Secure Group Key Dissemination Protocol in Cooperative Vehicular Platooning

Farah-Emma Braiteh[1,2,3]
[1]*AST, Renault Group*
farah-emma.braiteh@ampere.cars

Francesca Bassi [2]
[2] *IRT SystemX*
francesca.bassi@irt-systemx.fr

Rida Khatoun [3]
[3] *Télécom Paris*
rida.khatoun@telecom-paris.fr

***Abstract*—Cooperative vehicular platoons improve road safety and reduce congestion through synchronized maneuvers enabled by Vehicle-to-Vehicle (V2V) communication. Vehicles are authenticated using certificates from the Cooperative Intelligent Transport Systems (C-ITS) Public Key Infrastructure (PKI). Short-term certificates, serving as vehicle identifiers, change over time and distance, which may lead to legitimate members being misidentified as attackers in the platoon, resulting in false positives. Additionally, sensitive data, such as platoon IDs, must be protected from impersonation attacks by external vehicles. To address these challenges, we propose a secure group key-based authentication framework that uses post-quantum cryptography and Shamir's Secret Sharing for key exchange. This ensures accurate member authentication and protection of platoon data. Security analysis using the Scyther tool along with simulations using PLEXE simulator demonstrate the protocol's effectiveness in securing platoon operations against cyber threats.**

***Index Terms*—CAV Platooning, Group Key Sharing, Module-Lattice-Based Key-Encapsulation Mechanism, Shamir Secret Key Sharing, V2V.**

I. Introduction

Connected and automated vehicles (CAVs) are at the forefront of digital mobility, offering the potential to reduce road accidents, improve traffic efficiency, and lower the environmental footprint of transportation systems, as emphasized by the EU Roadmap to the Digitalisation of Transport [1]. In light of the European Commission's recent report on road safety, which revealed 20,384 road traffic fatalities in 2023 across European roads [2], the need to invest in CAV technology as a life-saving solution becomes even more pressing.

In CAV Platooning, vehicles rely on V2V data sharing to drive closely to each other and maneuver in a coordinated and synchronized manner. Platoon members multicast V2V messages containing kinematic vehicular data (position, speed, acceleration, etc.) and platoon-related information (number of members in the group, group speed, intended driving maneuvers, etc.) that allow the group to make collective decisions [3], thus enhancing safety, improving fuel efficiency, and reducing congestion.

In Europe, cooperative vehicles are authenticated using long-term certificates then granted authorization to access C-ITS services through short-term certificates issued by a trusted PKI [4]. These short-term certificates function as pseudonyms that change based on time and distance strategies [5] to enhance privacy. This dynamic pseudonym change helps mitigate tracking risks, making it difficult to link V2V messages with the identity of the originating vehicle over extended periods. Despite pseudonym changes the vehicle remains legitimate, as it retains a valid long-term certificate. Vehicles utilize the cryptographic elements of short-term certificates to sign and verify V2V messages, ensuring message integrity and user authenticity while protecting their true long-term identity.

In addition to their role in the vehicle communication, these pseudonyms play a critical role in the formation and operation of platoons. When a platoon forms, each vehicle maintains a list of member pseudonyms, updating it as new vehicles join or leave [6]. The pseudonyms in exchanged messages help verify member authenticity in C-ITS, but their dynamic nature allows a vehicle to update its "identity" through short-term certificate changes while remaining part of a platoon, potentially leading to misidentification risks. Vehicles in the platoon may thus incorrectly perceive a legitimate member having just performed a pseudonym change as an attacker transmitting sensitive platoon data. This can cause platoon disruption, such as splitting, dissolving, or operational interruptions, which may present serious safety risks.

While C-ITS authentication verifies a vehicle's legitimacy within the network, it does not ensure continuous verification of platoon membership, leaving platoons vulnerable to unauthorized access and disruptions from C-ITS certificate changes. This gap increases security risks, highlighting the need for a more robust and continuous system to authenticate and verify platoon members throughout their participation.

To address this, we propose a novel group key authentication mechanism. Given the significant limitations of existing decentralized architectures for key agreement, particularly in vehicular platoons, our protocol leverages post-quantum cryptography for secure group key establishment among vehicles and employs Shamir's Secret Sharing for key distribution and reconstruction. So, only C-ITS authenticated vehicles possessing the group key are recognized as legitimate platoon members. This approach ensures continuous authentication within the platoon, strengthening security and maintaining integrity even as vehicles change their pseudonyms.

