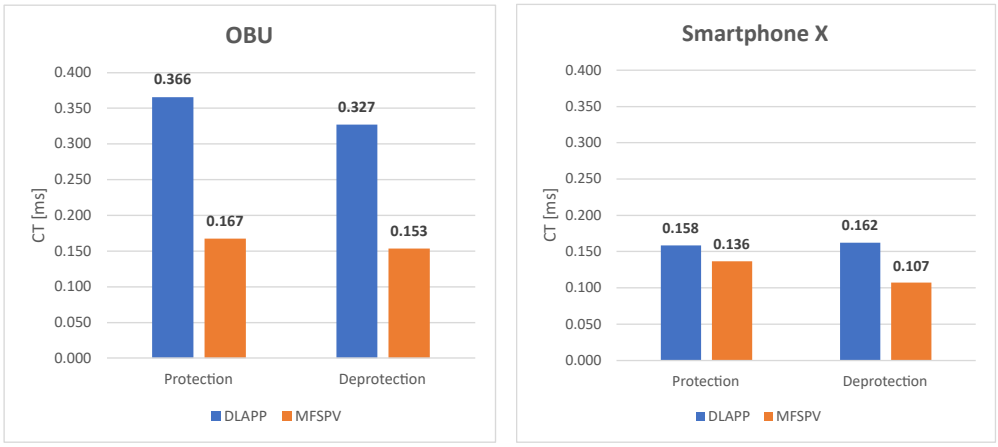


Figure 14. Total MFSPV operations (per second) in the developed applications for OBU and smartphone X.

Assuming the previous high-vehicle-density scenario, i.e., 1800 messages needing to be verified per second, the MFSPV was also computationally light enough to manage this high-node-density scenario on OBUs and smartphones.

4.1.3. Performance Analysis Comparison

Figure 15 reports the median security CT results for DLAPP and MFSPV in OBU and smartphone X. MFSPV outperforms DLAPP in both nodes. Analysing this difference from the perspective of operations per second, MFSPV allows the protection of 3248 and 1016 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 15. Median security CT results for DLAPP and MFSPV in (**a**) OBU and (**b**) smartphone X (in a total of ∼400 measurements).

4.1.4. Security Impact on Performance

This assessment measures transmission and reception computation times across all three security approaches. As a result, it provides insights into the impact of MFSPV and DLAPP on the application's performance. Figure 16 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 performances.

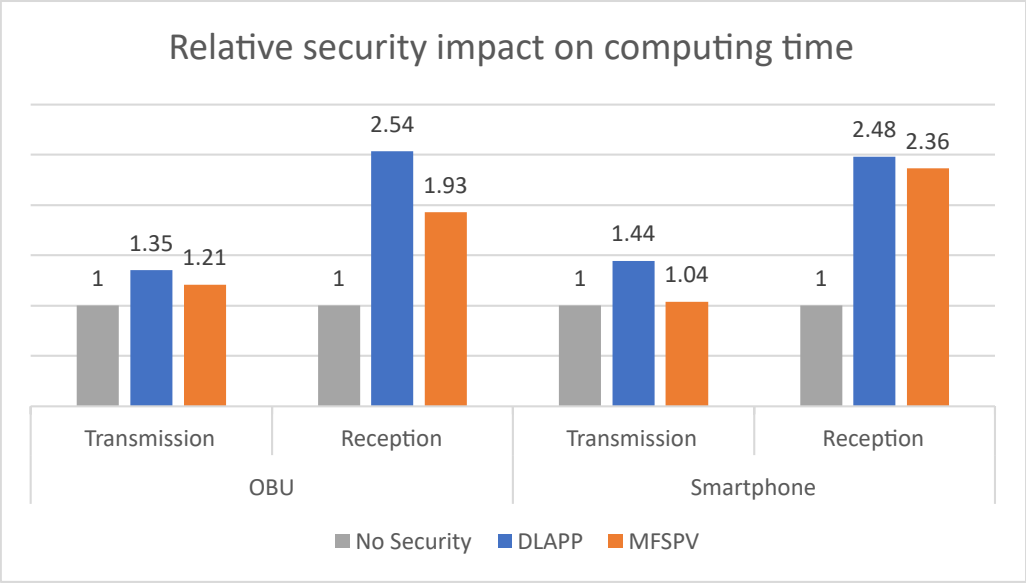


Figure 16. Performance impact of security protocols (DLAPP and MFSPV). The computation times are relative ratios to the respective 'No Security' task.

