# An RT-cloud solution towards security in Vehicular platooning systems

Rui Rafael, Harrison Kurunathan, Eduardo Tovar
*department of electrical and computer engineering*
*CISTER/ISEP, Porto, Portugal*
(1181357, jhk, emt)@isep.ipp.pt

***Abstract*—Vehicular platooning involves a group of vehicles traveling closely together, often in a convoy-like formation, which can lead to increased safety and fuel efficiency. However, this type of system also presents unique secure communication challenges. The data exchanged between platooning vehicles must be protected from interception and manipulation, while being readily available for all the vehicles within the platoon. The use of encryption schemes and secure communication protocols can help ensure data integrity. Moreover, the use of cloud services can assist in ensuring the availability and reliability of the communications between vehicles. In this paper, we present an RT-cloud architecture that utilizes state-of-the-art cryptography and cloud services for secure and reliable vehicular platooning communications. We provide an analysis of the impact of security and cloud schemes on the quality and stability of vehicle platoons.**

***Index Terms*—platooning, secure vehicular communication, security architecture, cloud, real-time**

## I. Introduction

The possibility of featuring cooperation between vehicles in an intelligent transportation system (ITS) has led to the emergence of connected vehicles, also called as platoons. Vehicle Platooning is a method for driving a group of vehicles together in transportation (Figure 1). A vehicular platoon consists of a leader vehicle and a number of following autonomous vehicles, where each vehicle maintains a small distance to its preceding vehicle [1]. With the emergence in technologies like smart cars, artificial intelligence can possibly help in taking complex take decisions like newer vehicles joining and leaving platoons [2]. Adding to the paradigm of being safer and efficient, these platoons also aim at improving being environmentally-friendly by reducing CO2 emissions [3].

Current standards of vehicular communications ETSI-ITS and IEEE 1609.2 [4] enable the vehicles in the platoon to exchange several types of data to enable a smart traffic system. This data includes vehicular-monitoring data such as the distance between the vehicles, speed, and acceleration which then can be used in the efficient control of the vehicles. It is of utmost importance that security in vehicle platooning is a vital aspect for that impacts the safety of the underlying platoon. Any compromise on this safety can result in a range of consequences or attacks performed on a vehicle platoon. These consequences can range from redirection of the vehicles, to causing accidents on the road and to exposition of confidential data. Some of the common attacks that are discussed in the literature include, Message falsification attack [5], Collision Message spoofing , Collision Message replay, control tampering [6], Collision DoS (jamming), Dissolved platoon and System tampering Collision [7]. Hence, there is a dire need to establish strong security schemes to ensure the safety of the platooning system. Some of these attacks and impacts studied from the literature are tabulated in Table I.

This work is a result of project FLOYD (POCI-01-0247-FEDER-045912), cofinanced by the European Regional Development Fund (ERDF) through the Operational Program for Competitiveness and Internationalisation (COMPETE 2020) and by the Portuguese Foundation for Science and Technology (FCT) under the CMU Portugal programme.

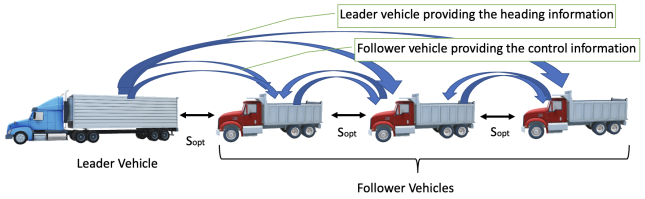


Fig. 1. Example of a platoon travelling with a safety distance between them and communicating wirelessly their respective headings and control information

TABLE I
Attacks and their respective impact on the platoons

| Paper | Type of the attack | Impact on the platoon |
|---|---|---|
| [6] | Collision induction attack | Collision |
| | Reduced headway attack | Decreased performance |
| | Joining without radar | Decreased performance |
| | Mis-report attack | Decreased stability |
| [7] | Destabilization attack | Decreased stability |
| | Platoon control taken attack | Dissolved platoon |
| [5] | Message falsification | collision |
| | Message spoofing | collision |
| | Message replay | collision |
| | DoS (jamming) | Dissolved platoon |
| | System tampering | collision |

Cloud-enabled platooning refers to the technology where multiple vehicles travel closely in a coordinated manner by utilizing the data transmitted over a cloud-based network. For instance, in V2V communication the vehicles share metrics such as speed, location, and acceleration to the neighboring vehicles and in V2I and I2V communication the vehicles to communicate their metrics with infrastructure such as traffic signals and road sensors and the infrastructure communicates information of the incoming lane. Cloud computing enables

hosted services, such as software, hardware, and storage, over the Internet. Cloud enabled architectures provides us with the benefits of rapid deployment, flexibility, low up-front costs, and scalability. A real-time efficient cloud enabled infrastructure that can assure some levels of determinism can enable critical application domains like platooning.

In this paper, we present a secure real time cloud platooning architecture that utilizes state-of-the-art cryptography and cloud hosted services in order to achieve a safe, reliable and available communication medium between the vehicles in a platoon. State-of-the-art cryptography is used on all communications within the platoon, ensuring a safe, authenticated, confidential and reliable communication medium that prevents tampering and eavesdropping on the platoon. The cloud hosted services provide a robust and scalable infrastructure that can handle the high demand of platooning systems while having geographically distributed resources, enabling the system to remain highly available and reliable even in the event of failures in individual components. Keeping in context of the performance requisites and real-time requirements of vehicular platooning, we analyze the performance of our proposed platooning architecture in order to determine the overall impact on the stability of vehicle platooning.