To clarify the operational flow of this process, Fig. 1 illustrates the various phases involved in a vehicle's authentication within a platoon. The process begins with the C-ITS authentication, followed by the procedure to join another vehicle or an existing platoon. Once the new member is correctly positioned, it authenticates itself using the group

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

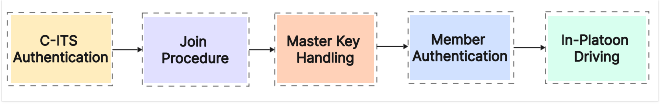

Fig. 1: *Phases of Vehicle Integration for Platoon Driving*

key. After successful authentication, the vehicle can exchange secure V2V platoon-related messages and actively participate in collective decision-making while driving within the platoon. These phases outline the structured and secure process for vehicle integration into a platoon, ensuring that only authorized vehicles are allowed to participate in platoon activities. Following the double authentication process, we use the group key to provide an additional layer of security by encrypting sensitive data in the V2V messages exchanged within the platoon. The encryption ensures that only authorized platoon members, who possess the correct group key, can decrypt and understand the transmitted information. This protects confidential data, securing the system against potential external vehicle attacks, including data interception and manipulation. Our contribution, in this paper, is the following:

- We design a protocol for group key establishment and secure distribution.
- We use the group key to authenticate members and to encrypt confidential platooning data.
- We perform a formal security analysis using Scyther tool to demonstrate the protocol's resilience against various attacks.
- We simulate and validate the protocol's effectiveness in securing platoon operations using the PLEXE simulator.

The remainder of this paper is structured as follows: Section II reviews the relevant literature pertaining to our contribution, Section III analyzes potential attacks and false positives that could impact the platoon. Section IV explains how V2V messages are signed and verified. Section V outlines the procedure for platoon creation and vehicle joining. Section VI details our proposed protocol for generating and distributing the group key. Membership authentication and data protection are discussed in Sections VII and VIII, respectively. Section IX presents and evaluates the simulation results, while Section X summarizes the findings and outlines future research directions.

II. Related Works

In this study, we aim to develop a protocol that allows vehicles to collaboratively generate a group key and securely distribute it to new members upon joining. The key will be used for continuous authentication, even as vehicle pseudonyms change during platoon movement.

Group key distribution and agreement approaches in VANETs have been widely studied, with many proposals adopting centralized architectures. For example: Lim *et al.* [7] proposed authenticating vehicles by a group signature using a private key issued by a RoadSide Unit (RSU). When a vehicle moves beyond the coverage area of its current RSU, it has to request a new key pair from the RSU corresponding to its new location. The authors in [8] investigated a group key agreement protocol in which vehicles exchange information with the RSU, which generates a signature based on this information and transmits it to the vehicle. The vehicle then utilizes the signature to derive the group key. Hao *et al.* [9] examined a group key distribution scheme in which a certificate authority initially issues private and public key pairs to RSUs. The RSUs then distribute these keys to vehicles entering their coverage area for signature. In [10], the authors proposed a scheme in which a regional manager server generates group keys and transmits them to RSUs. The RSUs then distribute these keys to nearby vehicles, enabling anonymous authentication. Zhao *et al.* [11] used a master key to encrypt messages exchanged within a platoon between the leader and the other members. This master key is distributed to vehicles by a server responsible for registering and authenticating the vehicles. In other dynamic networks as well, most protocols are based on centralized architectures, where a single entity generates a key and distributes it to other entities. The authors in [12] examined key management in dynamic wireless sensor networks and relied on cluster heads to generate a group key for encrypting secure data. To the best of our knowledge, existing decentralized architectures for key agreement, particularly in vehicular platoons, are very limited. Li *et al.* [13] introduced a secret key agreement scheme for data dissemination in platoons, which is based on a random value calculated using fading channel randomness and other public keys among members. Duan [14] presented an innovative key agreement solution for platoons, where the group key is established by multiplying public parameters shared by all members; however, the platoon has to be fully established and complete before this process can occur.

Existing authentication approaches often depend on a server or RSU to establish or distribute keys. In RSU-based schemes, the RSU acts as a fixed leader, managing key distribution within its coverage area. In contrast, in cooperative platooning, the platoon is a dynamic group with changing leadership and members, which requires a different decentralized and dynamic approach for key management as vehicles move together.

III. Cyber Attacks

In this section, we present the cyberattacks currently under study, which succeed without a group key.

Each vehicle uses a periodically changing pseudonym for temporary identification, ensuring privacy but potentially causing cyber security risks, such as member misidentification. Platoon members share a platoon ID, which ensures continuity and group association, but this makes it more susceptible to interception and potential exploitation for impersonation attacks.

A. False Positive Attack: Member Misidentification

This attack occurs when a legitimate member is mistakenly identified as an intruder. After updating its pseudonym and transmitting valid platoon information, the vehicle may be misidentified as malicious vehicle if its new pseudonym is unrecognized. As a result, the platoon may fragment or disband, reconfiguring its identification system to exclude the misidentified member.

B. Data Interception and Impersonation Attack

In this attack, an external vehicle intercepts the platoon ID and uses it as proof of membership. However, since members have to follow a join procedure and their pseudonyms are known within the platoon, the platoon ID alone is insufficient.

A successful attack requires both the platoon ID and the pseudonym of an accepted member for impersonation. The attack could be detected within the platoon, but the victim's pseudonym will be disregarded due to the security risk posed by the impersonator. As a result, the platoon may split or dissolve to exclude the compromised member.

IV. C-ITS Authentication and V2V Message Signature

Vehicles first obtain a long-term enrollment certificate then request short-term authorization certificates from trusted PKI, which issues and signs these certificates. With the short-term certificates, vehicles utilize an Elliptic Curve Digital Signature Algorithm (ECDSA) private/public key pair to sign outgoing messages and verify signatures of incoming messages. In this paper, we assume that the validity period of certificates is always maintained.

