EEGAP: ECC-based Efficient Group Authentication Protocol for dynamic vehicular platoon

Article

Accepted Version

Cheng, H., Wang, Z., Song, J., Zhong, Q., Pooranian, Z. and Martinelli, F. (2025) EEGAP: ECC-based Efficient Group Authentication Protocol for dynamic vehicular platoon. IEEE Transactions on Intelligent Transportation Systems, 26 (11). 20392 -20406. ISSN 1524-9050 doi: 10.1109/TITS.2025.3589792 Available at https://centaur.reading.ac.uk/124260/

To link to this article DOI: http://dx.doi.org/10.1109/TITS.2025.3589792

Publisher: IEEE

EEGAP: ECC-Based Efficient Group Authentication Protocol for Dynamic Vehicular Platoon

Hongyuan Cheng, Zihan Wang, Jingcheng Song, *Member, IEEE*, Qi Zhong, Zahra Pooranian, *Senior Member, IEEE*, Fabio Martinelli, *Senior Member, IEEE*, and Mohammad Shojafar, *Senior Member, IEEE*

***Abstract*— Vehicular platooning has emerged as a promising paradigm in intelligent transportation, offering significant benefits such as reduced energy consumption, improved road throughput and mitigated traffic congestion. However, the open nature of vehicular communication channels exposes platoons to a wide range of security and privacy threats. Although existing group key-based protocols provide foundational security services, they often incur substantial computation overhead and insufficiently address vehicle privacy, making them unsuitable for dynamic vehicular platoon. Therefore, this paper introduces an efficient group authentication protocol (EEGAP) for dynamic vehicular platoons, which ensures privacy-preserving and secure communication during platoon restructuring operations, such as merging and splitting, by integrating anonymous authentication, fog computing, and group key agreement mechanisms. Leveraging Elliptic Curve Cryptography (ECC) and secret sharing mechanisms, EEGAP enables lightweight yet robust group key negotiation, reducing computation overhead by 7.08% and communication overhead by 6.85% compared to existing schemes. Both formal security proofs and informal analysis confirm that EEGAP satisfies the stringent security requirements of vehicular platoon communication systems.**

***Index Terms*— Anonymous authentication, secure communication, group key agreement, fog computing, vehicular platoon.**

I. Introduction

VEHICULAR platoon [1] is considered a significant research hotspot in vehicular ad hoc networks (VANETs) and plays a crucial role in advancing intelligent transportation systems (ITS) [2]. A vehicular platoon consists of a platoon head vehicle leading several member vehicles at a fixed inter-vehicle distance. In particular, platoon control is mainly concerned with maintaining the required fixed distance between vehicles, which is divided into two main control strategies: the leader-predecessor and bidirectional-leader strategies [3]. The bidirectional leader strategy is widely used in the platoon system to maintain a fixed distance between vehicles, due to its feasibility and lower cost compared than the leader-predecessor strategy. Based on the bidirectional-leader topology, the platoon head vehicle collects crucial data (road conditions, speed, platoon length, etc.) from member vehicles and infrastructure for platoon topology adjustments. Meanwhile, member vehicles exchange information with their neighbors and rely on the platoon head vehicle for out-of-platoon communication [4]. Ultimately, a central server integrates data from multiple platoons for global scheduling and management. To reduce reliance on central servers, fog computing [5] has been adopted to offload data processing to fog nodes (FNs), lowering bandwidth usage, computational cost, and latency [6].

Furthermore, there are two primary communication modes in fog-assisted vehicular platoon systems: in-platoon communication and out-of-platoon communication. The platoon head vehicle plays a pivotal role in platoon, as depicted in Fig. 1, which is responsible for platoon decisions, including speed control and topology adjustments (platoon splitting and merging) and interactions between the vehicular platoon and out-of-platoon entities via vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) [7], both of which occur over public channels. Here, V2V enables communication within the platoon, while V2I connects the platoon head vehicle with infrastructures (roadside units (RSUs) or FNs).

However, open wireless communication exposes vehicular platoon systems to significant privacy and security vulnerabilities [8]. Privacy risks arise as vehicle messages often contain sensitive information, such as vehicle identity, location, and route history [9], [10]. Malicious attackers can extract this sensitive information from vehicle messages transmitted over public channels using data mining techniques, thereby posing significant risks to the safety and privacy of users [11]. Security threats pose a major challenge to vehicular platoon communication. Attackers may launch disruptive attacks, including spoofing, replay, and tampering, to manipulate platoon coordination or gain unauthorized access. Such attacks can compromise the integrity, availability, and confidentiality of the platoon communication systems, ultimately affecting platoon safety. Therefore, effectively addressing the aforementioned privacy and security threats faced by vehicular platoon communication systems is a critical requirement for

Received 2 August 2024; revised 14 December 2024 and 19 April 2025; accepted 4 July 2025. This work was supported in part by RDS—PTR25-27 P2.1 Cybersecurity, in part by the Key Science and Technology Research and Development Plan of Ningbo in 2023, and in part by the Third Batch of Approval Projects under Grant 2023Z228. The Associate Editor for this article was K. C. Leung. *(Corresponding author: Zahra Pooranian.)*

Hongyuan Cheng, Zihan Wang, and Jingcheng Song are with the School of Computer Science and Engineering, Linyi University, Linyi 276000, China.

Qi Zhong is with the Faculty of Data Science, City University of Macau, Macau 999078, China.

Zahra Pooranian is with the Department of Computer Science, University of Reading, RG6 6UR Reading, U.K. (e-mail: zahra.pooranian@gmail.com; z.pooranian@reading.ac.uk).

Fabio Martinelli was with the Consiglio Nazionale delle Ricerche (CNR), 56124 Pisa, Italy. He is now with the Institute for High Performance Computing and Networking, National Research Council of Italy, 87036 Rende, Italy.

Mohammad Shojafar is with 5G and 6G Innovation Centre (5G/6GIC), University of Surrey, GU2 7XH Guildford, U.K.

Digital Object Identifier 10.1109/TITS.2025.3589792

their practical deployment. It is crucial to ensure the privacy, confidentiality, and reliability of messages transmitted over wireless channels.

A. Motivations

In fog-assisted vehicular platoon communication systems, secure authentication and group key management technologies are essential for ensuring the integrity and confidentiality of messages within the platoon [12]. However, most existing research on group authentication primarily targets static platoon topologies and ignores the challenges of dynamic topology adjustments, such as platoon merging and splitting, for platoon secure communication. Dynamic topology adjustments are fundamental for the practical deployment of platoon systems, as they enable flexibility adapt to changing traffic conditions, vehicle join/leave events.

Platoon merging and splitting introduce specific challenges in maintaining platoon secure communication. During platoon merging, it is crucial to enable leaders of merging platoons to rapidly and securely negotiate a new group key, ensuring a unified, secure communication environment for all platoon members. In contrast, during platoon splitting, each newly formed platoon must independently generate a new group key to maintain communication confidentiality.

These dynamic topology changes directly impact group key management and member authentication, both critical for platoon communication security. Existing schemes face key limitations in dynamic environments: Firstly, most existing studies only focus on static or fixed platoon topologies, making them inefficient in securely updating group keys during platoon merging and splitting. Secondly, secure platoon communication relies on group key encryption, but improper key management during topology changes can lead to unauthorized access or key leakage. Additionally, most related work does not meet all the features presented in Table III, and thus cannot provide stronger security and privacy protection for platoon communication systems. Finally, existing group key management schemes typically rely on a central server for key agreement and are built on computationally expensive cryptographic primitives (e.g., bilinear pairing), which make them unsuitable for resource-constrained vehicular environments [13], [14]. Thus, developing communication protocols that efficiently and securely support dynamic topology adjustments is essential for ensuring seamless platoon coordination, maintaining vehicular platoon system integrity, and optimizing vehicular platoon capacity.

B. Contributions

This paper proposes an Efficient and Secure Authentication and Group Key Negotiation Protocol (EEGAP) to address the critical challenges of privacy, confidentiality, and communication efficiency in dynamic vehicular platoons with group key assistance. The proposed protocol is designed to enable seamless service provisioning, enhance vehicular privacy, and establish a secure, robust and trustworthy communication environment. The main contributions of EEGAP are as follows:

- **Efficient authentication mechanism for dynamic platoons.** We introduce an efficient authentication mechanism for dynamic vehicular platoons, enabling the platoon head vehicle to securely authenticate with forwarding nodes (FNs) or member vehicles while preserving identity privacy. Each vehicle autonomously generates a pseudonym, leveraging a public-private key pair initially provisioned by a Trusted Authority (TA), with the final public-private key independently derived by the vehicle itself. This approach eliminates the need for key escrow, thereby enhancing security and trust. Furthermore, by leveraging the capabilities of FNs, the proposed protocol significantly reduces computational overhead, ensuring lightweight and efficient identity verification in platoon communication.
- **Optimized Group Session Key Negotiation.** During the vehicular platoon splitting, Shamir's secret sharing is utilized to broadcast the group session key across the new platoon efficiently. This method significantly reduces communication overhead while maintaining strong security guarantees. The proposed approach ensures that newly formed platoons can rapidly establish secure communication channels without exposing key material to unauthorized entities.
- **Formal Security Model and Performance Evaluation.** We conduct a rigorous formal security analysis, demonstrating that EEGAP satisfies the stringent security and privacy requirements of vehicular platoon communication, effectively mitigates various potential attacks. Additionally, performance evaluations confirm that EEGAP outperforms existing schemes in terms of computation efficiency and communication overhead, making it a practical for real-world deployment in dynamic vehicular platoon systems.