The contributions of this paper are as follows :

- A real-time cloud architecture to facilitate the security and communication challenges of a vehicular platooning system.
- Latency analysis to analyse the impact of state-of-the-art security algorithms and cloud services in the stability of vehicular platooning.
- Key bit randomness analysis to show the randomness and the strength of the keys that are generated during every platooning session.

The rest of the paper is organised as follows, In Section II, we present the state-of-the art security and cloud enabled architectures in platooning systems and compare them with our architecture. In Section III, we present our system model where we discuss the security flow and the components of our architecture. In section IV, we present some results of our real time cloud model and finally conclude this paper with some discussions and future scopes.

## II. Literature Survey

The RT-cloud solution presented in this paper focuses on enhancing the security of vehicular platooning through the use of state-of-the-art cryptography and cloud services. Our framework ensures that the communication between vehicles is secure and protected from unauthorized access or tampering. We utilize two cloud-hosted services to further reinforce the security of the platoon while also ensuring that the system is easily scalable, reliable, and available. The usage of cloud resources enables us to quickly scale the system to meet changing demands, making it a flexible solution for next-generation transportation systems. We present the state of the art from two fronts, firstly, we provide some background towards cloud enabled platooning where cloud based architectures are used in aiding control and communication in platooning. Then, we provide background towards security in platooning that will serve as a baseline towards the security methods we have implemented on our architecture.

### A. Cloud enabled platooning

A cloud-enabled network provides a centralized platform for collecting, analyzing, and sharing data amongst the vehicles, and aid them operate more efficiently and safely. The collected data also can be analyzed in real-time to optimize the platoon's speed, distance between vehicles, and other factors. Researchers in [8], [9] present a cloud enabled platooning framework where vehicles report their speed and position periodically, and the MEC (Multi-Access edge Computing) runs a platoon formation algorithm to form platoons and define the acceleration of each vehicle to maintain formation with the necessary safety distance. The proposed real-time cloud architecture for vehicle platooning leverages the benefits of a cloud-enabled network, as highlighted in previous studies, to provide a centralized platform for efficient data collection, analysis, and sharing among vehicles, leading to improved safety and optimized platoon performance.

Researchers in [10] present a truck platooning that is facilitated by secure publish/subscribe system based on smart contract in autonomous vehicles. The truck platoons in this architecture are utilized as brokers of the publish/subscribe system for realizing an efficient and secure publish/subscribe system. The cloud server they adopt in their architecture acts as the authority platform that can show the publish information to the autonomous driving vehicles (ADV) and help the subscriber to choose an optimal broker to obtain the content. Our proposed solution for vehicle platooning also utilizes a secure publish/subscribe model to efficiently distribute information among vehicles.

Several frameworks [11], [12] use roadside units (RSU) to enable fleet management by sending diagnostics to the cloud, and then identifying issues on individual vehicles or the entire platoon. A cloud enabled network has been used for monitoring message exchange, performing data analytics and measuring the network Key Performance Indicators (KPIs). The proposed architecture for vehicle platooning also leverages the advantages of a cloud-enabled network by monitoring of message exchange and measurement of KPIs for the optimization of platoon performance.

The implementation in this paper is in line with the existing research on cloud-enabled platooning, leveraging several of its positive aspects such as collecting, analyzing, and sharing data amongst vehicles, efficient and secure distribution of information between the platoon and monitoring of message exchanges. By utilizing cloud services, this implementation can monitor the safety and stability of the platoon, as well as measure the platoon's performance metrics through KPIs. Additionally, it also allows for the cloud to interact with the platoon in emergency scenarios and dynamically scale the system up or down, according to the demand exerted on it.

### B. Security in platooning

In order to ensure security of the messages in V2X communications, asymmetric cryptography is used in conjunction with a public key infrastructure (PKI) for managing security credentials [13], [14]. In PKI, secure exchange of messages over the network is facilitated by an asymmetric key pair and a certificate. The certificate contains the public key with vehicle communication specific attributes such as ID and is signed by the key issuing. In our paper, we present how we use asymmetric cryptography/PKI as a way to ensure the authentication and integrity of the messages that are transmitted between the vehicles and other parties involved in vehicle platooning.

There also has been several standardization efforts to ensure security in V2X communications. For instance, IEEE has introduced V2X communications by the WAVE (wireless access in vehicular environments) protocol [15]. ETSI has also developed standards for V2X communications called the ETSI-ITS (ETSI intelligent transport system) [16]. This standard includes an overall architecture, a protocol stack as well as security requirements and mechanisms. The proposed model in this paper adheres to the standardization of security mechanisms related but not limited to management of trust within the communications and of all parties involved in the platooning, protection of sensitive data through data encryption, active measures against known attacks such as message replay attacks and usage of established cryptographic systems such as Advanced Encryption Standard(AES) with a key size of 256 bits.