A. Signing Messages

Before transmitting a message, M, a vehicle signs it with its private key and attaches the V2V data along with the signature and its short-term certificate. The attached certificate contains essential information such as a temporary vehicle ID, public key, and other relevant details. The message has the form:

$$M = (\text{V2V Data}, \sigma, \text{Cert[PubKey]}),$$

where Cert[PubKey] is the certificate containing the public key, and σ is the message signature and is given by:

$$\sigma = \text{Sign}_{\text{privatekey}}(\text{hash}(\text{V2V Data}))$$

B. Verifying Signatures

Upon reception, other vehicles verify the message signature using the provided public key, ensuring it corresponds to the sender's private key. This process upholds the integrity of the message and authenticity of the sender.

$$\text{Verify}(M) = \begin{cases} \text{True}, & \text{if } \sigma \text{ is valid using Cert[PubKey]} \\ \text{False}, & \text{otherwise} \end{cases}$$

V. Create or Join Platoon Procedure

After the vehicle's authentication as a legitimate C-ITS station, it has to follow a procedure to join another vehicle and create a platoon or to join an existing platoon. The secure protocol for platoon creation and joining was previously detailed and evaluated in [15]. The procedure is presented in the flowchart in Fig. 2.

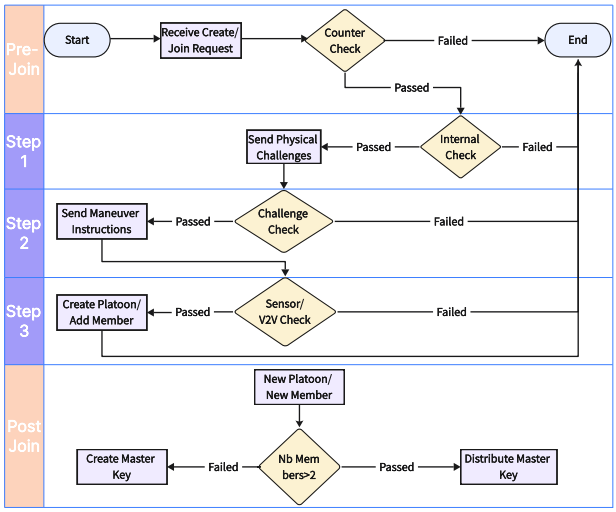

Fig. 2: *Platoon Formation: Pre-Join, Joining Process, and Master Key Handling*

The vehicle attempting to join the platoon begins by sending a join request to a vehicle in the platoon, specifically the one it will follow. This vehicle will be designated as the FrontJoin vehicle, as it will be positioned in front of the joining vehicle. The FrontJoin vehicle checks if the joiner has not exceeded an n number of join attempts. If the check passes, the join procedure begins.

Step 1: The FrontJoin vehicle first checks that the maximum number of platoon members has not been reached and that the platoon is not currently involved in other operations (such as joining, leaving, or lane-changing ...). It then prepares a series of physical challenges to send to the joining vehicle. Each challenge specifies a required distance and a time deadline. The joining vehicle has to reach the specified distance within the given time frame. The mechanism of using physical challenges was initially proposed in [16].

Step 2: The FrontJoin vehicle verifies the position and distance of the joining vehicle using broadcast vehicular data (such as position, speed, etc.) to confirm the physical checks. Once the verification is complete, the FrontJoin vehicle sends maneuver instructions to the joining vehicle. If the FrontJoin is not the last vehicle in the platoon, it will also send "opengap" instructions to its following vehicle in the platoon.

Step 3: After a period of time, the FrontJoin vehicle verifies that the joining vehicle is properly positioned behind it, using both the joiner's V2V data and its own sensor data. If this check is successful, the joining vehicle is considered part of the platoon.

In this work, we focus on the post-join phase to strengthen the security of the protocol. Specifically, we require that a joining vehicle obtains group membership through a group key, the Master Key. In the case of platoon creation, a Master Key has to be established between the two vehicles. For joining an existing platoon, the Master Key needs to be distributed as multiple shares to the joining vehicle for reconstruction.

VI. Platoon Master Key Handling

Upon successfully completing the join procedure, the new member has demonstrated accountability and genuine intent to join the platoon. However, continued membership requires ongoing validation by proving possession of the group key, referred to as the Master Key of the platoon. This section outlines the process of creating the Master Key and its distribution to new members.

A. Hybrid Key Exchange Protocol for Group Key Establishment

We aim to decentralize the creation of the Master Key by involving multiple vehicles, thereby eliminating a single point of failure. However, distributing the process of key generation among multiple entities requires secure data sharing, as the exchanged information is essential for constructing the key. Our protocol employs the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) [17], a post-quantum cryptographic algorithm standardized by the National Institute of Standards and Technology (NIST) and derived from the CRYSTALS-Kyber scheme, to securely establish a shared secret between the first two vehicles.

The key establishment process begins when a platoon is created. When a new vehicle joins another one, both have to agree on generating a Master Key for the platoon. To initiate this process, the leader of the platoon, V_0, sends a digitally signed message, as detailed in Section IV, to the joining vehicle, V_1, requesting the creation of the Master Key. To confirm its participation in the platoon, V_1 has to respond with an acknowledgment (ACK) message. Otherwise, the platoon, consisting of two vehicles, is disbanded.