The rest of this paper is organized as follows. Section II introduces the related work of this paper. Section III presents the preliminaries. Section IV describes the system model, threat model, and security objectives. Section V proposes the EEGAP scheme, while Section VI provides security proofs and analysis. Section VII discusses the computational and communication costs of the proposed scheme compared to existing alternatives. Finally, Section VIII concludes the paper.

II. Related Work

Authentication and key negotiation are critical for preventing unauthorized access and ensuring secure communication within vehicular platoon systems. The security of dynamic platoon adjustments relies heavily on robust authentication and group key negotiation protocols. Table I provides a comparative summary of existing certifiable group key negotiation schemes for vehicular networks.

Harishma et al. [15] proposed a mutual authentication and key exchange protocol for secure communication,, while Mansour et al. [16] introduced a centralized group key management protocol. However, both schemes [15], [16] impose a significant computational burden on the central server, as it must handle multiple cryptographic operations for node management. Zhang et al. [17] proposed a broadcast authentication scheme aimed at improving authentication efficiency between vehicles and fog nodes in a privacy-preserving manner. Nevertheless, the above schemes suffer from the potential problem

TABLE I
COMPARISON OF DIFFERENT SCHEMES WITH THE EEGAP IN TERMS OF ADVANTAGES AND LIMITATIONS

Scheme	Main Technology	Advantages	Limitations
[15]	Physically unclonable functions, Smart meter	Efficient authentication with lightweight cryptography, Defense against communication and physical attacks	Increase the server's burden
[17]	Fog computing, Elliptic Curve Cryptography	Efficient and privacy-preserving vehicle-fog authentication	High dependency on central authority, causing computational and communication burden
[18]	Elliptic Curve Cryptography, Fuzzy Logic Control System	Privacy-preserving 5G vehicle authentication, Efficient edge computing with low overhead	Risk privacy breaches
[19]	Hash-chain, Elliptic Curve Cryptography	Efficient authentication, simplified CSP selection	Vulnerable to short-lived secret leakage, susceptible to impersonation attacks
[20]	Bilinear maps, Threshold cryptography	Delegate authentication capability to edge nodes, Support fast handover authentication	Not consider impersonation attacks, Not provide the privacy protection, High computation and communication costs
[21]	Bilinear pairing	Enhanced trust through, improved performance in delay, delivery	High certificate management costs, insecure key custody during registration
[22]	Bilinear pairing, Chinese Remainder Theorem	Fine-grained permission distribution, Flexibility and high security	Excessive computation and communication
[24]	Bilinear maps	Efficient group key agreement, attribute-based information sharing	Pairing operations bring heavy computation burden
[25]	Elliptic Curve Cryptography	Secure and efficient communication, tree-based key agreement	Message reliability is not considered
EEGAP	Elliptic Curve Cryptography, Threshold cryptography	Support V2F anonymous authentication, achieve message reliability and confidentiality, and reduce vehicle memory burden	-

that it requires real-time participation from a central authority in all authentication and session key negotiations, leading to excessive computational and communication overhead.

To mitigate reliance on a central authority for authentication and session key negotiation, Zhang et al. [18] proposed a mutual authentication scheme between vehicles and edge computing devices. This scheme integrates pseudonymous with Elliptic Curve Cryptography (ECC) to enable secure authentication without requiring assistance from a trusted edge computing vehicle. However, it remains vulnerable to single-point failures, which can compromise privacy and system availability.

To address some of these limitations, Cui et al. [19] introduced an extensible conditional privacy-preserving authentication scheme that employs hash functions to encrypt vehicle anonymity, the true identity of Cloud Service Providers (CSPs), and temporary information for generating session keys. Despite its advantages, the scheme exposes the CSP's true identity to certified vehicles, introducing a potential privacy risk. Yang et al. [20] proposed an edge-assisted decentralized authentication protocol. This protocol enables fast handover authentication while mitigating the risk of a single point of failure by delegating authentication capabilities to distributed edge nodes. However, both schemes proposed by Cui et al. [19] and Yang et al. [20] fail to provide adequate identity privacy protection for vehicles.

Balaji et al. [21] further proposed an authentication and key agreement protocol for Vehicular Ad Hoc Networks (VANETs), integrating Elliptic Curve Cryptography (ECC), Diffie-Hellman key exchange protocols and bilinear mapping mechanisms to enhance communication security. While this scheme strengthens identity privacy protection, it fails to fully address the challenges associated with high certificate management costs and insecure key storage during vehicle registration. Although the scheme improves authentication and key negotiation efficiency by removing reliance on a central authority, it overlooks critical security and privacy concerns for resource-constrained vehicular environments.

Applying the aforementioned authentication and key agreement techniques to vehicular platoon communication systems poses significant challenges, as these methods primarily focus on securing single connected vehicles rather than dynamic platoons. To address the security and privacy threats in platoon communication, Liu et al. [22] proposed a scheme that supports dynamic vehicle adjustments, enabling seamless vehicle join and leave operations. However, this approach does not address the critical issue of group key negotiation within platoons. Several works have sought to address group key agreement in vehicular networks. Zhang et al. [23] proposed a hierarchical dynamic group key agreement protocol that incorporates an attribute revocation chain based on blockchain technology, enabling the revocation of ciphertext policy attributes. Similarly, Zhang et al. [24] introduced an asymmetric group key agreement protocol based on attribute authentication, which preserves the benefits of traditional identity-based key agreement protocols while enhancing user privacy protection and improving key management flexibility. Wei et al. [25] devised a tree-based key agreement algorithm to handle two key scenarios: the joining of authenticated vehicles and the departure of vehicles from the platoon. Additionally, Zhao et al. [26] proposed an identity-based encryption scheme for broadcast signatures in vehicular platoon communication, allowing the platoon head to securely negotiate a key with member vehicles to ensure data confidentiality, integrity, and authenticity. Although the aforementioned schemes enhance the security of platoon communications, these schemes rely on computationally intensive cryptographic operations, making them unsuitable for resource-constrained vehicular platoon systems.

To overcome these limitations, our proposed scheme effectively enhances security, privacy, and efficiency, as outlined in Table I. By establishing authenticated group keys, it ensures secure and efficient communication within dynamic vehicular platoons while minimizing computation overhead.

III. Preliminaries

This section presents the essential preliminaries, including Elliptic Curve Cryptography (ECC) and Shamir's secret sharing, which serve as foundational components for EEGAP.

A. Elliptical Curve Cryptosystem (ECC)

Miller and Koblitz proposed ECC which is widely utilized due to its ability to achieve strong security with shorter key lengths. F_q denotes a finite field, and the elliptic curve E on the finite field F_q is defined as $y^2 = x^3 + ax + b \bmod q$, where $a, b \in F_q$, $4a^3 + 27b^2 \neq 0$. The group G, composed of points on the elliptic curve along with the point at infinity O, constitutes an additive group G of order q [27], [28]. There are two binary operations and the computational difficulty of specific mathematical problems, as outlined below.

- **Point addition:** Let $P, Q \in G$ are two points on E, when $P + Q = R$ and $P \neq Q$, R is said to be the intersection point of the straight line. Otherwise, if $P + Q = R$ and $P = Q$, then $R = 2P$.
- **Scalar point multiplication:** Scalar multiplication involves computing mP as the repeated addition of P, expressed as $mP = P + P + \cdots + P(m$ times), where $m \in Z_q^*$.
- **Elliptic Curve Discrete Logarithm Problem (ECDLP):** Given two random points $P, R \in G$, it is computationally infeasible to output the random value x satisfying $R = xP$ when x is in an unknown state, where $x \in Z_q^*$ [29].
- **Elliptic Curve Computational Diffie-Hellman Problem (ECCDHP):** Given three random points $P, R, V \in G$, it is challenging to compute xyP when x, y are in an unknown state, where $R = xP$, $V = yP$ and $x, y \in Z_q^*$ [30].

B. Shamir's Secret Sharing

The Shamir Secret Sharing scheme [31] s a cryptographic protocol designed to distribute a secret among multiple participants in such a way that it can only be reconstructed when a predefined number of shares are combined.

Assuming there are n users $\{V_1, V_2, V_3, \cdots, V_n\}$ and a trusted dealer D. The D chooses a polynomial of order $t-1$ over the finite domain F_q: $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$, where $a_0, a_1, a_2, \cdots, a_{t-1} \in F_q$ and p is a large prime number. The secret value s is set as $s = a_0$ and t represents the threshold number of shares required to reconstruct the secret. The dealer generates n unique shares in the form of coordinate pairs (x_i, y_i) and distributes them to participants. To reconstruct the secret, at least t shares must be collected. Using these shares, the polynomial $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$ can be reconstructed, enabling the recovery of the secret value s.

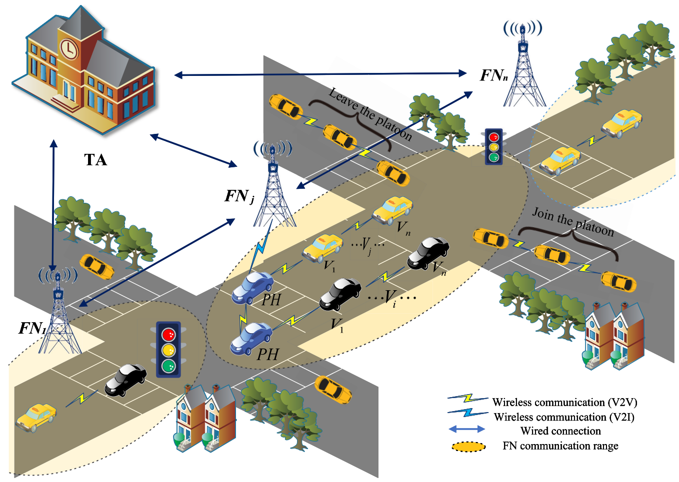


Fig. 1. The proposed network model. TA: Trusted Authority; FN: Fog Node; PH: platoon head vehicle; V_n: member vehicle.