Both ETSI ITS (Europe) and IEEE 1609.2 (U.S.) standards recommend the usage of PKI for providing security in V2X safety applications. Both of these standards mandate the usage of Elliptic Curve Digital Signature Algorithm (ECDSA) for faster authentication and non-repudiation at the cost of computationally expensive operations. There are several PKI schemes like the PRESERVE [17], [18] that proposed the deployment of a PKI-based V2X security system based on ETSI ITS architecture. This implementation is compatible with IEEE 1609 and ETSI certificate formats, and both infrastructure and vehicle certificates are based on ECC-256 keys. In our architecture, we will be analyzing the steps implemented to ensure the security and safety of the communications, as well as measure the performance impact of the defined steps in vehicle platooning.

### C. Novelty of this work

This paper presents a real-time cloud architecture for secure vehicle platooning that leverages the benefits of cloud computing to enable secure communication between vehicles, while also ensuring an available and reliable system that facilitates communication between vehicles. Our approach addresses the secure communication challenges related to platooning, and we evaluate the impact of the proposed real-time cloud architecture on the stability of the platoon.

In addition to analyzing the randomness of AES 256 bit cryptographic keys using the NIST statistical test suite, we provide a comprehensive time analysis of our architecture. This includes measuring the execution time of digital signature creation, comparing communication delay of the cloud-based system to a non-cloud system, measuring the communication delay on emergency message broadcast between the platoon, and assessing if the communication delay in the proposed model is sufficient to handle the stability and safety of the platoon.

Furthermore, our system model utilizes the cloud to deploy the services required for our architecture and utilizes cloud services to dynamically scale the infrastructure up and down as needed and required by the demand of vehicle platooning. This not only ensures the availability and reliability of the system, but also provides a cost-effective solution. Our work makes significant contributions to the field of vehicle platooning by presenting an architecture that addresses the challenges of secure communications through the utilization of cloud computing and advanced cryptography, while also providing a dynamic, scalable, and cost-effective infrastructure for vehicle platooning.

## III. RT CLOUD FRAMEWORK FOR SECURE PLATOONING

The proposed vehicle platooning system model involves a Platoon Session Manager(PSM) that is a cloud-hosted REST API on Amazon Web Services(AWS) cloud on an EC2 Instance and managed by AWS Elastic Beanstalk. To join a vehicle platoon that is headed to a specific destination(Figure 2: vehicles in the middle), a car makes an HTTPS request to the PSM. Before allowing the car to join the session, the PSM validates the car's certificate and checks if the root certificate authority(CA) that signed the car's certificate is known and trusted. Similarly, the car also validates the PSM certificate by checking if the root CA that signed the certificate is known and trusted. This mutual certificate validation ensures secure and authenticated communication between the car and the PSM.

Once the mutual certificate validation is successfully performed, the car can specify its request to join the session through a POST operation to the path api/Session/JoinSession in JavaScript Object Notation (JSON) format with a request body that contains the session unique identifier the car wants to join, the public key information of the car to be used during the session, and the type of public key (RSA or ECC).

After receiving the POST request body(Figure 2: right side of PSM), the PSM generates a unique identifier for the car and aggregates information about the session, including the information about the cars that are already in the platoon (public key's of the cars and unique identifier's), the MQTT-SN Broker address and the MQTT-SN topic identifier for the current running session, the AES 256 bits symmetric key and initialization vector to be used for encryption/decryption of messages, and the hashing algorithm to be used. The symmetric key/initialization vector is generated by the PSM when the session starts, and it's generated internally, which means that no malicious attacker can interfere with the generation process. The symmetric key/initialization vector generation does not take into account any platoon related attribute, which means

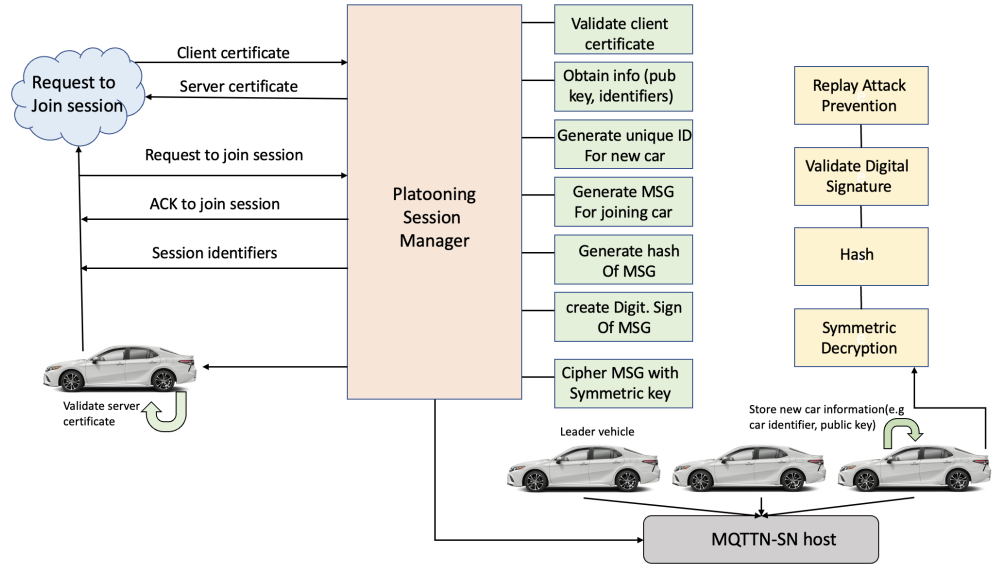


Fig. 2. System model of the proposed secure platooning architecture portraying the security steps in the event of a new vehicle (left end) trying to join the platooning session

that there is no possibility for a malicious attacker to have any platoon related information that would create a heuristic that could weaken the randomness of the key/initialization vector. The generation uses a cryptographically secure pseudorandom number generator (CSPRNG).