In transmissions, DLAPP increases the computational time by 35% on the OBU and 44% on the smartphone. In comparison, MFSPV increases it by 21% on the OBU and 4% on the smartphone. DLAPP has a more significant impact on the 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). 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.

4.2. Network Latency

Latency is an important performance indicator in communication. For this reason, this section assesses the latency of the developed hybrid network. To perform these tests, the round-trip time (RTT) was utilised. Figure 17 describes the adopted methodology.

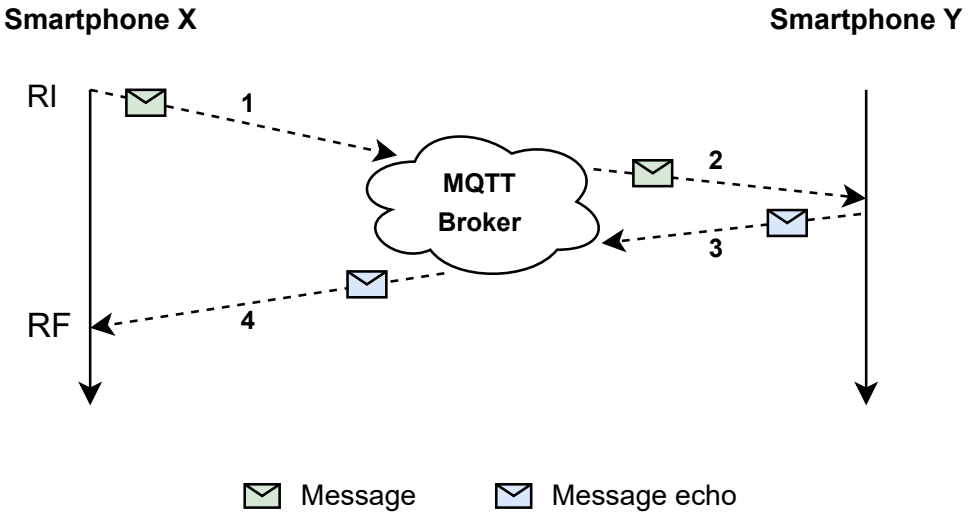


Figure 17. Methodology for calculating the RTT in communications involving the cellular network.

This methodology was used to calculate the communications latencies involving the cellular network segment. The timestamps (rtt_ts_{start} and rtt_ts_{end}) are used to calculate the transmission latency, as given by Equation (8).

$$cl = \frac{RF - RI}{2} \tag{8}$$

where:

cl — latency measurement of a cellular network segment;
RI — initial timestamp;
RF — final timestamp.

The above methodology was used in the calculation of latency of the following communication flows: (i) X → Smartphone Y, (ii) Smartphone X → RSU, (iii) M_RSU → Smartphone X. A slightly different strategy was adopted to calculate the communication latency in G5 (between the P_RSU and OBU). Figure 18 illustrates the methodology used, while Equation (9) shows the latency calculation.

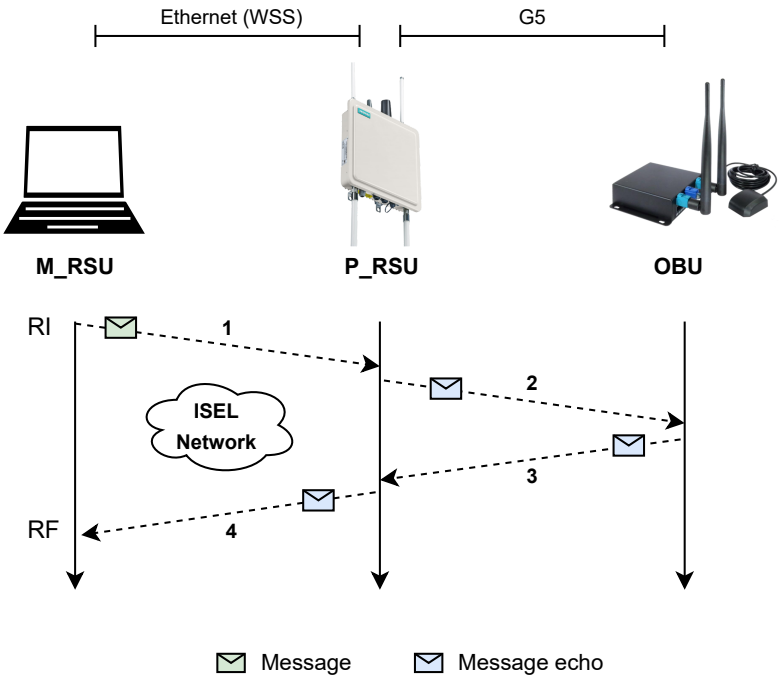


Figure 18. Methodology for calculating latency in G5 communications.

$$gl = \frac{RF - RI}{2} - wl \tag{9}$$

where:

gl — latency measurement of G5 network segment;
RI — initial timestamp;
RF — final timestamp;
wl — latency measurement of WSS communication (between the M_RSU and the P_RSU).