IV. Models of EEGAP

This section presents the network model and threat model underlying the proposed EEGAP protocol.

A. Network Model

As shown in Fig.1, the designed network model comprises three participants: Trusted Authority (TA), Fog Nodes (FNs), and Vehicles. Their respective roles are described as follows:

TA is a fully trusted and uncompromised entity equipped with substantial computational, storage, and communication capabilities. Its core responsibilities include registering vehicles and FNs, issuing digital certificates, and generating global system parameters. Furthermore, as the sole entity capable of tracing the real identity of contested vehicles, the TA maintains a registry of all malicious vehicles' real identities.

FNs as a network edge node, comprising a local data storage server and wireless communication infrastructure, allowing wide-area coverage at the network edge. FNs facilitate communication with vehicles using predefined protocols and are responsible for broadcasting and forwarding information within vehicular platoon systems. With robust storage and computing capabilities, the FN acts as a gateway between the vehicular platoon and the TA, facilitating access authentication for vehicular platoons within its communication range. FNs monitor real-time traffic conditions and may issue platoon topology adjustment commands (e.g., merging or splitting). As semi-trusted entities, FNs are assumed to execute protocols honestly but may be curious about vehicular platoon privacy (vehicle identity or platoon session keys).

Vehicles. Each platoon includes a platoon head vehicle and multiple member vehicles, which maintain a defined inter-vehicular distance. The platoon head vehicle is responsible for coordinating platoon movement, maintaining communication between member vehicles within the platoon by V2V, and between the platoon and external entities (FNs or TA) by V2I. Member vehicles follow the control commands issued by the platoon head. All vehicles act as a semi-trusted entity, honestly executing the protocol but curious about other platoons interaction messages and vehicle identities.

Fig. 1 depicts several vehicular platoons traveling within the communication range of a fog node FN_j. Each platoon is managed by a head vehicle and maintains coordination through V2V and V2I communications. Upon receiving topology adjustment commands (e.g., platoon fusion or split) from FN_j, the platoon updates its topology. Subsequently, a secure group session key negotiation and authentication process is initiated to ensure reliable and confidential intra-platoon communication. This work focuses on securing platoon communication after topology adjustments, rather than the formation of platoon topologies.

B. Threat Model

All protocol phases except the registration phase are executed over public channels (insecure channels) in the vehicular platoon system. TA is assumed to be fully trustworthy and immune to malicious compromise. However, vehicles and FNs, as semi-trusted entities, are considered potential insider adversaries capable of launching attacks to disrupt normal platoon communication operations. Accordingly, the security properties of this work are evaluated by using the Dolev–Yao (DY) threat model and the Canetti and Krawczyk (CK)-adversary model [32]. The DY threat model not only helps an attacker to eavesdrop on transmitted messages, but also interrupts them by modifying, replaying, or injecting false messages into the communication channel. Consequently, the EEGAP is vulnerable to both external attacker A_1 and internal attacker A_2, as described below:

- **External Attackers:** External attacker A_1 represents an external entity that attempts to masquerade as a legitimate vehicle. A_1 aims to compromise the confidentiality and integrity of the vehicular platoon system by launching various attacks, including replay, man-in-the-middle and impersonation. Furthermore, it seeks to infer sensitive information such as vehicle identities and negotiated group session keys.
- **Internal Attackers:** Internal attacker A_2 is a legitimate vehicle that performs malicious behavior, which can obtain secret parameters in the vehicular platoon system but cannot replace the public key of the legitimate vehicle. It maximizes own benefits primarily by sending false information to legitimate vehicles, impersonating their identities, and colluding with other malicious attackers.

Furthermore, external attacker A_1 in the EEGAP can also exploit the currently widely recognized *de facto* CK-adversary model, which offers a stronger adversarial capability than the DY model. Due to the openness of the platoon communication environment, A_1 can fetch all the secret credentials stored in the memory of vehicle terminals through the power analysis attack. Thus, the proposed protocol is exposed to a potential active attack called Ephemeral Secret Leakage (ESL) attack, i.e. A_1, which attempts to compute the session key established by two communicating entities during the access control process while obtaining a short-term secret.

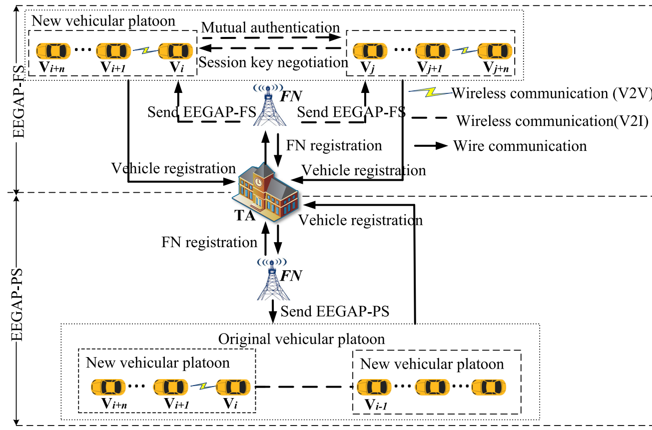


Fig. 2. The overall flow of the EEGAP.

TABLE II
NOTATIONS AND DESCRIPTIONS IN THIS PAPER

Notation	Description
TA	A trusted authority
V_i,V_j	i-th vehicle, j-th vehicle
G	A cyclic additive group
q,P	The order and generator of G
ID_{fj},RID_i	The real identity of FN_j,V_i
PID_i	A pseudonym of V_i
P_{pub},s	System public and private key pair
$h()$	The hash functions
(u_i, d_i)	The full private key pair of V_i
(U_i, Z_i)	The full public key pair of V_i
T, t_i, t_j	The timestamp
σ_i,σ_j	Signatures from V_i and V_j
SK,SK_{ij},SK_{ji}	Session key between V_i and V_j
GK	Group key
u_i, z_i, w_j, x_i, x_j	Random numbers
$\oplus$	Bitwise XOR operation

V. PROPOSED PROTOCOL

This section provides extensive details on the phases and workflow of EEGAP. The parameter symbols and definitions designed in this paper are summarized in Table II. EEGAP comprises four main phases: System Initialization, Registration, Vehicular Platoon Fusion (EEGAP-PF), and Vehicular Platoon Split (EEGAP-PS). The overall procedural flow of EEGAP is shown in Fig. 2, and each phase is discussed in detail as follows.

A. System Initialization

During this phase, TA generates essential parameters of the entire vehicular platoon communication system. The steps are as follows:

Step 1: TA selects a cyclic additive group G of order prime q, with P as the generator of G.

Step 2: TA picks the elliptic curve $E : y^2 = x^3+ax+b \bmod q$ defined over the finite field F_q, where $x, y \in [0, q-1]$, $a, b \in F_q$, and the E satisfied $4a^3 + 27b^2 \neq 0 \bmod q$.

Step 3: TA randomly selects a primary private key $s \in Z_q^*$ and computes the corresponding system public key $P_{pub} = sP$.

Step 4: TA picks four hash functions $h_0 : \{0,1\}^* \to Z_q^*$, $h_1 : \{0,1\}^* \to Z_q^*$, $h_2 : \{0,1\}^* \to Z_q^*$, $h_3 : \{0,1\}^* \to Z_q^*$.

Step 5: TA securely stores the primary key s and maintains a registration status list $\{ID_i, iv_i\}$, where $iv_i = 1$ indicates registered and $iv_i = 0$ indicates deregistration. The public system parameters $params = \{G, p, P, P_{pub}, h_0, h_1, h_2, h_3\}$ are broadcast to all registered entities.

B. Registration

All vehicles (platoon head and member vehicles) as well as FNs must register with TA before participating in the platoon communication system. This registration process, conducted over a secure channel, allows vehicles and FNs to obtain their respective partial and long-term key pairs respectively.

- **Vehicle Registration**

Step 1: A vehicle V_i selects a random value $u_i \in Z_q^*$ as the partial private key and calculates the corresponding partial public key $U_i = u_i P$.

Step 2: The vehicle V_i sends its real identities RID_i and U_i to TA to initiate registration and obtain the full key pair.

Step 3: When receiving the registration request from V_i, TA selects a random number $z_i \in Z_q^*$ and calculates part of the public key $Z_i = z_i P$. Together with the system private key and the real identity of the vehicle, a parameter containing identity information and a partial private key $D_i = Z_i \oplus h_0(RID_i)\, U_i$, $d_i = z_i + s h_1(U_i, Z_i) \bmod q$ are generated for the vehicle by TA. Finally, TA returns (d_i, D_i) to the vehicle and sets the vehicle registration status $iv_i = 1$.

Step 4: V_i constructs its full private key (u_i, d_i) and full public key (U_i, Z_i), which are stored securely within its tamper-proof device.

- **FN Registration**

Similarly, FN will send its own identity ID_{fj} to TA for registration and obtain a complete key pair.

Step 1: TA selects a random number $w_j \in Z_q^*$ as FN_j's private key and calculates the corresponding public key $W_j = w_j P$, and $sk_j = w_j + s h_2(ID_{fj}, W_i) \bmod q$.