The PSM then creates a new message of type 'NewCarOnSession' and runs the message through a security pipeline. The security pipeline for sending a message contains several stages, including the generation of a secure message that contains the 'NewCarOnSession' message, the serialization of the secure message using the MessagePack binary serialization format, the hashing of the serialized message using the hashing algorithm of the session, the creation of a digital signature using a private key, and the creation of a new secure packet containing the serialized secure message and the digital signature of the packet. Finally, the content of the secure packet is encrypted using the AES 256 bits session key. After that, the message is published on the MQTT Broker and on the topic of the session.

When the other cars that are already on the session receive the message, they run the received message through the security pipeline(Figure 2: last vehicle on the right side) that has the following stages: first, it decrypts the message with the session symmetric key/initialization vector, then it deserializes the decrypted content, then it checks the unique identifier present on the packet and verifies if that unique identifier belongs to the current session. If it does, the public key corresponding to the entity that has that unique identifier is grabbed. Then, a hash is generated out of the received message, and the digital signature of the packet is verified using the public key. The message is also verified to ensure that it has not been replayed by using the hash of the message. Finally, the message is deserialized, interpreted, and the information about the new car that is joining is added to the current in-memory information of the session.

After this has been done, the PSM responds back to the car that requested to join the session with the entire information about it, which then allows the car to join the platoon.

While on the session, the cars communicate with each other through an MQTT-SN broker that is hosted on the AWS cloud in an EC2 instance and managed by AWS Elastic Beanstalk. The MQTT-SN broker acts as a communication hub, allowing cars to send and receive messages between each other. Every message that is sent or received by the cars has to go through the security pipeline previously described.

The system model aforementioned allows for a number of security features that ensure the integrity, confidentiality, and availability of the system. By requiring authentication before allowing a car to join a session, the system ensures that only valid cars can participate, preventing unauthorized access to the system. In addition, by allowing only cars that are already part of the session to communicate with each other, the system provides a level of confidentiality that prevents eavesdropping on communication between cars. Furthermore, the system's design ensures that the system is available by allowing cars to communicate with each other in real-time, enabling the system to be responsive to changing conditions on the road. The system model also uses cloud-deployed services managed by AWS Elastic Beanstalk. This management service from AWS can scale up or down the infrastructure as needed, based on the high load within the PSM and the platooning system. This ensures that the system can handle the necessary processing and communication requirements while maintaining high availability and reliability. By leveraging AWS Elastic Beanstalk, the system can seamlessly adjust to the changing demands, making it a highly scalable and efficient solution.

Overall, the system model provides a secure and reliable framework for connected car technology.

## IV. Use case scenarios

In this section we present two use cases that are vital in any platooning scenario. Firstly, we present the platoon session joining use case, where a new vehicle tries to join the platoon by connecting itself with the platooning session manager. Secondly, the emergency braking use case, where the leader vehicle anticipates a collision and sends an emergency brake message to the entire platoon.

### A. Platoon session joining scenario

The use case of platoon session joining is an important use case of the system model for vehicle platooning. It involves a car requesting to join a platooning session(Figure 3: vehicle on the left), and the PSM validating the car's certificate and ensuring that it is authorized to join the session. Similarly, the car validates the PSM certificate to ensure that it is communicating with a legitimate PSM.

Once the car is authorized to join the session, the PSM emits a message to the other cars already on the session(Figure 3: vehicles on the right) using the security pipeline, providing them with information about the new vehicle that is joining the session. This message includes information such as the car unique identifier and public key. After the message is sent, the PSM responds back to the car that requested to join with information about the session, such as the session symmetric key and the other cars already on the session. This information allows the car to communicate securely with the other cars on the session.

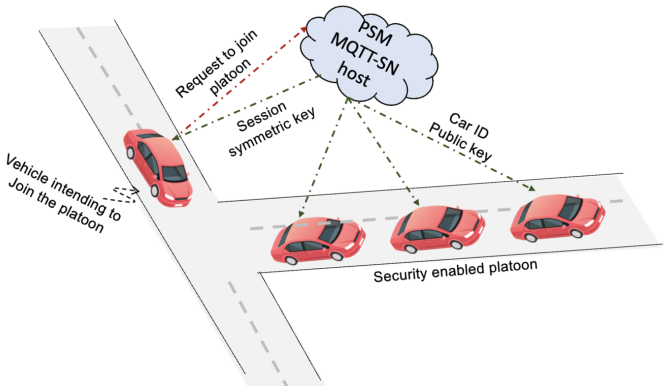


Fig. 3. Vehicle on the left requests the PSM to join the session, PSM emits a message to the platoon that a new car is joining and the requesting vehicle receives information about the platoon session

When a new vehicle joins in a platoon it is vital to coordinate the actions and maintain safe inter-vehicle distances, therefore, they must communicate with each other. It must be noted that several delays can affect the stability of the platoon. The communication delay between vehicle $a$ and $b$ can be modeled as:

$$Dt_{a,b} = \delta + \rho D_{a,b} + \phi \quad (1)$$

where, $\delta$ is the communication delay between vehicle a and vehicle b, $\rho$ is the delay due to processing and transmission, $D_{a,b}$ is the the distance between vehicle a and vehicle b and $\phi$ is the random delay due to fluctuations in the communication channel.