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. 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 19, the median values are reported.

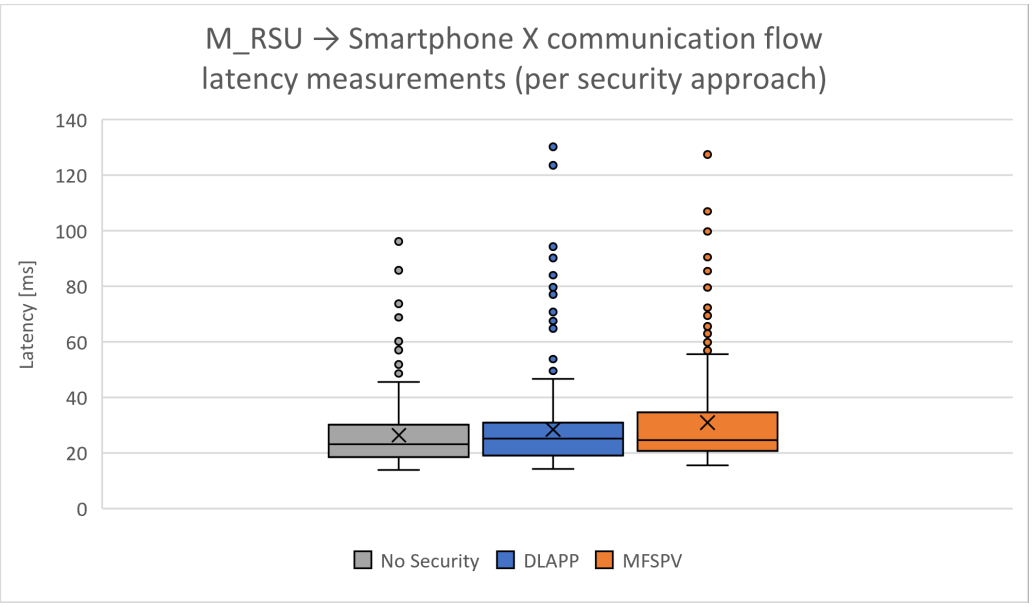


Figure 19. Box plot of latency measurements in M_RSU → Smartphone X communication flow.

4.2.1. Latency Measurements Analysis: Cellular Network

The latency values involving the cellular network are shown in Table 9.

Table 9. Communications latency measurements results involving the cellular networks using different security approaches.

Communication Flow			No Security [ms]	DLAPP [ms]	MFSPV [ms]
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

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 11), the M_RSU is in a privileged position as it is connected via Ethernet to the ISEL network, as is the MQTT broker. It is also observed that there are higher latencies in communications between smartphones, which is justified by the fact that both are on the cellular network, which contributes to higher latencies.

Upon individual analysis of each communication flow, the omission of a security protocol results in the most favourable latency measurements, which is expected due to message payload overhead. Among the results of each protocol, DLAPP, with four less bytes of overhead than MFSPV, exhibits slightly better performance on two occasions when compared to MFSPV. These results indicate that the difference of 4 bytes does not significantly influence the use of one protocol over the other. On average, when compared to scenarios where no security is used, DLAPP increases the cellular network latency by 6.8%, while MFSPV increases it by 7.5%.

4.2.2. Latency Measurements Analysis: G5 Network

The latency measurements of G5 communications (RSU and OBU) are reported in Table 10. 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 10. ITS-G5 communications latency measurements between RSU and OBU using different security approaches.

No Security [ms]	DLAPP [ms]	MFSPV [ms]
10.196	10.792	11.251

4.2.3. Latency Measurements Analysis Comparison

The comparison of the latency measurements of the cellular network and the G5 is illustrated in Figure 20. The G5 network, on average, has 63.6% lower latency than the cellular network. It achieves a 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 the 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, this difference is minimal as it represents tenths of milliseconds, which is justified by the fact that there is only a 4-byte difference in the payload.