Step 2: TA sends (W_j, sk_j) to FN_j. Eventually, FN_j secretly stores (W_j, sk_j).

C. Scenario 1: Vehicular Platoons Fusion (EEGAP-PF)

This scenario occurs when two platoons, due to traffic conditions or strategic maneuvering, must merge into a single platoon. The FN initiates the fusion by sending a proposal to the respective platoon head vehicles V_i and V_j. These two vehicles must authenticate each other and establish a secure group session key without assistance from the TA. The prompt and secure negotiation of a new session key for the restructured platoon becomes crucial for maintaining both the efficiency and security of in-platoon communication. The fusion process employs the ECC and anonymous authentication mechanism, detailed in Algorithms 1 and 2. The step-by-step procedure is as follows.

Step 1: Platoon head vehicle V_i picks a random number $x_i \in Z_q^*$ and computes $X_i = x_i P$. For subsequent anonymous communication, V_i combines the timestamp T_i, the real identity RID_i and partial public key Z_i to generates a temporary pseudonymous $PID_i = RID_i \oplus h_3(x_i P_{pub}, Z_i, T_i)$, which can be performed offline.

Step 2: After that, V_i uses the random number x_i and its full private key (u_i, d_i) to generate a signature $\sigma_i = d_i + u_i + x_i \alpha_i \bmod q$, where $\alpha_i = h_2(PID_i, Z_i, X_i, T_i)$. Then, it sends the authentication message $M_1 = \{PID_i, X_i, U_i, Z_i, \sigma_i, T_i\}$ to the platoon head vehicle V_j of the platoon to be merged.

Step 3: Upon receiving the message M_1, the platoon head vehicle V_j checks the freshness via $T_i^* - T_i < \Delta T$ and validates the signature by verifying: $\sigma_i P = Z_i + P_{pub} h_1(U_i, Z_i) + U_i + X_i \alpha_i$, where $\alpha_i = h_2(PID_i, Z_i, X_i, T_i)$.

Step 4: If the verification fails, the message M_1 is discarded, otherwise V_j selects a random number $x_j \in Z_q^*$, calculates $X_j = x_j P$, and generates the temporary pseudonym $PID_j = RID_j \oplus h_3(x_j P_{pub}, Z_j, T_j)$ along with timestamp T_j and its partial public key Z_j.

Step 5: V_j generates the session key $SK_{ji} = h_3(x_j X_i, PID_j, PID_i)$ and signature $\sigma_j = d_j + u_j + x_j \alpha_j \bmod q$, where $\alpha_j = h_2(PID_j, Z_j, X_j, T_j)$. It responds with message $M_2 = \{PID_j, X_j, U_j, Z_j, \sigma_j, T_j\}$ to V_i.

Step 6: Upon receiving M_2 from V_j, V_i checks if $\sigma_j P = Z_j + P_{pub} h_1(U_j, Z_j) + U_j + X_j \alpha_j$ holds. If the check fails, V_i rejects M_2. Otherwise, it calculates the session key

Algorithm 1 Vehicle Signature Generation

Input: random number x_i, z_i; private key=(u_i, d_i); the real identity RID_i; timestamp T_i; the generator of elliptic curves P;
Output: message M_1;
: $X_i \leftarrow x_i P$, $Z_i \leftarrow z_i P$;
: $\alpha_i \leftarrow h_2(PID_i, Z_i, X_i, T_i)$;
: $\sigma_i \leftarrow d_i + u_i + x_i \alpha_i \bmod q$;
: $PID_i \leftarrow RID_i \oplus h_3(x_i P_{pub}, Z_i, T_i)$;
: **Return** $M_1 \leftarrow \{PID_i, X_i, U_i, Z_i, \sigma_i, T_i\}$;

Algorithm 2 Anonymous Authentication and Session Key Agreement in Vehicular Platoons Fusion

Input: random number x_j, z_i; private key=(u_i, d_i); the real identity RID_i RID_j; timestamp T_j; the generator of elliptic curves P;
Output: negotiate session keys $SK = SK_{ji} = SK_{ij}$;
: $X_i \leftarrow x_i P$; $Z_i \leftarrow z_i P$;
: $\alpha_i \leftarrow h_2(PID_i, Z_i, X_i, T_i)$;
: if $\sigma_i P == Z_i + P_{pub} h_1(U_i, Z_i) + U_i + X_i \alpha_i$;
: $X_j \leftarrow x_j P$;
: $PID_j \leftarrow RID_j \oplus h_3(x_j P_{pub}, Z_j, T_j)$;
: $\alpha_j \leftarrow h_2(PID_j, Z_j, X_j, T_j)$;
: $\sigma_j \leftarrow d_j + u_j + x_j \alpha_j \bmod q$;
: $SK_{ji} \leftarrow h_3(x_j X_i, PID_j, PID_i)$;
: Else reject;
: End if;
: $MessageM_2 \leftarrow \{PID_j, X_j, U_j, Z_j, \sigma_j, T_j\}$;
: $X_j \leftarrow x_j P$;
: $SK_{ij} \leftarrow h_3(x_i X_j, PID_i, PID_j)$;
: $\alpha_j \leftarrow h_2(PID_j, Z_j, X_j, T_j)$;
:   **If** $\sigma_j P == Z_j + P_{pub} h_1(U_j, Z_j) + U_j + X_j \alpha_j$;
: $SK \leftarrow SK_{ji} == SK_{ij}$;
:   **Else** reject;
:   **End if**;
:   **Return** $SK = SK_{ji} = SK_{ij}$;

$SK_{ij} = h_3\left(x_i X_j, PID_i, PID_j\right)$. Once the session key SK_{ij} is obtained, V_i can use it to conduct subsequent secure sessions with V_i, where $SK_{ij} = SK_{ji} = SK$.

Step 7: Referring to our previous work [33], it was possible to evaluate the reputation of the platoon head vehicles V_i and V_j of the two platoons. Then, the vehicle with a high reputation score is recommended as the new platoon's head vehicle. Eventually, the original platoon head vehicles of both vehicular platoons will broadcast the new vehicular platoon head vehicle and session key SK to respective platoons.

D. Scenario 2: Vehicular Platoons Split (EEGAP-PS)

Vehicular platoon splitting is another crucial operation in platoon topology adjustment. When a single platoon is divided into two or more sub-platoons, the newly formed platoons must establish trust and secure communication channels to maintain the integrity and confidentiality of their operations. This scenario typically occurs in response to dynamic traffic conditions or vehicular behavior, as assessed by the Fog Node (FN). Upon evaluating real-time traffic data and the operational status of the platoon, the FN issues a platoon split suggestion. The platoon head vehicle of the original platoon continues to serve as the platoon head vehicle of the new platoon after the split, while the FN designates new platoon head vehicles for the remaining sub-platoons. Simultaneously, to incentivize safe and lawful driving, the reputation scores of all candidate platoon head vehicles are assessed in accordance with the evaluation mechanism proposed in [33]. After splitting, the newly formed platoons negotiate their respective platoon's group keys with the help of group key agreement technology. Taking the platoon led by vehicle V_i as an example, the key negotiation procedure for establishing the group key GK_i is outlined below. Algorithm 3 shows the pseudocode of the vehicular platoon split.

Step 1: Assume that the platoon head vehicle V_i leads a group of K member vehicles. Then V_i uses its own private key u_i and the public key U_f of K member vehicles V_f to directly calculate a secret information $sk_{if} = u_i U_f = u_i u_f P$ for secure communication.

Step 2: The platoon head vehicle V_i selects a group session key GK_i, and uses public system parameters along with the public keys of all K member vehicles to construct a set of tuples $\left\{PID_f, \left(sk_{if}, T_f\right)\right\}$, where T_f represents the current timestamp.

Step 3: V_i then uses the K point and a point $(0, GK_i)$ to constructs a polynomial function $f(x) = GK_i + a_1 x + a_2 x^2 + \cdots + a_k x^k$, where GK_i represents the group key.

Step 4: Using the constructed polynomial $f(x)$, V_i computes additional K points $(\lambda_k, y_k) = (\lambda_k, f(\lambda_k))\ (k = 1, \cdots, K)$. Then, the K points are securely broadcast within the communication range of the platoon.

Step 5: Each member vehicle in the platoon that receives K points reconstructs the polynomial $f(x)$ with the received K points along with the self-generated points $\left\{PID_f, \left(sk_{fi}, T_f\right)\right\}$, and calculates $GK_i = f(0)$, where $sk_{fi} = u_f U_i$ and $sk_{fi} = sk_{if}$. Notably, due to the reliance on mutual key derivation and the necessity of possessing valid partial keys, an adversary or an unauthorized vehicle cannot derive the group key GK_i of the platoon led by V_i. This method effectively minimizes communication overhead while establishing a secure and robust communication environment for the newly formed platoons.

Algorithm 3 Group Session Key Generation in Vehicular Platoons Split

Input: private key u_i; public key U_f; group key GK; timestamp T_f;

Output: Group session key $GK_i = f(0)$;

1 : $sk_{if} \leftarrow u_i U_f = u_i u_f P$;

2 : $K\ \ pcs\left\{\left(PID_f, sk_f \parallel T_f\right)\right\} \leftarrow \left(params, U_f, u_i\right)$;

3 : $f(x) = GK_i + a_1 x + a_2 x^2 + \cdots + a_k x^k \leftarrow (0, GK_i) + \left\{\left(PID_f, sk_f \parallel T_f\right)\right\}$;

4 : $(\lambda_k, y_k) = (\lambda_k, f(\lambda_k))\ (k = 1, \cdots, K) \leftarrow f(x) = GK_i + a_1 x + a_2 x^2 + \cdots + a_k x^k$;

5 : $GK_i = f(0) \leftarrow (\lambda_k, y_k) + \left\{PID_f, \left(sk_{fi}, T_f\right)\right\}$;

6 : **Return** $GK_i = f(0)$;

VI. Security Proof and Analysis

This section provides a security analysis of EEGAP, formally proving its semantic security and informally verifying its security goals.