After completing the agreement phase, V_1 generates an ML-KEM key pair: an encapsulation key, pk, and a decapsulation key, sk. The encapsulation key is digitally signed and sent to V_0. Upon verifying the signature, V_0 uses the received pk to perform encapsulation, which produces a cipher text, c, and a shared secret, K. The cipher text c is then signed and sent to V_1, which decapsulates it using sk to recover K. This shared secret will serve as the Master Key in our protocol.

Security Evaluation using Scyther: To validate the security of the protocol for exchanging ML-KEM parameters to establish a shared key, we used Scyther [18], a formal verification tool designed for analyzing cryptographic protocols. Scyther allows for the modeling of protocol behavior and the evaluation of its robustness against a wide range of attacks, including replay, man-in-the-middle, and other adversarial actions. Given that ML-KEM has been tested and standardized by NIST, we assume its cryptographic soundness. Consequently, our analysis focuses exclusively on verifying the integrity and authenticity of the exchanged protocol messages between V_0 and V_1. The description of Scyther's security features is provided in Table I [19].

TABLE I: *Description of Scyther Security Features*

Feature	Description
SKR (Session Key Reveal)	Ensures the confidentiality of session keys, preventing unauthorized access to cryptographic keys used in a session
Secret	Protects the confidentiality of data and messages from being disclosed to unauthorized parties
Alive	Confirms that communication with the intended target is possible, proving that the entity exists and is responsive
Niagree (Non-Injective Agreement)	Ensures that both parties recognize the communication and agree on the exchanged data
Nisynch (Non-Injective Synchronization)	Guarantees that messages are exchanged in the correct sequence, ensuring proper communication order along with agreement on data
Commit	Represents a commitment to maintaining the security and integrity of communication, ensuring the correctness of the authentication and confidentiality claims

The results from Scyther's tool for the creation of the Master Key are shown in Fig. 3. Confidential parameters are represented by the claim *'Secret'*. All exchanged messages in Scyther were signed by the sending vehicle using its private ECDSA key.

B. Master Key Distribution and Reconstruction Using Shamir Secret Sharing

At this stage, the platoon consists of two vehicles that prove their membership using the Master Key. In our approach, securely sharing the Master Key with new members is essential, without relying on a single vehicle, such as the leader, to transmit it directly. For this, we use Shamir's Secret Sharing Scheme (SSSS), a cryptographic method that allows a secret S to be divided into n shares, such that at least k shares are required to reconstruct S, relying on polynomial interpolation over a finite field [20]. In our protocol, we choose $k = 3$ and $n = N_{max}$, where N_{max} is the maximum number of members in a platoon. This method enables a new member to securely reconstruct the Master Key by communicating with any three members of the platoon who hold shares of the key.

Avoiding the direct transmission of the Master Key from the leader to a joining vehicle, in an encrypted message, offers several key advantages. First, it eliminates a single point of failure; if the leader were to be compromised, the entire security structure would be at risk. The proposed algorithm allows any two or three members to participate in retrieving the Master Key. Although the leader is incorporated in our current implementation, the system does not solely rely on it. This design also introduces improved redundancy. In traditional approaches, if the leader becomes unavailable, new members cannot obtain the Master Key. SSSS introduces redundancy, enabling multiple members to collectively provide the key. Additionally, this approach enhances scalability. The

Scyther results : verify

Claim				Status		Comments
mlkemsigned	v1	mlkemsigned,v11	Secret sk1kem	Ok	Verified	No attacks.
		mlkemsigned,v12	Secret KemDec(KemEnc(sharedsecret,pk1kem),sk1kem)	Ok		No attacks within bounds.
		mlkemsigned,v13	Alive	Ok		No attacks within bounds.
		mlkemsigned,v14	Niagree	Ok		No attacks within bounds.
		mlkemsigned,v15	Nisynch	Ok		No attacks within bounds.
	v0	mlkemsigned,v01	Secret ss0	Ok	Verified	No attacks.
		mlkemsigned,v02	Alive	Ok	Verified	No attacks.
		mlkemsigned,v03	Niagree	Ok	Verified	No attacks.
		mlkemsigned,v04	Nisynch	Ok	Verified	No attacks.

Done.

Fig. 3: *Scyther Validation- Master Key Establishment Message Exchange*

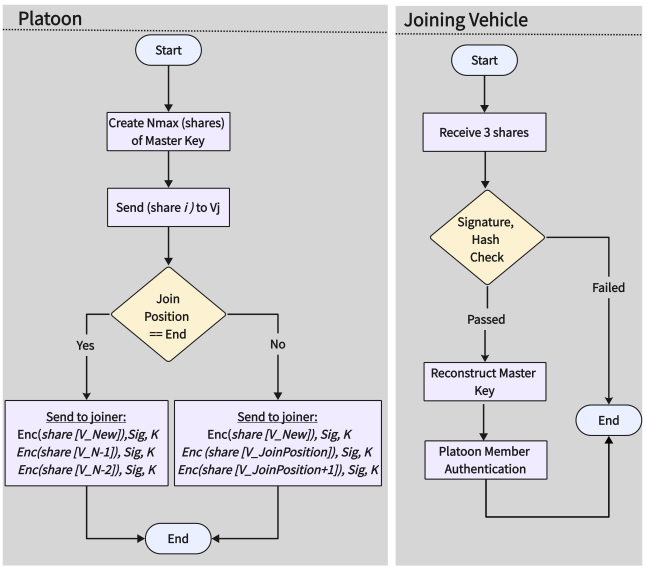

Fig. 4: *Master Key Distribution and Reconstruction*

distributed nature of SSSS ensures that the system remains scalable and robust, even as the number of members increases. The platoon's initiation of the key distribution process and the new member's reconstruction of the Master Key are illustrated in Fig. 4.