The value of $\rho$ has to be defined in order to ensure a minimal level of determinism for enabling this time critical application. This value also encompasses several overheads namely the security overhead (encryption, hash, digital signature, decryption), the processing overhead and acknowledgement overheads. In this work, we provide analysis towards these security overheads pertaining to vehicular platooning.

This use case is important for ensuring that the platooning system remains secure and efficient while allowing new cars to join a running session. By validating certificates and using a secure communication pipeline, the system model is able to ensure that only authorized cars can join a session and that all communication within the session is secure.

### B. Emergency braking

The emergency braking use case is a crucial one that ensures the stability of vehicles in a platooning session. It involves a scenario where a car detects a road situation that requires an immediate emergency brake(Figure 4: vehicle on the left). Once this is detected, the car generates a new message and publishes it to MQTT-SN and to the current platooning session(Figure 5: vehicles on the right). This message contains critical information about the emergency brake and is sent securely using the security pipeline that includes hashing, digital signature creation/validation, and symmetric encryption/decryption to ensure authenticity, confidentiality, and prevention of replay attacks.

The message is then received by all the other cars in the platooning session, alerting them of the emergency brake situation. This ensures that all the cars can take appropriate and timely action's to avoid any accidents. The use of the security pipeline guarantees that the message is received and processed in a secure manner, minimizing the risk of any malicious actors tampering with or intercepting the message.

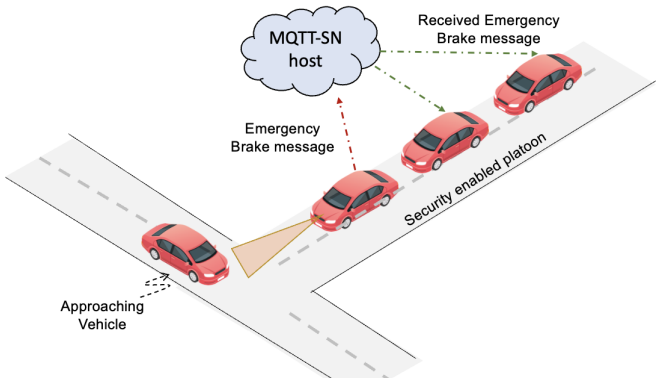


Fig. 4. Leader vehicle detects an emergency situation and notifies the other vehicles about an emergency braking

Apart from averting catastrophic events like accidents, timely reception of emergency messages and actuation aid in ensuring the stability of the platoon. There is stability criterion for a vehicular platoon which must be satisfied for the platoon to be considered stable. This stability criterion is defined based on the eigenvalues and eigenvectors of the corresponding linearised platoon model. This defines the small perturbations around the steady-state behavior of the platoon. The eigenvalues represent the growth rates or decay rates of these perturbations, and the eigenvectors represent the corresponding modes of vibration or oscillation.

The stability criterion for a n-vehicle platoon can be given by:

$$\lambda_n = Re(S_n) > 0 \tag{2}$$

where, $\lambda_n$ is the eigenvalue of the n-th platoon, $S_n$ is the corresponding eigenvector, and $Re()$ denotes the real part of the complex number. Several works [19], [20] in the literature use the foundation of the Gersgorin Disk Criterion to validate the stability of the platoon. One of the recent works [21] studies the influence of topology and delay on their internal stability. The platoon stability was investigated by measuring the platoon internal stability index, based eigenvalues of the topology. The worst case delay values generated from this work have been used as a baseline to study and validate the efficiency of our proposed framework in the emergency brake scenario.

## V. Performance Analysis

This section discusses in detail the performance parameters and test scenarios that are aimed at evaluating and validating the efficiency of the proposed system.

- Key bit randomness: This performance metric shows the randomness amongst the key bits that are generated in every session for AES 256-bit keys. Higher level of randomness helps in reducing the possibility of cracking the generated keys.
- Digital signature execution time: This performance metric shows how much time it takes to create a digital signature. This showcases the execution time exclusively to implement a security algorithm
- Emergency braking latency: This scenario measures the time it takes for an emergency braking message to be transmitted from the leader vehicle to the platoon, and for the message to be received by a vehicle in the session. This measurement is taken for both cloud and edge deployments of the system. The results will provide insights into the time delay of transmitting emergency messages in different deployment scenarios, which is crucial for ensuring the stability and security of the platoon.
- Session joining latency: This is a performance analysis that measures the time it takes for a new vehicle to join a platoon session, comparing the system's deployment on the cloud versus on the edge. The PSM is responsible for handling the joining process, and the measurement is taken from the moment a vehicle requests to join until an existing vehicle in the session receives a message about the new arrival.
- Cloud impact on stability of vehicle platoons: This is a scenario to evaluate the impact of a cloud deployment on our system model and on the platoon stability. We compared our results with the acceptable time delay thresholds proposed in previous studies. This scenario provides insights into the impact that the proposed system model has on the overall integrity of vehicle platooning.