A. Security Model

EEGAP operates within a certificateless cryptographic framework in which the Key Generation Center (KGC) is responsible for generating only partial private keys, thereby mitigating the necessity of placing full trust in the KGC. However, the system remains susceptible to two primary classes of adversaries: malicious users A_1 and a potentially malicious KGC A_2. The KGC operates under the management of the TA.

- Malicious user A_1 can query and replace the public key of a legitimate user but does not possess knowledge of the system's primary key. A_1 is restricted from querying the private key or any part of the private key of the challenged identity. However, A_1 can generate keys and certificates for any identity except the challenged one and cannot replace a public key that has been previously revealed.
- A malicious KGC A_2 holds the primary secret key but cannot query or replace the public key of a legitimate user or regenerate existing public parameters. Specifically, as A_2 already holds the system's primary key, there is no necessity to request certificate generation.

Assuming that A and B are two participants in the authentication protocol, $\Pi^k_{A,B}$ denotes the $k-th$ session instance of this protocol between them. The following games define the session key security of the authentication protocol.

Game 1: The challenger C generates the public parameters $params$ and P, then provides $params$ to A_1 while keeping them secret. It can query A_1 as follows.

Secret value-query. A_1 asks for the secret value of identity ID, and the challenger C executes the secret value generation algorithm to derive X_{ID} and returns it to A_1.

Public key-query. A_1 initiates a query to generate a public key for the user's ID. It sends the ID to C, where the challenger executes the secret value generation algorithm and the public key generation algorithm to return the corresponding public key P_{pub} to A_1.

Part of the private key-query. A_1 requests the generation of a portion of the private key for the user's ID, sends the ID and P_{pub} to C, and challenger C runs the partial private key generation algorithm to generate the corresponding portion of the private key Y_{ID} and sends it to A_1.

Private key-query. A_1 performs a private key generation query for identity ID sends ID to C. C runs the secret value generation algorithm and part of the private key generation algorithm and returns the secret value X_{ID} and part of the private key Y_{ID} as the private key to A_1.

Public key substitution-query. A_1 replaces the public key P_{pub} of identity ID with a maliciously chosen P^*_{pub}, which C records.

Send-query. The query simulates an active attack by A_1, if A_1 with message m sends this query, processes message m according to the protocol specification, and returns the result to adversary A_1. A_1 initiates the identity legitimacy authentication protocol by sending this query.

Reveal-query. The query simulates a key association attack. If session $\Pi^k_{A,B}$ has completed and derived a session key, that key is returned to A_1.

$Test\left(\Pi^k_{A,B}\right)$*-query.* The query simulates the attacker's capability to acquire a session key. The adversary selects a session $\Pi^k_{A,B}$ to $Test$ the query, and upon receiving the query, C randomly selects $b \in \{0, 1\}$. If $b = 1$, the correct session key is provided. Otherwise, C furnishes A_1 with a random integer of the same length as the session key. After the $Test$ is asked, a guess of the random number b is output b'. The game is considered successful if the following conditions are satisfied:

- A_1 does not perform partial private key generation or private key generation inquiries for A and B, nor does it execute a *Reveal query* for session $\Pi^k_{A,B}$.
- Participants A and B negotiate the same session key SK, which remains unknown to any party other than A and B.

The advantage of adversary A_1 in winning this game is defined as: $Adv_{A_1}(\kappa) = \left|Pr[b' = b] - \frac{1}{2}\right|$, where κ is the security parameter [34].

Game 2: The challenger C generates the public parameters $params$ and the primary key P_{pub}, and sends them together to adversary A_2. The types of queries available to A_2 namely *Secret value-query, Public key-query, Send-query, Reveal-query* and $Test\left(\Pi^k_{A,B}\right)$*-query* follow the same structure and semantics as those in **Game 1** and are therefore omitted here for brevity.

B. Security Proof

This section presents formal security proofs concerning the non-enforceability of authentication and the negotiation of secure session keys under adversarial conditions. It is divided into two cases according to the adversary's capabilities.

Theorem 1: Assuming that adversary A_1 impersonates the platoon head vehicle and successfully forges a valid authentication request with a non-negligible probability. A probabilistic polynomial time (PPT) adversary A_1 can solve the ECDLP with an evident advantage $\varepsilon'_1 \geq \left(1 - \frac{1}{e}\right) \frac{\varepsilon_1}{q_h e(q_1+q_2+q_3+1)}$, where q_1, q_2 and q_3 denote the frequencies of various types of queries (part of the private key, private key, and oracle queries), and e represents the natural logarithmic base.

Proof: Adversary C obtains the public parameters (q, P, G) and the challenge tuple (P, sP), where $P_{pub} = sP$. Then, C sends the public parameters $params = \langle q, P, G, P_{pub}, H_0, H_1, H_2, H_3, H_4\rangle$ to A_1, where H_0, H_1, H_2, and H_3 are one-way hash functions, and H_4 is a random oracle machine. C maintains four initially empty lists: L_{H_4}, L_{pub}, L_{sk} and L_d, which store the queries to the H_4 oracle machine, public keys, private keys, and partial private values. Additionally, C uses the list L_s to record the identity authentication requests submitted by A_1. C randomly selects RID^*_i as the challenge identity.

Part of the private key-query. For the query (RID_i, U_i), if $RID_i = RID^*_i$, C terminates. Otherwise, C checks if $(RID_i, Z_i, d_i) \in L_d$. If it exists, C returns (Z_i, d_i) to A_1. If not, C randomly selects $d_i, H_i \in Z^*_q$ such that $\left(*, *, *, H_{u_i}\right) \notin L_{H_4}$, calculates $Z_i = d_i P - H_i P_{pub}$, and returns (Z_i, d_i) to A_1. C then adds tuples (RID_i, Z_i, d_i) and (RID_i, U_i, Z_i, H_i) to the list L_d and L_{H_4}, respectively.

H_4*-query*. The query input is (RID_i, U_i, Z_i). If there exists $(RID_i, U_i, Z_i, H_i) \in L_{H_4}$, then C returns the corresponding H_i to A_1. Otherwise, C randomly selects $u_i \in Z^*_q$, computes $U_i = u_i P$, and performs part of the private key generation query on RID_i by submitting tuples (RID_i, U_i) to C. Upon obtaining the relevant response, C records (RID_i, U_i, Z_i, H_i) in L_{H_4} and returns H_i to A_1.

Private key-query. For a private key query on identity RID_i, if $RID_i = RID^*_i$, C terminates the game. Otherwise, if $(RID_i, u_i, d_i) \in L_{sk}$, the private key is returned directly to A_1. If not, C randomly selects $u_i \in Z^*_q$, calculates $U_i = u_i P$, and generates the partial private key query for (RID_i, U_i). After receiving the corresponding response (Z_i, d_i), C returns (u_i, d_i) to A_1, and records the tuples (RID_i, u_i, d_i) and (RID_i, U_i, Z_i) in L_{sk} and L_{pub}, respectively.

Public key-query. For a public key query with input RID_i, if $(RID_i, U_i, Z_i) \in L_{pub}$, C returns the corresponding public key to A_1. If $RID_i = RID^*_i$, then C randomly selects $u^*_i, z^*_i \in Z^*_q$, calculates $U^*_i = u^*_i P$ and $Z^*_i = z^*_i P$, returns $\left(U^*_i, Z^*_i\right)$, and records $\left(RID^*_i, U^*_i, Z^*_i\right)$ in L_{pub}.

Public key substitution-query. In this query, A_1 is allowed to substitute the original public key pk_i of a known RID_i. C updates the corresponding entry in L_{pub} accordingly.

Send-query: The following forms of Send queries simulate interactions between platoon head vehicles:

Send $\left(\Pi^k_{i,j}, start\right)$. This query simulates the situation where the platoon head vehicle V_i sends an authentication request to the corresponding platoon head vehicle V_j, and A_1 can access the message m_{v_1} sent from V_i. When C

receives the query, it selects the random numbers $x_i \in Z_q^*$, calculates $PID_i = RID_i \oplus h_3\left(x_i P_{pub}, Z_i, T_i\right)$, $\sigma_i = d_i + u_i + x_i\alpha_i \bmod q$, where $\alpha_i = h_2\left(PID_i, Z_i, X_i, T_i\right)$. C then sends the message $m_{v_1} = \{PID_i, X_i, U_i, Z_i, \sigma_i, T_i\}$ to A_1.

Send $\left(\Pi_{i,j}^k, m_{v_1}\right)$. The query simulates the platoon head vehicle V_j responding to an authentication request from V_i. A_1 can access the message m_{v_1} sent by V_j. Upon receiving the query, C first verifies the correctness of σ_i. If it is correct, C calculates $X_j = x_j P$, $SK_{ji} = h_3\left(x_j X_i, PID_j, PID_i\right)$ and $\alpha_j = h_2\left(PID_j, Z_j, X_j, T_j\right)$. Otherwise, C stops the query and sends $\left\langle PID_j, X_j, U_j, Z_j, \sigma_j, T_j\right\rangle$ to A_1.