Key Distribution: the lead vehicle, V_0, defined as the first vehicle in the platoon, generates up to $N_{\max}$ shares, corresponding to the maximum number of platoon members. We consider $N_{max} = 5$ shares, labeled share$_0$ to share$_4$. V_0 assigns a share$_i$ to the existing member, where $i < N_{max}$. If V_0 leaves the platoon, V_1 is designated as the lead vehicle.

Upon joining, the new vehicle receives three shares. The first share is randomly chosen from the shares created by V_0 that have not yet been assigned to existing members and is sent by V_0. The second share is sent by the vehicle preceding the joining point, $V_{\text{JoinPosition}}$. For example, if a vehicle joins at the end of the platoon, the second share is transmitted by the current last vehicle, V_{N-1}, where N is the actual number of platoon members. The third share depends on the joining position:

- If the vehicle joins at the end of the platoon (JoinPosition == End in Fig. 4), it receives the share of the vehicle just before the last one, V_{N-2}. In this paper, all vehicles join from the end.
- If the vehicle joins in the middle of the platoon, it receives the share of the vehicle immediately after the joining point, $V_{\text{JoinPosition}+1}$.

Each share is encrypted and sent separately to the joiner in a signed message.

Secret Reconstruction according to Shamir: Given at least k shares, the secret can be reconstructed using Lagrange interpolation [21]. The secret corresponds to the value of the polynomial at $x = 0$. Shamir's Secret Sharing ensures Security and Threshold Property:

- Threshold Property: Any k shares can reconstruct S.
- Perfect Secrecy: Fewer than k shares reveal no information about S, since the remaining degrees of the polynomial introduce uncertainty.

The joining vehicle verifies the C-ITS signature of the received messages, decrypts the received shares using its ECIES private key, and reconstructs the Master Key of the platoon.

Security Evaluation using Scyther: The message exchange between the three vehicles: V_0, V_1, and V_2, where V_2 is the joiner, was verified using Scyther. The description of the security features used is the same as the ones provided in section VI-A in Table I. Scyther tool's results are shown in Fig. 5.

C. Master Key Revocation

When a vehicle leaves the platoon or the Master Key is compromised, this key is revoked. In this case, the leader V_0 establishes a new Master Key with the succeeding vehicle V_1. It then generates shares of the new key and distributes these shares to the remaining members. Each vehicle receives one share from the leader, one from its preceding vehicle, and one

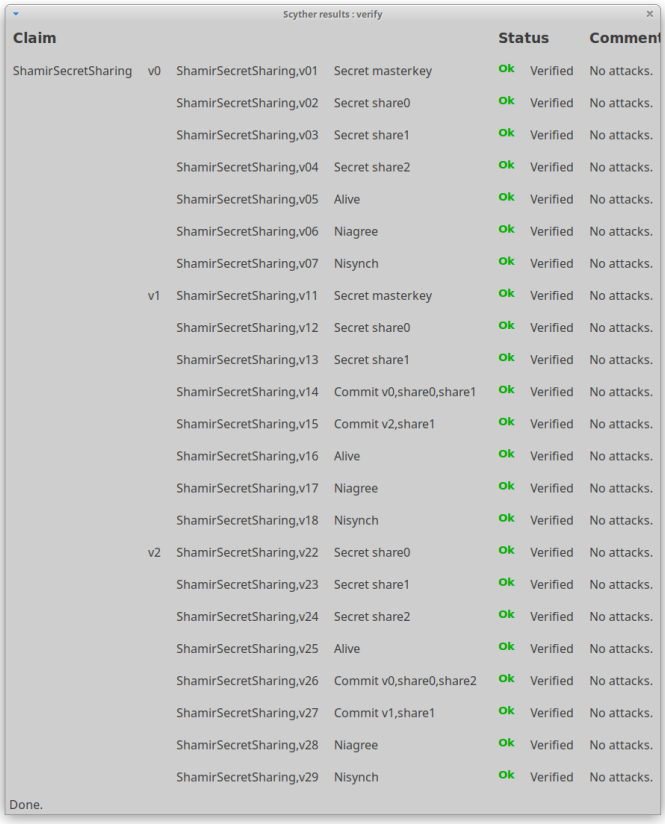

Fig. 5: *Scyther Validation - Master Key Distribution and Reconstruction Message Exchange*

from its following neighbor; in the case of the last vehicle, this share is provided by the member preceding its front vehicle.

VII. Group Key-Based Authentication for Platoon Members

After constructing the Master Key, platoon members shall use it to compute message hashes for V2V communication within the platoon, which are then signed and used to verify the membership of the sending vehicles.