### A. Key bit randomness

To verify the randomness in the keys generated, we used the standard randomness test suite from NIST [22]. The NIST statistical suite provides a set of 16 tests to verify the randomness, among them we use 6 tests to verify the effectiveness of our generated keys. The P-value represents the probability that a perfectly tuned random number generator would have produced a sequence less random than the input sequence that is tested. To pass the test, all p-values must be greater than 0.01. Some of the tests of this suite needed larger input bit stream, hence we chose the tests that are appropriate for our dataset.

| NIST Tests | K-1 | K-2 | K-3 | K-4 | K-5 |
|---|---|---|---|---|---|
| Frequency Test | 0,38 | 0,71 | 0,62 | 0,38 | 0,05 |
| Block Frequency Test | 0,64 | 0,92 | 0,78 | 0,07 | 0,13 |
| Cumulative SUMS (FWD) | 0,42 | 0,80 | 0,91 | 0,21 | 0,09 |
| Cumulative SUMS (REV) | 0,18 | 0,97 | 0,75 | 0,75 | 0,09 |
| Runs | 0,11 | 0,80 | 0,11 | 0,19 | 0,37 |
| FFT | 0,42 | 0,82 | 0,73 | 0,14 | 0,73 |

TABLE II
NIST P-VALUES FOR 256-BIT KEYS GENERATED FOR AES ALGORITHM HAVING SATISFIED THE THRESHOLD OF 0.01 TO PROVE RANDOMNESS

The aforementioned table presents the p-values of 256-bits keys generated for AES algorithm for 5 keys(K-1, K-2, K-3, K-4, K-5). It is evident that the keys generated during every session in the platoon using our architecture have p values that are larger than the threshold of 0.01 to pass the test. This specific threshold indicates that the generated secret bit streams are completely random, with a confidence of 99%. This large randomness increases the complexity of cracking the keys by any eavesdropper.

### B. Digital signature execution time

The aim of this experiment is to see what will be the exact time to establish a digital signature. For this experiment we compare the algorithms ECC and RSA. Based on the results obtained in Figure 5, we can see that on average, the creation of a digital signature with ECC 256 bits is 35,88% faster than RSA 3072 bits.

This behavior is because of the difference in key sizes that is required to obtain the same security level. In order to obtain the same security level with a digital signature created by RSA with 3072 bits, ECC requires a 256 bits key, which leads to a less computational expensive operation [23]. In

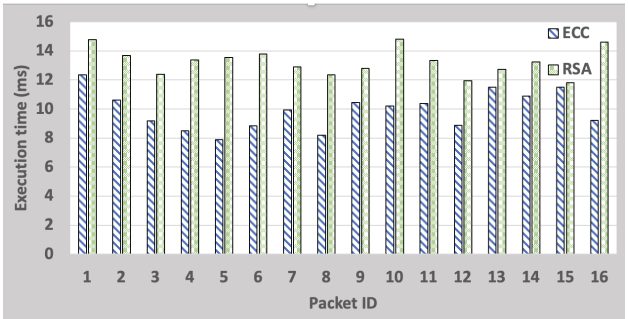


Fig. 5. Digital signature execution time for creating a digital signature with ECC 256 bits and RSA 3072 bits for a packet size of 468 bytes

our experiments, we have found that RSA is slightly faster in digital signature verification than ECC, but the difference in digital signature verification is much smaller compared to the performance gap in digital signature creation, indicating that ECC is the superior choice in terms of digital signature operations. Overall, this variation in the execution time will have a direct impact in a real-time platooning scenario.

### C. Emergency braking latency

This performance analysis measures emergency braking message propagation time from the leader vehicle to a platoon vehicle. The results compare system performance in cloud and edge deployments, highlighting the impact of network proximity and architecture on system latency.

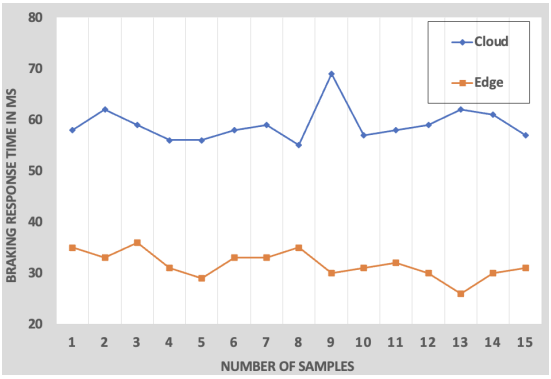


Fig. 6. Emergency braking message propagation time from leader vehicle to platoon vehicle with cloud deployment and edge deployment

The results of this performance analysis demonstrate that the emergency braking message propagation is significantly faster when deployed on edge compared to when deployed on the cloud. The 15 samples taken show an average time of 59.0 milliseconds on the cloud and 31.67 milliseconds on edge, indicating that emergency braking on the edge is 86.53% faster than on the cloud. These results happen due to the difference in network architecture and proximity, as the edge deployment leverages a more geographically proximate network connection. In contrast, the cloud deployment relies on an internet-based connection that is located farther away from the platoon, resulting in a longer latency for emergency braking message propagation.

### D. Session joining latency

This analysis aims to evaluate the latency of the process for a car to join a platoon session with a cloud and edge deployment approach.