Suppose an adversary forges a legitimate authentication request message $\{PID_i, X_i, U_i, Z_i, \sigma_i, T_i\}$. The following verification equation must hold: $\sigma_i P = Z_i + P_{pub} h_1\left(U_i, Z_i\right) + U_i + X_i h_1\left(PID_i, Z_i, X_i, T_i\right)$. Upon obtaining a forged but valid signature σ_i, C faces two unknown parameters: (i) the value s, which represents the solution to the ECDLP, and (ii) the random scalar x_i chosen by the adversary A_1 when generating the signature. Therefore, it is not possible to determine their specific values, and C cannot successfully forge an authentication request by solving the ECDLP.

To analyze the adversary's advantage in successfully completing the above security game, let us define the following events: E_1 indicates that the game was not terminated during the simulation, E_2 indicates that an identity request message about the challenge identity was forged, and E_3 indicates that a valid authentication request message $\{PID_i, X_i, U_i, Z_i, \sigma_i, T_i\}$ has been generated. Note that during the *Part of the private, Private key* and *Send-query* phases, the simulation terminates if $RID_i = RID_i^*$. Therefore, there is $Pr\left(E_1\right) = \left(1 - \frac{1}{q_1+q_2+q_3+1}\right)^{q_1+q_2+q_3}$, $Pr\left(E_2\right) = \frac{1}{q_1+q_2+q_3+1}$, $Pr\left(E_3\right) \geq \varepsilon_2$, then there is $\varepsilon_2' = Pr\left(E_1 \wedge E_2 \wedge E_3\right) \geq \frac{\varepsilon_2}{e(q_1+q_2+q_3+1)}$.

In conclusion, if an adversary A_1 exists that can successfully forge a valid authentication message with non-negligible probability, then this adversary can solve the ECDLP with an advantage: $\varepsilon_1' \geq \left(1 - \frac{1}{e}\right) \frac{\varepsilon_1}{q_h e(q_1+q_2+q_3+1)}$.

Theorem 2: Assume that adversary A_1 successfully impersonates the platoon head vehicle and forges a legitimate response message with a non-negligible probability ε_2. Then, adversary A_1 can solve the ECCDHP with a significant advantage $\varepsilon_2' \geq \left(1 - \frac{1}{e}\right) \frac{\varepsilon_2}{e(q_1+q_2+q_3+1)}$, where q_1, q_2, and q_3 denote the number of *Part of the private, Private key*, *Send* and *Oracle* queries, respectively.

Proof: The adversary obtains the corresponding public parameters (q, P, G) and tuple $(P, \alpha P, \beta P)$ from the challenger C of the ECCDHP. After running the initialization algorithm, it sends the public parameters $params = \left\langle q, P, G, P_{pub}, H_0, H_1, H_2, H_3, H_4\right\rangle$ to A_1, while secretly storing the primary private key. Here, H_0, H_1, H_2 and H_3 are one-way hash functions, and H_4 is a random oracle machine. C maintains three initially empty lists L_{pub}, L_{sk}, and L_d, which store the public keys, the private keys and the partial private keys. Additionally, C uses list L_s to record the identity authentication requests submitted by A_1. C randomly selects RID_i^* as the challenge identity.

Part of the private key-query, Private key-query, and *Public key substitution* are similar to those described in **Theorem 1**.

Public key-query. The input to this query is RID_j. If $\left(RID_j, U_j, Z_j\right) \in L_{pub} C$ returns $\left(U_j, Z_j\right)$ to A_1. Otherwise, C performs the following operations. If $RID_j = RID_j^*$, then $U_j^* = sP$, C randomly selects $d_j^*, H_j^* \in Z_q^*$ satisfying $\left(*, *, *, h_j^*\right) \notin L_{H_4}$, and calculates $Z_j = z_j P$. It adds tuples $\left(RID_j^*, \perp, y_j^*\right)$, $\left(RID_j^*, U_j^*, Z_j^*\right)$, $\left(RID_j^*, U_j^*, Z_j^*, H_j^*\right)$ to the list L_{sk}, L_{pub} and L_{H_4}, respectively, and returns $\left(U_j^*, Z_j^*\right)$ to A_1. Otherwise, C randomly selects $u_j \in Z_q^*$, calculates $U_j = u_j P$ and queries the partial private key generation for $\left(RID_j, U_j\right)$ to obtain $\left(Z_j, d_j\right)$. It returns $\left(U_j, Z_j\right)$ to A_1 and adds $\left(RID_j, u_j, d_j\right)$ and $\left(RID_j, U_j, Z_j\right)$ to L_{sk} and L_{pub}, respectively.

Send-query. In this game, A_1 can send three types of *Send* queries.

Send $\left(\Pi_{i,j}^k, start\right)$. If $RID_j = RID_j^*$ and $RID_i \in L_{sk}$, C calculates $X_j = x_j P$. Then, $M_2 = \left\{PID_j, X_j, U_j, Z_j, \sigma_j, T_j\right\}$ is calculated according to the protocol and returned to A_1. Otherwise, C retrieves the private key $\left(u_j, d_j\right)$ for RID_i from the private key generation query, generates an authentication request message, and updates L_s.

Send $\left(\Pi_{j,i}^k, (\sigma, T)\right)$. If $RID_j = RID_j^*$, the game is terminated. Otherwise, C generates the private key for RID_j by querying for it and then verifies the legitimacy of σ_j. If verification passes, C computes the response according to the protocol and returns it to A_1. Otherwise, the query is rejected.

Send $\left(\Pi_{i,j}^k, \left(U_j, X_j, Z_j, T\right)\right)$. If the information of RID_i can be retrieved in L_s, C retrieves the corresponding RID_j and verifies whether the equation $\sigma_j P = Z_j + P_{pub} h_1\left(U_j, Z_j\right) + U_j + X_j h_1\left(PID_j, Z_j, X_j, T_j\right)$ holds. If the above equation holds, A_1 successfully impersonates and forges a legitimate identity authentication response message. Otherwise, the game is terminated.

If Adversary A_1 can forge a valid authentication reply message, it implies that C successfully provides the correct parameter x_j to adversary A_1. Where $X_j = x_j P$, C then outputs $\sigma_j = d_j + u_j + x_j\alpha_j \bmod q$ as the solution to the ECCDHP.

To evaluate the advantage of adversary C in the aforementioned game, event E_1 is defined to signify that the game is not terminated during the simulation. E_2 indicates that the adversary forged an identity authentication response to the challenge, and E_3 indicates a valid authentication reply message. Therefore, there are $Pr\left(E_1\right) = \left(1 - \frac{1}{q_1+q_2+q_3+1}\right)^{q_1+q_2+q_3}$, $Pr\left(E_2\right) = \frac{1}{q_1+q_2+q_3+1}$, $Pr\left(E_3\right) \geq \varepsilon_2$, hence $\varepsilon_2' = Pr\left(E_1 \wedge E_2 \wedge E_3\right) \geq \frac{\varepsilon_2}{e(q_1+q_2+q_3+1)}$.

In conclusion, if there exists an adversary A_1 capable of generating a legitimate identity authentication response with a non-negligible probability ε_2' then the challenger C can solve the ECCDHP with a significant advantage $\varepsilon_2' \geq \frac{\varepsilon_2}{e(q_1+q_2+q_3+1)}$.

Theorem 3: If adversary A_1 achieves a non-negligible advantage in the session key security game, it be leveraged to solve the ECCDHP. Here, q_1, q_2, and q_3 represent the frequencies of Part of the private, Private key, and Send queries, respectively. Additionally, e denotes the natural logarithmic base.

Proof: Adversary C acquires the relevant public parameters (q, P, G) and the challenge tuple $\left(P, x_i P, x_j P\right)$ from the challenger of the ECCDHP. After running the initialization algorithm, C sends the public parameters $params = \langle q, P, G, P_{pub}, H_0, H_1, H_2, H_3, H_4\rangle$ to A_1, while secretly storing the primary private key. Here, H_0, H_1, H_2 and H_3 represent one-way hash functions, and H_4 serves as a random oracle machine. C maintains three initially empty lists: L_{pub}, L_{sk} and L_d, which respectively store queries related to public keys, private keys, and parts of the private key. Additionally, C uses list L_s to track identity authentication requests submitted by A_1. Furthermore, C randomly selects RID_i^* and RID_j^* as challenge identities.

Part of the private key, Private key-query, Public key-query and *Public key substitution* queries are similar to those described in **Theorem 1.**

Send-query: In this game, A_1 can send two types of Send queries.

$Send\left(\Pi_{i,j}^k, start\right)$. If $RID_i = RID_i^*$, then let $X_i = x_i P$. Otherwise, x_i' is randomly selected from Z_q^*, and $X_i = x_i' P$ is calculated. Then, C obtains the private key (u_i, d_i) for RID_i via the private key generation query, calculates $M_1 = \{PID_i, X_i, U_i, Z_i, \sigma_i, T_i\}$ according to the protocol description, returns it to A_1, and updates L_s.

$Send\left(\Pi_{j,i}^k\left(\sigma, X_j, T\right)\right)$. If $RID_j = RID_j^*$, then let $X_j = x_j P$. Otherwise, if $x_j \in Z_q^*$ is randomly selected, calculate $X_j = x_j P$. If $RID_i \neq RID_i^*$ and $RID_j \in L_{sk}$, C retrieves the corresponding private key from the appropriate list. Otherwise, the private key is obtained by generating a query with RID_j as input. Then it calculates $\sigma_j = d_j + u_j + x_j \sigma_j \bmod q$, where $\alpha_j = h_2\left(PID_j, Z_j, X_j, T_j\right)$. C verifies that the equation $\sigma_j P = Z_j + P_{pub} h_1\left(U_j, Z_j\right) + U_j + X_j h_1\left(PID_j, Z_j, X_j, T_j\right)$ holds true. If not, terminate the session. Otherwise, C calculates and returns the response to A_1, while updating the list L_s.