A. Message Transmission

Once the new member computes the Master Key, it has to use it to generate the HMAC of the V2V platooning data, which will then be signed using the ECDSA private key, as explained in section IV. We modify the traditional message signature, σ, and introduce a new variant, σ', defined as follows:

$$\sigma' = \text{Sign}_{\text{privatekey}}(\text{HMAC}(\text{V2V Platooning Data}, \text{MasterKey}))$$
$$HMAC_{value1} = \text{HMAC}(\text{V2V Platooning Data}, \text{MasterKey})$$

where the MasterKey is that of the sender.

So, the message, M, retains its original structure while incorporating the modified signature generated using the Master Key:

$$M = (\text{V2V Platooning Data}, \sigma', \text{Cert}[\text{PubKey}]).$$

B. Message Reception

When a platoon member receives M, it first verifies the signature σ' using the sender's ECDSA public key, as detailed in Section IV. If the verification succeeds, the receiver checks whether the sender possesses the Master Key by computing an HMAC over the extracted V2V platooning data from M using its own Master Key, resulting in $HMAC_{value2}$. It then compares $HMAC_{value2}$ with $HMAC_{value1}$. Matching values confirm that the sender used the same Master Key to generate the message digest, thereby validating group membership and ensuring message integrity.

$$HMAC_{value2} = \text{HMAC}(\text{V2V Platooning Data}, \text{MasterKey}),$$

where MasterKey is that of the receiver.

$$\text{Verify}(M) = \begin{cases} \text{True}, & \text{if } HMAC_{value2} \text{ equals } HMAC_{value1} \\ \text{False}, & \text{otherwise} \end{cases}$$

VIII. Platooning Confidential Data Protection

To mitigate the risk of interception and impersonation attacks, described in Section III-B, which could lead to the dissolution of the platoon, we encrypt the platoon ID using the Master Key and transmit it within a signed message. So, the V2V message structure remains consistent with that discussed in Section VII:

$$M = (\text{V2V Platooning Data}, \sigma', \text{Cert}[\text{PubKey}])$$

However, V2V platooning data now consists of the concatenation of the encrypted Platoon ID and other V2V dynamic data: $V2VPlatooningData = (Enc_{MasterKey}(PlatoonID)||dynamicdata)$

The Master Key can be used to encrypt the entire V2V Platooning Data, rather than just a portion of it. However, in the context of car passenger platoons, this level of encryption is unnecessary, as the other information does not present a security risk. This approach, however, may be more relevant in military platoons to secure data exchanged between vehicles.

IX. Simulation Results

To validate the Master Key establishment and distribution protocol described in Sections VI-A and VI-B, we used the PLEXE simulator coupled with SUMO. PLEXE is an extension of Veins designed specifically for simulating cooperative driving and platooning scenarios, providing support for platooning control algorithms and inter-vehicle communication. SUMO allows modeling vehicle dynamics and mobility patterns. The simulation parameters are summarized in Table II. Common scenario: Vehicles V_0 and V_1, which were traveling on $Lane1$, formed the platoon. Vehicles V_2 and V_3, initially positioned on $Lane2$, and V_4, already on $Lane1$ joined the platoon sequentially, following the procedure outlined in Section V, until the platoon reached the maximum size, $N_{max} = 5$ members. The validation of the join procedure is explored in a separate work and is beyond the scope of the present study.

Once the platoon was complete, V_0 remained the lead vehicle and V_4 became the last one. Each vehicle had been

TABLE II: *Simulation Parameters*

Platoon Params	**Value**	**V2V-Specific Params**	**Value**
N max	5	TX frequency	10 Hz
N members	5	V2V data size	150 Bytes
Speed	110 km/h	Signature size	64 Bytes
Controller	CACC	PubKey size	64 Bytes

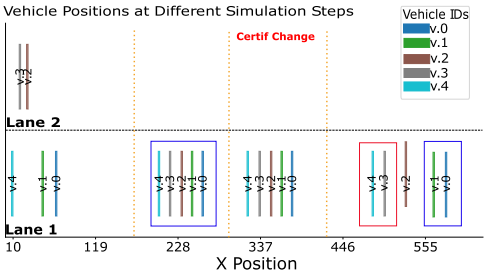

Fig. 6: *Platoon Formation Update Without Master Key*

TABLE III: *Platooning Message Statistics for Vehicles- Without a Master Key*

Vehicle	Sent Msgs	Received Msgs	Accepted	Rejected
V.0	104	335	325	10
V.1	104	334	324	10
V.2	88	316	316	0
V.3	73	286	276	10
V.4	70	286	276	10

operating for some time on the road with its respective short-term certificate. As new members joined, each member updated its platoon members list. Notably, V_2 changed its short-term certificate, thus its pseudonym, after driving for a while in the platoon. In the following subsections, we compare the impact of the pseudonym change of V_2 on the platoon, first without the Master Key of the platoon and then after its implementation.