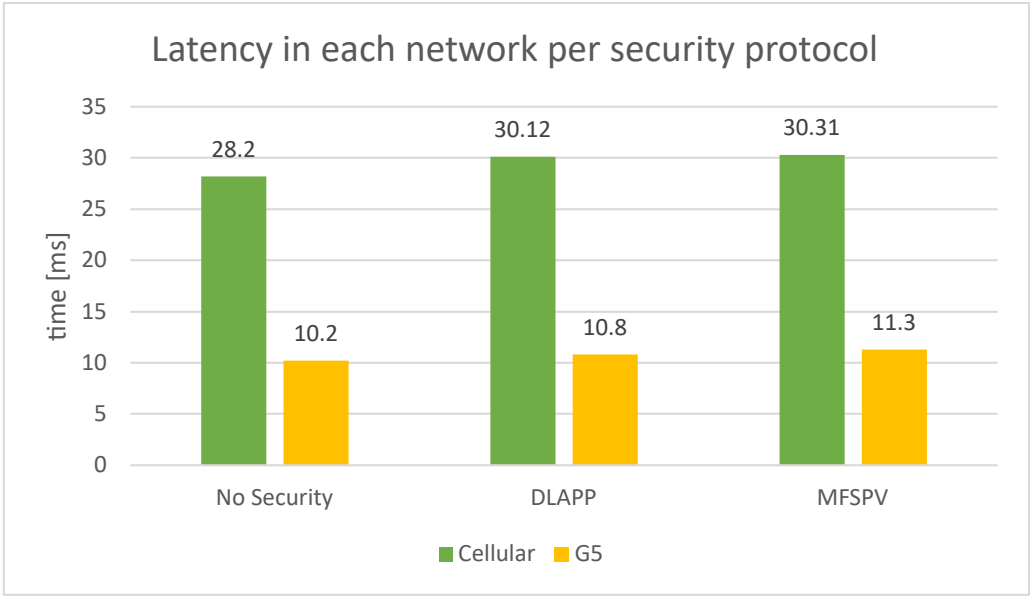


Figure 20. Cellular and G5 latency measurements comparison for each security approach. The cellular latencies are the average of the ones reported in Table 9.

4.3. End-to-End Assessment

End-to-end (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 sections 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 uses the median values obtained in the computing and networking latency experiments. 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 11.

Table 11. E2E times for the various prototype flows with different security approaches. Communication flows are divided according to the network segment they use.

Network Segment	Communication Flow			No Security [ms]	DLAPP [ms]	MFSPV [ms]
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 are analysed from two perspectives: the network segment and security approach. Following this analysis, the applicability of the developed proof-of-concept is briefly discussed based on the results obtained.

4.3.1. 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, which was anticipated. The hybrid network's communications flow 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, where 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 11), it can be concluded that hybrid communication flows impose (on average) an extra 28.29 ms of E2E time, which translates into an increase of 218% compared to G5-only communication flows. Table 12 illustrates the average E2E results obtained in each network segment.

Table 12. Average E2E communication flow latencies associated with each network segment.

G5 [ms]	Cellular [ms]	Hybrid [ms]
12.97	31.54	41.26

4.3.2. Analysis per Security Approach

Figure 21 summarises the E2E results, focusing on the analysis of the protocol impact.

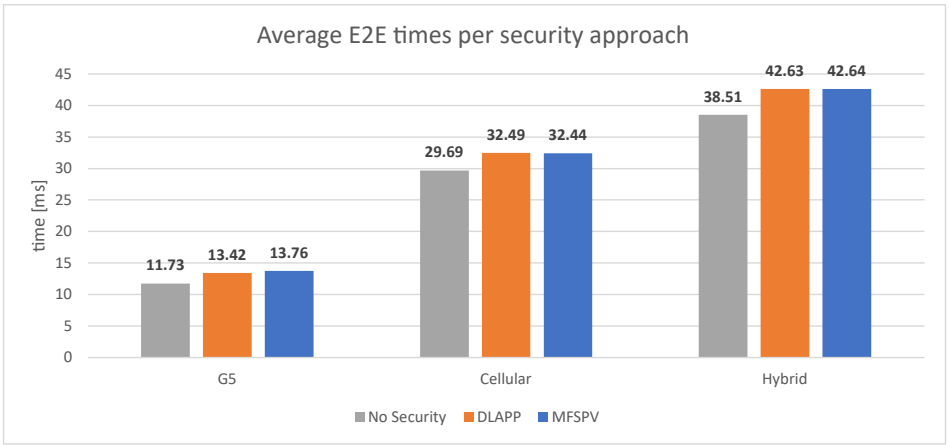


Figure 21. Average E2E times of communication per security approach. The study is also divided according to the network segment.