$Reveal\left(\Pi_{i,j}^k\right)$. If a tuple corresponding to $\Pi_{i,j}^k$ exists in L_s, C accepts the session and returns SK. Otherwise, it returns $\perp$.

$Test\left(\Pi_{i,j}^t\right)$. After receiving the query, C selects $b \in \{0, 1\}$. If $b = 1$, C outputs the session key that appears in the $Reveal\left(\Pi_{i,j}^k\right)$ query. Otherwise, C provides a random integer of the same length as the session key to A_1.

Defining event E_1 means that the game does not terminate when adversary A_1 makes inquiries, such as partial private key generation and private key generation, in the above games. E_2 indicates that the game does not terminate when the *Send-query* is made, and E_3 indicates that the game does not terminate when A_1 makes the *Test-query*. Therefore, there are $Pr\left(E_1\right) = \left(1 - \frac{1}{q_1+q_2+1}\right)^{q_1+q_2}$, $Pr\left(E_2\right) \geq \varepsilon_3$, $Pr\left(E_3\right) = \frac{1}{q_3(q_3-1)}$, then it can be inferred that $\varepsilon_3' = Pr\left(E_1 \wedge E_2 \wedge E_3\right) \geq \frac{\varepsilon_3}{eq_3(q_3-1)}$.

To sum up, if A_1 wins the session key security game with a non-negligible advantage ε_3', then challenger C can be constructed to solve the ECCDHP with an obvious advantage $\varepsilon_3' \geq \frac{\varepsilon_3}{eq_3(q_3-1)}$.

Theorem 4: Assuming that adversary A_2 is capable of forging a valid identity authentication response with non-negligible advantage. it implies the existence of an challenger C that can solve the ECDLP with non-negligible advantage.

Proof: C receives the public parameters (q, P, G) and the tuple (P, sP) from the challenger of the ECDLP. It then runs the initialization algorithm to generate the corresponding public parameters and primary private key. C sends the public parameters $params = \langle q, P, G, P_{pub}, H_0, H_1, H_2, H_3, H_4\rangle$ to A_2. Here, H_0, H_1, H_2 and H_3 are one-way hash functions, and H_4 is a random oracle machine. C maintains four initially empty lists L_{H_4}, L_{pub}, L_{sk} and L_d, which store the queries of H_4 oracle machine, public keys, private keys and part of the private keys. Additionally, it uses the list L_s to record the identity authentication requests submitted by A_2. C randomly selects RID_i^* as the challenge identity.

H_4-query. Given a query of the form (RID_i, X_i, Z_i, T_i). If $\left(RID_i, X_i, Z_i, T_i, H_i'\right) \in L_{H_4}$, then C returns H_i' to A_2. Otherwise, C randomly selects H_i' from the range of H_4 and returns H_i'. Subsequently, C adds the tuple$\left(RID_i, X_i, Z_i, T_i, H_i'\right)$ to the list L_{H_4}.

Private key-query. The input to the query is RID_i. If $(RID_i, u_i, d_i) \in L_{sk}$, C returns (u_i, d_i) to A_2. Otherwise, C performs the following operations. If $RID_i = RID_i^*$, C terminates. Otherwise, C randomly selects $u_i, z_i \in Z_q^*$, calculates $U_i = u_i P$, $Z_i = z_i P$. It returns (u_i, d_i), adds tuples (RID_i, u_i, d_i), and (RID_i, U_i, Z_i) to the lists L_{sk} and L_{pub}, respectively.

Public key-query. The input to the query is RID_i. If $(RID_i, U_i, Z_i) \in L_{pub}$, then C returns (U_i, Z_i) to A_2. Otherwise, C proceeds as follows. If $RID_i = RID_i^*$, C randomly selects $u_i^* \in Z_q^*$, calculates $U_i^* = u_i^* P$, returns (U_i, Z_i) to A_2, and adds $\left(RID_i^*, U_i^*, Z_i^*\right)$ and $\left(RID_i^*, u_i^*, \perp\right)$ to L_{pub} and L_{sk}, respectively. Otherwise, C randomly selects $u_i, z_i \in Z_q^*$, calculates $U_i = u_i P$, $Z_i = z_i P$ and $d_i = z_i + sh_1\left(U_i, Z_i\right) \bmod q$, returns (u_i, d_i). It also adds tuples (RID_i, u_i, d_i) and (RID_i, U_i, Z_i) to the list L_{sk} and L_{pub}, respectively.

Send-query is similar to **Theorem 1**.

Theorem 5: If adversary A_2 can impersonate the platoon head vehicle and forge a legitimate identity authentication response message with a non-negligible advantage, then there exists C can solve the ECCDHP with a significant advantage.

Theorem 6: If adversary A_2 can win the session key security game with a non-negligible advantage, then C can solve the ECCDHP with a significant advantage. Note: The proof structure of **Theorem 6** closely mirrors that of **Theorem 3** and is therefore omitted to avoid redundancy.

In the security analyses of **Theorem 2** and **Theorem 3**, C is granted the ability to supply adversary A_2 with both the primary key and individual private keys as required. Therefore, **Theorem 5** and **Theorem 6** can be similar to **Theorem 2** and **Theorem 3**.

C. Informal Security Analysis

The proposed EEGAP protocol is designed to resist a wide spectrum of security threats and meets various essential security requirements. This section presents an informal analysis of its security properties, demonstrating its robustness against common security attacks.

1) Identity Anonymity: In EEGAP, the real identity RID_i of vehicle V_i is concealed within the pseudonym $PID_i = RID_i \oplus h_3(x_i P_{pub}, Z_i, T_i)$. An adversary attempting to retrieve RID_i would need to compute $RID_i = PID_i \oplus h_3(x_i s P, Z_i, T_i)$. However, without access to the x_i, s, recovering RID_i would require solving the ECDLP, which is computationally infeasible. Therefore, the protocol ensures strong identity privacy.

2) Message Authentication: As proven in **Theorem 1**, no PPT adversary can forge valid messages signature that satisfies the verification equation $\sigma_i P = Z_i + P_{pub} h_1(U_i, Z_i) + U_i + X_i \alpha_i$. This guarantees the authenticity and integrity of messages exchanged within EEGAP.

3) Traceability and Revocation: The TA is the only entity capable of recovering the true identity RID_i from a pseudonym PID_i via the equation $RID_i = PID_i \oplus h_3\left(x_i P_{pub}, Z_i, T_i\right)$, where $Z_i = z_i P$, $x_i \in Z_q^*$. This ensures that, in the event of misbehavior or disputes, misbehaving vehicles can be effectively traced and revoked by the TA.

4) Forward Security: If a session key $SK_{ij} = SK_{ji} = h_3(x_i x_j P, PID_i, PID_j)$ negotiated between the platoon head vehicle V_i and V_j is compromised, the attacker cannot retroactively compute previous session keys without knowledge of the one-time random values x_i and x_j. Given the difficulty of the ECCDHP, this property is preserved. The same reasoning applies to the group key in the platoon-splitting phase, ensuring forward secrecy in both EEGAP-PF and EEGAP-PS.

5) Session Key Agreement: In EEGAP-PF, the two platoon head vehicles V_i and V_j independently compute $SK_{ij} = h_3(x_i X_j, PID_i, PID_j)$ and $SK_{ji} = h_3(x_j X_i, PID_i, PID_j)$, without the real-time involvement of the TA in the authentication and key agreement phase. Due to $x_i X_j = x_j X_i = x_i x_j P$, V_i and V_j hold the same session key $SK = SK_{ij} = SK_{ji}$. In EEGAP-PS, each platoon member reconstructs the polynomial $f(x)$ using distributed K shares along with the self-generated points $\left\{PID_f, \left(sk_{fi}, T_f\right)\right\}$, and calculates the group session key as $GK_i = f(0)$. Hence, EEGAP could finish session key agreement.

6) Resist Replay Attack: To ensure freshness and prevent replay attacks, each message includes unique random values x_i, z_i and x_j and timestamps T_i, T_j. Even if an attacker replays previously transmitted messages M_1 and M_2, they cannot compute the current session key without solving ECCDHP. Thus, the EEGAP is protected from replay attack.

7) Resist Man-in-Middle Attack: Suppose an adversary poses as a platoon head vehicle V_j to perform a man-in-middle attack and to generate a valid signature σ_j on an intercepted message M_2, which is impossible due to the mentioned **Theorem 1**. Similarly, it can be concluded that an adversary cannot impersonate a legitimate vehicle V_i to generate a signature σ_i. Hence, the EEGAP is secure against a man-in-middle attack.

TABLE III

FUNCTIONAL COMPARISON AMONG VARIOUS SCHEMES

Features	[34]	[35]	[13]	[36]	[37]	EEGAP
Anonymous authentication	✓	×	✓	✓	✓	✓
Resist replay attack	✓	✓	✓	✓	✓	✓
Resist man-in-the-middle attack	✓	✓	✓	✓	✓	✓
Resist impersonation attack	✓	✓	✓	✓	✓	✓
Keep group key security under the CK-adversary model	×	×	×	×	×	✓
No pre-storage of pseudonyms and private keys in vehicles	×	×	×	×	×	✓

8) Resist Impersonation Attack: To impersonate a platoon head vehicles V_i or V_j, an adversary must construct a valid message M_1. However, In EEGAP, each vehicle periodically updates its pseudonym $PID_i = RID_i \oplus h_3\left(x_i P_{pub}, Z_i, T_i\right)$ (valid for T_i), where x_i is a random number selected by V_i, and s is the system's primary secret key. Without knowledge of x_i, RID_i, s, an adversary cannot reconstruct a valid pseudonym or signature that passes verification. Hence, the protocol effectively prevents impersonation, even in dynamic vehicular environments.