A. Results Without the Master Key Implementation

The platoon formation is illustrated in Fig. 6. Initially, vehicles V_0 to V_4 traveled in close proximity across different lanes. As additional vehicles joined V_0, a five-member platoon was formed. Upon acceptance, each new member began multicasting platooning V2V messages, which were verified as outlined in Section IV. At a later stage, V_2 changed its pseudonym but continued transmitting V2V messages. However, since its identity was no longer recognized, signature verification failed, and its messages were rejected. After 1 second (10 rejected messages), the platoon classified V_2 as an attacker and ceased communication with it. As a result, V_2 was removed from the member list, causing the platoon to split into two groups: V_0 and V_1 in one, and V_3 and V_4 in another. Table III presents the message exchange statistics over a period of 10 seconds, from platoon creation until its split. Eventually, V_2 resumed driving independently and changed lanes, as shown in Fig. 6. After the split, each new platoon assigned itself a new ID, and its members updated their IDs and member lists accordingly.

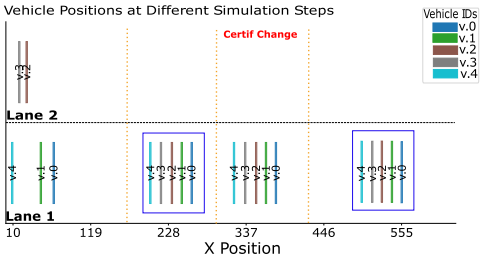

Fig. 7: *Platoon Formation Update With Master Key*

TABLE IV: *Platooning Message Statistics for Vehicles- With a Master Key*

Vehicle	Sent Msgs	Received Msgs	Accepted	Rejected
V.0	195	702	702	0
V.1	195	701	701	0
V.2	179	684	684	0
V.3	164	653	653	0
V.4	164	652	652	0

TABLE V: *Cryptographic Evaluation Parameters*

Key Size	**Bytes**	**Avg Time**	**msec**
Encaps Key, pk	800	Master Key establishment	200
Decaps key, sk	1632	Message signature	0.57
Master Key	32	Master Key distribution and reconstruction	0.17
ECDSA Key	32	Message verification	1.76

B. Results After the Master Key Implementation

The same scenario was simulated again, this time with the implementation of the Master Key. Signature verification was performed as described in Section VII.

1) Operational Results: Fig. 7 illustrates the platoon formation at the same simulation steps shown in Fig. III.

After V_2 changed its pseudonym, the platoon remained intact and fully operational. This was further confirmed by the steady increase in message exchanges and the absence of rejected messages, indicating uninterrupted communication among platoon members, as shown in Table IV. We stopped the simulation after 19.5 seconds of the platoon's creation.

2) Performance Evaluation and Discussion: All experiments were conducted on a machine running Ubuntu 20.04.6 LTS with an Intel Core i7-6820HQ CPU, without the use of specialized hardware acceleration. The evaluation of the cryptographic parameters is shown in Table V. The challenge of using post-quantum cryptography in C-ITS lies in the resulting message sizes, which can exceed the maximum allowed size of approximately 1400 bytes for wireless short-range communication, including the payload, signature, and certificate. This is due to the large post-quantum key sizes, often several kilobytes, appended to encrypted messages when post-quantum encryption algorithms are used [22], or the substantial signatures generated by post-quantum signature algorithms [23], which are much larger than those used in classical cryptography. In our protocol, we use post-quantum cryptography to generate the Master Key, which is then used

to compute the HMAC of the message. The resulting digest is subsequently signed using a standard ECDSA key. This approach ensures that the message signature size matches the typical ECDSA signature size of 64 bytes, leading to the expected results for signature generation and verification.

The latency for non-safety-critical V2V applications should not exceed 300 milliseconds. So, the Master Key generation time is considered acceptable for V2V communication, as it occurs during the platoon creation phase, before the platoon becomes fully functional and vehicles begin exchanging real-time, safety-critical information. Its distribution and reconstruction takes less than 0.2 milliseconds, which is well within acceptable limits and does not introduce any significant latency impact.

The number of signature verifications per second is approximately 560, which is considered acceptable but close to the lower limit. This is attributed to the use of a low-end machine running the verification algorithms and the decision to verify the signatures of all received V2V messages, across various message types. To improve efficiency, a strategic approach should be adopted by selectively verifying messages rather than verifying every incoming message.

X. Conclusion and Future Works

In this research, we focus on ensuring secure and seamless communication among connected and cooperative vehicles within a platoon, particularly during pseudonym changes. We propose the creation of a group key between the first two vehicles using post-quantum cryptography. As new vehicles join, shares of the key are distributed to facilitate its reconstruction by the joiner, utilizing the Shamir Secret Sharing Scheme. Through simulations, we demonstrated that the protocol successfully identifies vehicles as members, even as their pseudonyms change while driving in the platoon. Additionally, we used the group key to encrypt sensitive data, thereby preventing interception and manipulation attacks. Achieving a balance between identity protection, security, and operational efficiency is crucial for maintaining the integrity of vehicle platoons. In our simulations, the Master Key is not updated. However, this key should be updated regularly. Future work will focus on dynamic updates of the Master Key.

Acknowledgment

This work has been supported by the French government under the "France 2030" program, as part of the Cybersecurity for Trusted Mobility (CTM) project at the SystemX Technological Research Institute.