As observed in the experiments conducted so far, the security protocols have a low impact on performance compared, for instance, to the extension to hybrid networks. In the E2E results, the same 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 attains slightly shorter latency values. Nonetheless, looking at the big picture (Figure 21), both impose a similar additional E2E time.

4.3.3. Applicability Considerations

On a final note, as referred in [49,50], 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 11, do not surpass ∼47 ms. This 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 is important to note that these values are considered outliers. Outliers were identified using the interquartile range (IQR) method, specifically, values above $Q3 + 1.5 \times IQR$ or below $Q1 - 1.5 \times IQR$, as illustrated in the box plot in Figure 19.

Lastly, the same analysis was repeated, i.e., considering maximum values but now excluding outliers. In this case, the maximum E2E time observed was 86 ms in the smartphone X → OBU communication flow using the MFSPV. This means that, when assuming the worst-case scenario while excluding outliers, the results obtained in this study still remain at 14% below the maximum latency requirements for many use cases [50], such as automated shuttle remote driving [49].

5. Conclusions

C-ITS/V2X communication is open and vulnerable to attacks, posing privacy and safety risks to road users. Moreover, a high fatality rate is correlated with more vulnerable modes of transportation. Therefore, developing C-ITS solutions requires considering all road users' needs, not only vehicles. This study proposed a security approach within a C-ITS ecosystem while accommodating soft-mobility users and legacy vehicles. Security protocols were used in a C-ITS setting that enabled integration between connected ITS stations using ITS-G5 and soft-mobility users through smartphones over cellular technologies (hybrid networks). Two security protocols (MFSPV and DLAPP) were implemented using real hardware equipment (OBUs, RSUs and smartphones), and for each computing environment, an application was developed. Experiments were performed to evaluate the developed ecosystem. More specifically, computational, transmission and end-to-end latency were assessed.

For the used experimental setup, MFSPV proved to be 16% to 113% more efficient than DLAPP, depending on the computational node and operation (protection or verification). 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. Moreover, both presented a 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 6.4%, whereas MFSPV provides an 8.8% increase. Furthermore, the G5 network, on average, has 63.6% lower latency times when compared to the cellular network. Regarding the end-to-end assessment, the most time-consuming E2E communication flows were seen in the hybrid

network communication flows, which is expected since messages travel via both G5 and cellular networks. In particular, the highest E2E time was 46.75 ms. Conversely, the E2E times achieved by G5-exclusive communication flows were the lowest. On average, the extension for hybrid communication imposes an extra 28.29 ms of E2E time. Concerning security, the additional E2E time imposed by using security in hybrid communications was ∼11% in both protocols.

In general, the DLAPP and MFSPV protocols imposed similar additional E2E times. Therefore, choosing one over the other in terms of efficiency is not straightforward. The choice should depend on the specific priorities of the application. As a final remark, the suitability of the presented approach depends on the specific nature of the ITS applications it will incorporate. That is, different ITS use cases have distinct maximum latency demands. The most stringent ones, such as emergency services, require a 50 ms latency, and most others allow up to 100 ms [49,50]. This study's median E2E values do not surpass ∼47 ms, aligning well with the requirements of most use cases. In a worst-case analysis, the E2E time reached around 190 ms. However, it represents an unusual scenario. Therefore, outliers were isolated using the IQR method. In this scenario, the worst-case E2E latency remained at 86 ms, 14% below the maximum latency of 100 ms. Thus, one may conclude that the obtained results align well with the requirements for many use cases. Finally, the architecture proposed for implementation in the Siemens RSU and Unex OBU presents versatile applicability to other commercial equipment. This implementation recommendation serves as a practical approach as it avoids the need for protocol stack and software modifications across diverse equipment from various manufacturers.

For future work, the CA should be developed to allow evaluation of the whole prototype. A security analysis should be performed, including a risk assessment of our implementation using appropriate tools. In addition, experiments should be carried out with more OBUs and RSUs from different manufacturers. This would allow a comparison of the results and thus strengthen the applicability of the solutions. Finally, it is also desirable to carry out evaluations under more stress/overload conditions, including both computational and network aspects, in order to analyse the response of the system to extreme real-world scenarios.