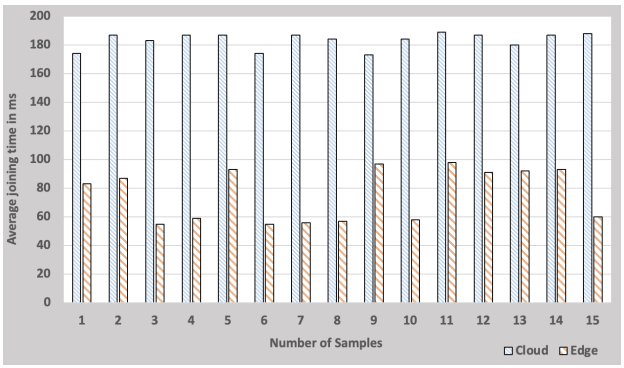


Fig. 7. Cloud and edge deployment time duration from vehicle request to join session, until platoon car receives notification of new car joining

The results of this performance analysis indicate that the deployment of a vehicle platooning system on the edge has a faster joining time than the cloud deployment. The analysis was based on 15 samples, which revealed that the average joining time on the cloud was 183,4 milliseconds, while on the edge it was 75,6 milliseconds. This means that, on average, car joining time in the cloud is 121,3% more than car joining time on the edge.

The performance analysis conducted on the system's emergency braking and session joining latency scenarios reveals that there is a significant gap between cloud and edge deployments in the latter scenario. The reason for this is that the session joining latency involves multiple steps, including an HTTPS request to the PSM, a publication of a message to an MQTT-SN broker by the PSM, and the subsequent receiving of that message by another car in the platoon. This multistep process contributes to a higher latency when deploying the system on the cloud compared to the edge. In contrast, the emergency braking latency scenario only involves the publication of a single MQTT-SN message and its subsequent receiving, which results in a smaller gap between the cloud and edge deployments.

### E. Cloud impact on stability of vehicle platoons

Several works provide numerical analysis towards platoon stability and acceptable latencies in case of emergency brakes [24]. In this test we utilize the numerical analysis of stability in platooning developed in [21] and compare them with the average communication delay resulting from our RT-cloud enabled platooning architecture. In order for the platoon internal stability to be achieved, $\lambda_n$ must be at all times greater than 0. The results showed that the platoon becomes less stable as both delay and eigenvalue increase, and that a lower communication delay is required for higher eigenvalue.

The graph in Figure 8 developed from the stability criterion in [21] depicts the relationship between the maximum communication delay and the eigen values for platoon internal stability, with the $\lambda_n$ threshold represented by a red line. The Y-axis ranges from 0 to 0.5 seconds, with intervals of 0.1 seconds, while the X-axis ranges from 0 to 5 with intervals of 1. By plotting an asymptotic line on this previous graph, that represents the average communication delay on the cloud for the emergency braking scenario, we can determine that the proposed system model is safe, reliable and functional when applied to vehicle platooning.

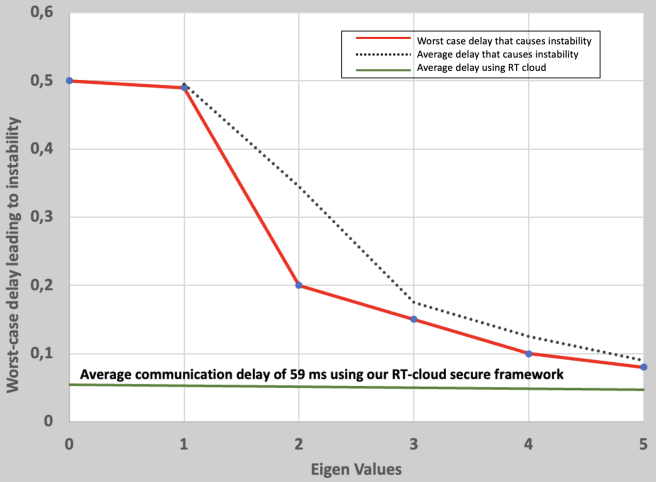


Fig. 8. Comparing the average communication delay of or RT-cloud secure model with the numerical analysis of the maximum Communication Delay that leads to instability of Platoon developed in [21]

Based on the analysis conducted in the scenario emergency braking latency which showed an average communication delay between vehicles of 59 milliseconds(0.059 seconds) on the cloud, it can be concluded that the system model is safe and functional when applied to vehicle platooning. This is supported by the fact that when plotted on the graph of the previous study, the communication delay required for platoon stability always exceeds the communication delay observed in the cloud deployment.

String stability is a crucial aspect of platooning, as it ensures that vehicles maintain a safe distance and travel in a coordinated manner. In order to ensure string stability in vehicle platooning, communication delay must be as minimal as possible. According to the studies in [25], string stability can be achieved when the preceding vehicle communication delay is between an upper bound and a lower bound of 1.2 seconds and 80 milliseconds. The analysis conducted in the emergency braking latency scenario shows that the average communication delay between vehicles on the cloud is only 59 milliseconds. This finding suggests that the proposed system model is safe and effectively maintains the string stability of the platoon.

Comparing the results of the session joining latency scenario to those of previous studies is not appropriate, as the former focuses on the joining process itself, which does not involve the exchange of control messages between vehicles. In contrast, the presented studies focus on delay of messages that are transmitted between vehicles in a platoon while they are travelling, which is critical for ensuring the stability of the platoon on the road. Therefore, the session joining latency scenario should be seen as a separate scenario that reflects the delay incurred by a vehicle joining an existing platoon, rather than as a benchmark for the platoon's overall performance during travel.

## VI. Conclusion

In this paper, we present a real-time cloud architecture to facilitate security in a platooning system. We used the NIST statistical tool to show the randomness of the keys generated using our security architecture. We measured the impact of the architecture in the stability of the platoon. Furthermore, we were able to concur that a cloud deployment as a bigger communication delay in the platoon than an edge deployment, but the overall impact did not affect the stability of the platoon, demonstrating that it is a fitting architecture to ensure a reliable and safe system. As a future scope, we intend to develop a dynamic security algorithm switching architecture that can change the digital signature algorithms on real time based on the security overheads.