References

[1] European Commission, "2030 Digital Compass: the European Way for the Digital Decade," 2021. [Online]. Available: https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX%3A52021DC0118

[2] ——, "European Road Safety Observatory," 2025. [Online]. Available: https://road-safety.transport.ec.europa.eu/european-road-safety-observatory_en

[3] F.-E. Braiteh, F. Bassi, and R. Khatoun, "Platooning in Connected Vehicles: A Review of Current Solutions, Standardization Activities, Cybersecurity, and Research Opportunities," *IEEE Transactions on Intelligent Vehicles*, 2024.

[4] ETSI, "Intelligent Transport Systems (ITS); Security; ITS Communications Security Architecture and Security Management; Release 2," Technical Specification TS 102 940 V2.1.1, 2021, accessed: 10 March 2025.

[5] ——, "Intelligent Transport Systems (ITS); Security; Pre-standardization Study on Pseudonym Change Management; Release 2," Technical Report TR 103 415 V2.1.1, 2025, accessed: 10 March 2025.

[6] F.-E. Braiteh, D. Tse, O. Yhia, F. Bassi, and R. Khatoun, "A Secure and Cooperative Departure Protocol for Connected Automated Platoons," in *12th IEEE/IFIP International Conference on New Technologies, Mobility and Security*, 2025.

[7] K. Lim, K. M. Tuladhar, X. Wang, and W. Liu, "A Scalable and Secure Key Distribution Scheme for Group Signature-Based Authentication in VANET," in *Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. IEEE, 2017, pp. 478–483.

[8] L. Liu, Y. Wang, J. Zhang, and Q. Yang, "A Secure and Efficient Group Key Agreement Scheme for VANET," *Sensors*, vol. 19, no. 3, p. 482, 2019.

[9] Y. Hao, Y. Cheng, and K. Ren, "Distributed Key Management with Protection Against RSU Compromise in Group Signature-Based VANETs," in *Proceedings of IEEE GLOBECOM 2008 - IEEE Global Telecommunications Conference*. IEEE, 2008, pp. 1–5.

[10] Y. Sun, Z. Feng, Q. Hu, and J. Su, "An Efficient Distributed Key Management Scheme for Group-Signature Based Anonymous Authentication in VANET," *Security and Communication Networks*, vol. 5, no. 1, pp. 79–86, 2012.

[11] Y. Zhao, Y. Wang, Y. Liang, H. Yu, and Y. Ren, "Identity-Based Broadcast Signcryption Scheme for Vehicular Platoon Communication," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 6, pp. 7814–7824, 2022.

[12] U. Vasala and D. G. Sakthidharan, "Effective Key Management in Dynamic Wireless Sensor Networks," *International Journal of Computer Engineering in Research Trends*, vol. 4, no. 7, pp. 308–312, 2017.

[13] K. Li, L. Lu, W. Ni, E. Tovar, and M. Guizani, "Secret Key Agreement for Data Dissemination in Vehicular Platoons," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 9060–9073, 2019.

[14] C. Duan, "Group Key Agreement Schemes for Platooning with a Dynamic Lead," Master's thesis, The Ohio State University, 2021.

[15] F.-E. Braiteh, F. Bassi, and R. Khatoun, "Securing Cooperative Vehicular Platooning with a Set of Reinforced Checks," in *IEEE International Wireless Communications and Mobile Computing (IWCMC)*, 2025, pp. 1027–1033.

[16] M. Arslan, M. F. Majeed, R. Abu Bakar, J. Khan, S. Hussain, Y. Lee, and F. Khan, "CAVVPM: Challenge-Based Authentication and Verification of Vehicle Platooning at Motorway," *Sensors*, vol. 22, no. 20, p. 7946, 2022.

[17] NIST, "Module-lattice-based key-encapsulation mechanism standard," U.S. Department of Commerce, Tech. Rep. FIPS 203, Aug. 2024. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.203

[18] C. Cremers, *Scyther User Manual*, 2023, available at: https://github.com/cascremers/scyther/blob/master/gui/scyther-manual.pdf.

[19] A. Nemalikanti and M. Saifulla, "EN-LAKP: Lightweight Authentication and Key Agreement Protocol for Emerging Networks," *IEEE Access*, vol. 11, pp. 28 645–28 657, 2023.

[20] D. Bogdanov, "Foundations and Properties of Shamir's Secret Sharing Scheme Research Seminar in Cryptography," *University of Tartu, Institute of Computer Science*, vol. 1, 2007.

[21] R. Chen, "A Review of Research on Secret Sharing," *Applied and Computational Engineering*, vol. 88, pp. 202–207, 2024.

[22] B. Lonc, F.-E. Braiteh, and F. Bassi, "Hybrid Key Exchange and Signature Design for Quantum-safe C-ITS," in *IEEE/IFIP Workshop on Secure and Resilient Communication for Drones and Autonomous Vehicle Networks (SRC-DAV)*, 2025.

[23] B. Lonc, A. Aubry, H. Bakhti, M. Christofi, and H. A. Mehrez, "Feasibility and Benchmarking of Post-quantum Cryptography in the Cooperative ITS Ecosystem," in *IEEE Vehicular Networking Conference (VNC)*, 2023, pp. 215–222.