Author Contributions: Conceptualization, R.S., J.S. and N.D.; methodology, R.S., J.S. and N.D.; software, R.S.; validation, R.S., J.S., N.D. and A.S.; formal analysis, R.S., J.S., N.D. and A.S.; investigation, R.S., J.S., N.D. and A.S.; resources, J.S., N.D. and A.S.; writing—original draft preparation, R.S.; writing—review and editing, R.S., J.S., N.D. and A.S.; supervision, J.S. and N.D.; funding acquisition, J.S., N.D. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Cooperative Streets project, under the grants 2018-PT-TM-0099-S from INEA, and by NOVA LINCS (UIDB/04516/2020), LASIGE (UIDB/00408/2020) and INESC-ID Lisboa UIDB/50021/2020 with financial support from FCT—Fundação para a Ciência e a Tecnologia, through national funds.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to being private to the company.

Acknowledgments: For their ongoing support and advice, the authors would like to thank the team at Future Internet Technologies Research Group, Lisbon School of Engineering, Polytechnic of Lisbon, involved in Cooperative Streets project. All Cooperative Streets partners are also thanked by the authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. World Health Organization (WHO). *Global Status Report on Road Safety*; Technical Report; World Health Organization: Geneva, Switzerland, 2018.
2. Vălean, A.I. *EU Road Safety Policy Framework 2021–2030. Next Steps towards "Vision Zero"*; Technical Report; European Comission: Brussels, Belgium, 2020.
3. World Health Organization (WHO). *European Regional Status Report on Road Safety*; Technical Report; World Health Organization: Geneva, Switzerland, 2019.