## References


[1] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa, "Overview of platooning systems," in *Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)*, 2012.

[2] J. Rezgui, É. Gagné, G. Blain, O. St-Pierre, and M. Harvey, "Platooning of autonomous vehicles with artificial intelligence v2i communications and navigation algorithm," in *2020 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, 2020, pp. 1–6.

[3] L. A. Maglaras, P. Basaras, and D. Katsaros, "Exploiting vehicular communications for reducing co2 emissions in urban environments," in *2013 International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 2013, pp. 32–37.

[4] J. E. Naranjo, E. Talavera, J. Pérez, and C. Hidalgo, "Cooperative driving," in *Decision-Making Techniques for Autonomous Vehicles*. Elsevier, 2023, pp. 245–262.

[5] M. Amoozadeh, A. Raghuramu, C.-N. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, 2015.

[6] B. DeBruhl, S. Weerakkody, B. Sinopoli, and P. Tague, "Is your commute driving you crazy? a study of misbehavior in vehicular platoons," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2015, pp. 1–11.

[7] S. Dadras, R. M. Gerdes, and R. Sharma, "Vehicular platooning in an adversarial environment," in *Proceedings of the 10th ACM Symposium on Information, Computer and Comm. Security*, 2015, pp. 167–178.

[8] A. Virdis, G. Nardini, and G. Stea, "A framework for mec-enabled platooning," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*. IEEE, 2019, pp. 1–6.

[9] B.-J. Chang, Y.-L. Tsai, and Y.-H. Liang, "Platoon-based cooperative adaptive cruise control for achieving active safe driving through mobile vehicular cloud computing," *Wireless Personal Communications*, vol. 97, pp. 5455–5481, 2017.

[10] R. Xing, Z. Su, Q. Xu, and A. Benslimane, "Truck platooning aided secure publish/subscribe system based on smart contract in autonomous vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 782–794, 2020.

[11] A. Lekidis and F. Bouali, "C-v2x network slicing framework for 5g-enabled vehicle platooning applications," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, 2021, pp. 1–7.

[12] U. Montanaro, S. Fallah, M. Dianati, D. Oxtoby, T. Mizutani, and A. Mouzakitis, "On a fully self-organizing vehicle platooning supported by cloud computing," in *2018 Fifth International Conference on Internet of Things: Systems, Management and Security*. IEEE, 2018, pp. 295–302.

[13] A. C. Serban, E. Poll, and J. Visser, "A security analysis of the etsi its vehicular communications," in *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings 37*. Springer, 2018, pp. 365–373.

[14] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "Vanet security challenges and solutions: A survey," *Vehicular Communications*, vol. 7, pp. 7–20, 2017.

[15] R. A. Uzcátegui, A. J. De Sucre, and G. Acosta-Marum, "Wave: A tutorial," *IEEE Comm. magazine*, vol. 47, no. 5, pp. 126–133, 2009.

[16] D. Eckhoff, N. Sofra, and R. German, "A performance study of cooperative awareness in etsi its g5 and ieee wave," in *2013 10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. IEEE, 2013, pp. 196–200.

[17] J. P. Stotz, N. Bißmeyer, F. Kargl, S. Dietzel, P. Papadimitratos, and C. Schleiffer, "Preserve d1. 1 security requirements of vehicle security architecture," *PRESERVE consortium, Deliverable*, vol. 1, no. 1, pp. 1–69, 2011.

[18] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive can bus," in *2015 World Congress on Industrial Control Systems Security (WCICSS)*. IEEE, 2015, pp. 45–49.

[19] Y. Zheng, S. E. Li, J. Wang, D. Cao, and K. Li, "Stability and scalability of homogeneous vehicular platoon: Study on the influence of information flow topologies," *IEEE Transactions on intelligent transportation systems*, vol. 17, no. 1, pp. 14–26, 2015.

[20] S. E. Li, X. Qin, Y. Zheng, J. Wang, K. Li, and H. Zhang, "Distributed platoon control under topologies with complex eigenvalues: Stability analysis and controller synthesis," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 206–220, 2017.

[21] B. Liu, F. Gao, Y. He, and C. Wang, "Robust control of heterogeneous vehicular platoon with non-ideal communication," *Electronics*, vol. 8, no. 2, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/2/207

[22] K. Marton and A. Suciu, *Science and Technology*, vol. 18, no. 1, pp. 18–32, 2015.

[23] F. Mallouli, A. Hellal, N. Sharief Saeed, and F. Abdulraheem Alzahrani, "A survey on cryptography: Comparative study between rsa vs ecc algorithms, and rsa vs el-gamal algorithms," in *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 2019, pp. 173–176.

[24] H. Kurunathan, R. Severino, Ê. Filho, and E. Tovar, "Wicar-simulating towards the wireless car," in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops: DECSoS 2020, DepDevOps 2020, USDAI 2020, and WAISE 2020, Lisbon, Portugal, September 15, 2020, Proceedings 39*. Springer, 2020, pp. 136–147.

[25] X. Liu, A. Goldsmith, S. Mahal, and J. Hedrick, "Effects of communication delay on string stability in vehicle platoons," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, 2001, pp. 625–630.