4. Comission, E. *Final Report of the Single Platform for Open Road Testing and Pre-Deployment of Cooperative, Connected and Automated and Autonomous Mobility Platform (CCAM Platform)*; Technical Report; European Comission: Brussels, Belgium, 2021.
5. Forum, I.T. *New Directions for Data-Driven Transport Safety Corporate Partnership Board Report*; Technical Report; International Transport Forum: Paris, France, 2019.
6. Nwakanma, C.I.; Ahakonye, L.A.C.; Njoku, J.N.; Odirichukwu, J.C.; Okolie, S.A.; Uzondu, C.; Ndubuisi Nweke, C.C.; Kim, D.S. Explainable Artificial Intelligence (XAI) for Intrusion Detection and Mitigation in Intelligent Connected Vehicles: A Review. *Appl. Sci.* **2023**, *13*, 1252. [CrossRef]
7. Hasan, M.; Mohan, S.; Shimizu, T.; Lu, H. Securing Vehicle-to-Everything (V2X) Communication Platforms. *IEEE Trans. Intell. Veh.* **2020**, *5*, 693–713. [CrossRef]
8. Serban, A.C.; Poll, E.; Visser, J. A Security Analysis of the ETSI ITS Vehicular Communications. In *Proceedings of the Computer Safety, Reliability, and Security Conference, Västerås, Sweden, 18 September 2018*; Gallina, B., Skavhaug, A., Schoitsch, E., Bitsch, F., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 365–373. [CrossRef]
9. Yoshizawa, T.; Singelée, D.; Muehlberg, J.T.; Delbruel, S.; Taherkordi, A.; Hughes, D.; Preneel, B. A Survey of Security and Privacy Issues in V2X Communication Systems. *ACM Comput. Surv.* **2023**, *55*, 185. [CrossRef]
10. *ETSI TS 302 665 v1.1.1*; ITS Security—Communications Architecture. European Standard (ETSI): Sophia Antipolis, France, 2010.
11. Noor-A-Rahim, M.; Liu, Z.; Lee, H.; Khyam, M.O.; He, J.; Pesch, D.; Moessner, K.; Saad, W.; Poor, H.V. 6G for Vehicle-to-Everything (V2X) Communications: Enabling Technologies, Challenges, and Opportunities. *Proc. IEEE* **2022**, *110*, 712–734. [CrossRef]
12. Caputo, S.; Mucchi, L.; Umair, M.A.; Meucci, M.; Seminara, M.; Catani, J. The Role of Bidirectional VLC Systems in Low-Latency 6G Vehicular Networks and Comparison with IEEE802.11p and LTE/5G C-V2X. *Sensors* **2022**, *22*, 8618. [CrossRef] [PubMed]
13. Abdel Hakeem, S.A.; Abd El-Gawad, M.A.; Kim, H. A Decentralized Lightweight Authentication and Privacy Protocol for Vehicular Networks. *IEEE Access* **2019**, *7*, 119689–119705. [CrossRef]
14. Alfadhli, S.A.; Lu, S.; Chen, K.; Sebai, M. MFSPV: A Multi-Factor Secured and Lightweight Privacy-Preserving Authentication Scheme for VANETs. *IEEE Access* **2020**, *8*, 142858–142874. [CrossRef]
15. Kurdi, H.; Thayananthan, V. A Multi-Tier MQTT Architecture with Multiple Brokers Based on Fog Computing for Securing Industrial IoT. *Appl. Sci.* **2022**, *12*, 7173. [CrossRef]
16. He, Y.; Wu, B.; Dong, Z.; Wan, J.; Shi, W. Towards C-V2X Enabled Collaborative Autonomous Driving. *IEEE Trans. Veh. Technol.* **2023**, 1–14. [CrossRef]
17. Pham, V.N.; Nguyen, V.; Nguyen, T.D.T.; Huh, E.N. Efficient Edge-Cloud Publish/Subscribe Broker Overlay Networks to Support Latency-Sensitive Wide-Scale IoT Applications. *Symmetry* **2020**, *12*, 3. [CrossRef]
18. Debysern, A. *Road Safety in the EU—European Parliamentary Research Service*; Technical Report; European Parliamentary: Strasbourg, France, 2019.
19. Festag, A. Cooperative intelligent transport systems standards in europe. *IEEE Commun. Mag.* **2014**, *52*, 166–172. [CrossRef]
20. Santa, J.; Pereñiguez, F.; Moragón, A.; Skarmeta, A.F. Experimental evaluation of CAM and DENM messaging services in vehicular communications. *Transp. Res. Part C Emerg. Technol.* **2014**, *46*, 98–120. [CrossRef]
21. *ETSI 302 637-2 v1.4.1*; ITS Vehicular Communications Basic Set of Applications Part 2: Specification of Cooperative Awareness Basic Service. European Standard (ETSI): Sophia Antipolis, France, 2019.
22. *ETSI 302 637-3 v1.3.1*; ITS Vehicular Communications Basic Set of Applications Part 3: Specification of Decentralised Environmental Notification Basic Service. European Standard (ETSI): Sophia Antipolis, France, 2019.
23. *ETSI TS 103 301 v1.3.1*; ITS Basic Set of Applications—Facilities Layer Protocols and Communication Requirements for Infrastructure Services. European Standard (ETSI): Sophia Antipolis, France, 2020.
24. Du, W. *Computer & Internet Security: A Hands-On Approach*; CreateSpace Independent Publishing Platform: Scotts Valley, CA, USA, 2017.
25. *ETSI TS 102 940 v2.1.1*; ITS Security—ITS Communications Security Architecture and Security Management. European Standard (ETSI): Sophia Antipolis, France, 2021.
26. Stotz, J.P.; Kargl, F.; Petit, J. *Security Requirements of Vehicle Security Architecture*; Technical Report; PRESERVE Project: European Commission, Brussels, Belgium, 2011.
27. Wasef, A.; Lu, R.; Lin, X.; Shen, X. Complementing public key infrastructure to secure vehicular ad hoc networks (Security and Privacy in Emerging Wireless Networks). *IEEE Wirel. Commun.* **2010**, *17*, 22–28. [CrossRef]
28. Gonçalves, M.; Datia, N.; Serrador, A. A safety perspective for soft mobility in the ITS ecosystem. In Proceedings of the Atas do 13o Simpósio de Informática (Inforum 22), Guarda, Portugal, 8–9 September 2022; pp. 330–341.
29. van Dam, J.F.; Bißmeyer, N.; Zimmermann, C.; Eckert, K. Security in Hybrid Vehicular Communication Based on ITS G5, LTE-V, and Mobile Edge Computing. In *Proceedings of the AmE 2018 Automotive meets Electronics, 9th GMM-Symposium (Fahrerassistenzsysteme 2018), Dortmund, Germany, 7–8 March 2018*; Bertram, T., Ed.; Springer: Wiesbaden, Germany, 2019; pp. 80–91. [CrossRef]
30. Scholliers, J.; Jutila, M.; Valta, M.; Kauvo, K.; Virtanen, A.; Pyykönen, P. Co-operative Traffic Solutions for Hybrid Communication Environments. *Transp. Res. Procedia* **2016**, *14*, 4542–4551. [CrossRef]
31. Wang, F.; Xu, Y.; Zhang, H.; Zhang, Y.; Zhu, L. 2FLIP: A Two-Factor Lightweight Privacy-Preserving Authentication Scheme for VANET. *IEEE Trans. Veh. Technol.* **2016**, *65*, 896–911. [CrossRef]

32. *ETSI TR 102893 v1.2.1*; ITS Security—Threat, Vulnerability and Risk Analysis (TVRA). European Standard (ETSI): Sophia Antipolis, France, 2017.
33. *ETSI TS 102 941 v1.4.1*; ITS Security—Trust and Privacy Management. European Standard (ETSI): Sophia Antipolis, France, 2021.
34. *ETSI TS 102943 v1.1.1*; ITS Security—Confidentiality Services. European Standard (ETSI): Sophia Antipolis, France, 2012.
35. *ETSI TS 103 097 v2.1.1*; ITS Security—Security Header and Certificate Formats. European Standard (ETSI): Sophia Antipolis, France, 2021.
36. Hiller, M. Key Derivation with Physical Unclonable Functions. Ph.D. Thesis, Technische Universität, München, Germany, 2016.
37. Marias-i Parella, J.; Pino, A.; Cordero, B.; Casademont, J.; Carmona-Cejudo, E.; Vázquez-Gallego, F. Demo: Interoperability between Cellular and V2X Networks (802.11p/LTE-PC5) under a Cloud Native Edge Scenario. In Proceedings of the IEEE INFOCOM 2023—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hoboken, NJ, USA, 20–23 May 2023; pp. 1–2. [CrossRef]
38. Bréhon-Grataloup, L.; Kacimi, R.; Beylot, A.L. Mobile edge computing for V2X architectures and applications: A survey. *Comput. Netw.* **2022**, *206*, 108797. [CrossRef]
39. Pu, C.; Ding, X.; Wang, P.; Xie, S.; Chen, J. Semantic Interconnection Scheme for Industrial Wireless Sensor Networks and Industrial Internet with OPC UA Pub/Sub. *Sensors* **2022**, *22*, 7762. [CrossRef]
40. Mirampalli, S.; Wankar, R.; Srirama, S.N. Evaluating NiFi and MQTT based serverless data pipelines in fog computing environments. *Future Gener. Comput. Syst.* **2024**, *150*, 341–353. [CrossRef]
41. Sommer, P.; Schellroth, F.; Fischer, M.; Schlechtendahl, J. Message-oriented Middleware for Industrial Production Systems. In Proceedings of the 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), Munich, Germany, 20–24 August 2018; pp. 1217–1223. [CrossRef]
42. Raptis, T.P.; Cicconetti, C.; Falelakis, M.; Kalogiannis, G.; Kanellos, T.; Lobo, T.P. Engineering Resource-Efficient Data Management for Smart Cities with Apache Kafka. *Future Internet* **2023**, *15*, 43. [CrossRef]
43. Yongguo, J.; Qiang, L.; Changshuai, Q.; Jian, S.; Qianqian, L. Message-oriented Middleware: A Review. In Proceedings of the 2019 5th International Conference on Big Data Computing and Communications (BIGCOM), Qingdao, China, 9–11 August 2019; pp. 88–97. [CrossRef]
44. Machovec, F. *ESCoS Roadside Unit ITS XFER Gateway Interface Specification*; For RSU version 1.2.2; Siemens: München, Germany, 2019.
45. hmac_sha256: HMAC-SHA256 Implementation. 2022. Available online: https://github.com/h5p9sl/hmac_sha256 (accessed on 6 November 2023).
46. Ohnheiser, J. *ESCoS Roadside Unit User Manual ETSI*; For RSU version 1.4.25; Siemens: München, Germany, 2019.
47. Wadatkar, P.V.; Garroppo, R.G.; Nencioni, G. 5G-MEC Testbeds for V2X Applications. *Future Internet* **2023**, *15*, 175. [CrossRef]
48. Bustamante-Bello, R.; García-Barba, A.; Arce-Saenz, L.A.; Curiel-Ramirez, L.A.; Izquierdo-Reyes, J.; Ramirez-Mendoza, R.A. Visualizing Street Pavement Anomalies through Fog Computing V2I Networks and Machine Learning. *Sensors* **2022**, *22*, 456. [CrossRef] [PubMed]
49. 5G-MOBIX. *Deliverable 2.5: Initial Evaluation KPIs and Metrics*; Technical Report; 5G-MOBIX, European Commission, Brussels, Belgium, 2019.
50. Castañeda, O.; Baños, J.; Garrido, A.J.; Cárdenas, C.; Mendes, C.; Serrador, A.; Cota, N.; Datia, N.; Cruz, N. Latency Assessment for CAM Services over 5G. In Proceedings of the IEEE 5G Virtual Summit for Connected and Automated Mobility, Brussels, Belgium, 11–12 May 2021.