9) Resist Ephemeral Secret Leakage (ESL) Attack: In the proposed protocol, the group session key SK is derived from both short-term secrets $\left(x_i, x_j\right)$ and long-term secrets (u_i, u_i, d_i, d_i, s). Even if an adversary compromises the short-term ephemeral secrets, it must also obtain the corresponding long-term private values and the primary key to successfully impersonate or compute the session key SK. Since acquiring both sets of secrets simultaneously is computationally infeasible under the hardness of ECCDHP, the protocol provides strong resilience against ESL attacks.

VII. SIMULATION AND PERFORMANCE ANALYSIS

This section presents both theoretical and simulation-based evaluations to validate the performance and effectiveness of the proposed EEGAP scheme. The theoretical analysis mainly involves four aspects: functionality comparison, computation burden, communication burden and platoon serving capability. The performance is further verified through simulation using NS-3 and SUMO, while result analysis is conducted in MATLAB.

A. Theoretical Analysis

- **Comparison of Functionality**

All schemes were compared based on their design objectives and security analyses [13], [34], [35], [36], [37]. The results are shown in Table III, where ✓ and × indicate whether the scheme meets or does not meet the functionality. EEGAP satisfies all evaluated security properties, whereas other schemes exhibit vulnerabilities or lack certain functionalities, thereby rendering them less robust in practical deployment scenarios.

- **Comparison of Computation Burden**

The cryptographic runtime measurements are based on the hardware platform equipped with an Intel Core i5-8300 processor (2.30GHz), 16GB RAM, running Windows 10.

TABLE IV
CRYPTOGRAPHIC OPERATION AND EXECUTION TIMES

Notation	Description of cryptographic operations	Execute time (ms)
T_{ecc}^{m}	Time of scalar multiplication operation based on ECC	1.252
T_{ecc}^{a}	Time for point addition operation based on elliptic curve cryptography	0.009
T_h	Time of one-way hash operation	0.003
T_{bp}	Execution time of bilinear pairing operation	6.165
T_{bp}^{m}	Execute time of bilinear pairing-based multiplication operations	1.100
T_{ex}	Exponentiation operation in G_1	0.018
T_{mtp}	The time to execute the MapToPoint	0.082

Table IV defines the runtime of several cryptographic operations and the notation for the operation execution time. Simple code for calculating the execution time of some cryptographic operations is attached to the GitHub project link: https://github.com/wzh199808/EEGAP.

In this paper, the computation overhead is compared with that of [13], [34], [35], [36], and [37]. EEGAP supports two primary scenarios: vehicular platoon fusion (EEGAP-PF) and vehicular platoon splitting (EEGAP-PS). In EEGAP-PF, the total computation time required for mutual authentication and session key negotiation between two platoon head vehicles is $6T_{ecc}^{m} + 2T_{ecc}^{a} + 5T_h = 7.545ms$, while in EEGAP-PS, the computation cost for the platoon head vehicle to negotiate a group key with its members is $T_{ecc}^{m} + T_h = 1.255ms$. In our scheme, group key session key negotiation in both vehicular platoon fusion and vehicular platoon splitting does not require the assistance of infrastructure such as FNs and RSUs. The computation overhead of related protocols [13], [34], [35], [36], [37] is also computed in the same way.

From the perspective of computation cost, the two scenarios involved in our scheme are significantly superior to all other schemes, as depicted in Fig.3. Mutual authentication, as well as response delays during key negotiation, are monitored. Fig.4 illustrates that the corresponding delay increases linearly with the message, yet a minimal computation burden is maintained by the EEGAP. Specifically, EEGAP outperforms other related schemes [13], [34], [35], [36], [37] by $\frac{26.318n-7.545n-1.255n}{26.318n} \approx 66.56\%$, $\frac{18.582n-7.545n-1.255n}{18.582n} \approx 52.64\%$, $\frac{9.471n-7.545n-1.255n}{9.471n} \approx 7.08\%$, $\frac{11.34n-7.545n-1.255n}{11.34n} \approx 22.40\%$ and $\frac{18.547n-7.545n-1.255n}{18.547n} \approx 52.55\%$, respectively. Obviously, our scheme achieves better performance.

- **Comparison of Communication Burden**

EEGAP employs cryptographic parameters based on a bilinear pairing algorithm and elliptic curve algorithm with an 80-bit security level. Specifically, let G_1 be an additive group generated by the q'-order point P on a supersingular elliptic curve $E : y^2 = x^3 + ax + b \bmod q$ with a degree of embedding 2, where p' and q' are 512-bit and 160-bit prime numbers, respectively. Similarly, construct a similar q-order G as an additive group on a non-singular elliptic curve, where p and q are $160bits$. Therefore, $|G_1| = 1024bits$ and $|G| = 320bits$. Parameter sizes used in computation include: identity length $|I| = 128bits$, timestamp $|T| = 32bits$, hash output $|h()| = 256bits$, random numbers $\left|Z_q^*\right| = 160bits$, and Advanced Encryption Standard (AES) key length $|AES| = 256bits$.

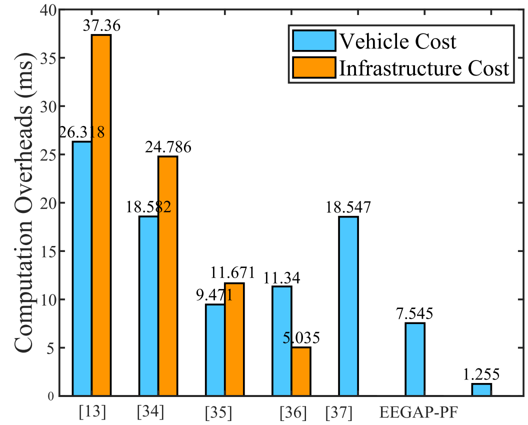


Fig. 3. Computation delay for authentication and negotiation of a single session key.

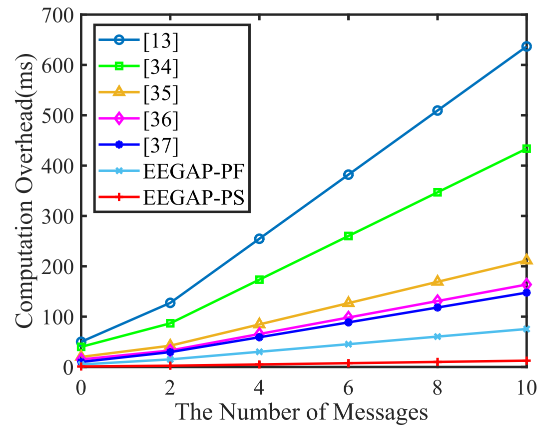


Fig. 4. Computation cost comparison among different schemes.

To measure the communication cost of the protocol, we have only considered the messages transmitted during the negotiation of authenticated keys in the vehicular platoon. In our scheme, during the vehicle authentication negotiation phase, vehicle V_i transmits $\{PID_i, X_i, U_i, Z_i, \sigma_i, T_i\}$ to vehicle V_j, and then vehicle V_j returns $\left\{PID_j, X_j, U_j, Z_j, \sigma_j, T_j\right\}$ to vehicle V_i. Here, $X_i, U_i, Z_i \in G$, $\sigma_i, \sigma_j \in Z_q^*$, and T_i, T_j are timestamps. Hence, the total communication cost is calculated as $3|G| + 2|I| + 2\left|Z_q^*\right| = 2176bits$.

Fig. 5 illustrates that EEGAP incurs the lowest communication burden among the evaluated protocols. As depicted in Fig.6, communication cost scales linearly with message volume, yet remains consistently lower than the alternatives. Specifically, the EEGAP outperforms other related schemes [13], [34], [35], [36], [37] by $\frac{3296n-2176n}{3296n} \approx 33.98\%$, $\frac{4224n-2176n}{4224n} \approx 48.48\%$, $\frac{4128n-2176n}{4128n} \approx 47.29\%$, $\frac{2336n-2176n}{2336n} \approx 6.85\%$ and $\frac{2432n-2176n}{2432n} \approx 10.53\%$, respectively. Nonetheless, the communication overhead of the proposed protocol is less than all related works [13], [34], [35], [36], [37].

- **Platoon Serving Capability**

To evaluate the performance of the platoon head vehicle that undertakes the communication of the entire platoon, this subsection first defines the time spent by the platoon head vehicle to process a single message as T_{gen}. The T_{gen}

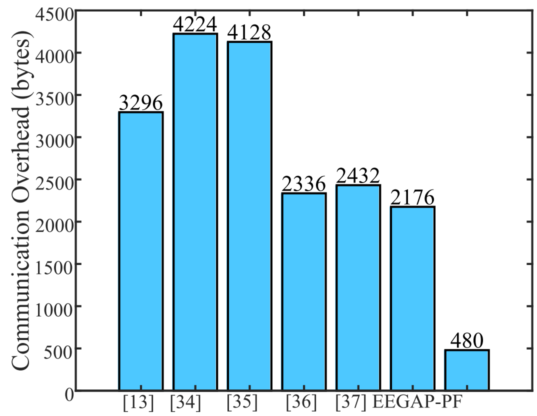


Fig. 5. Communication cost comparison.

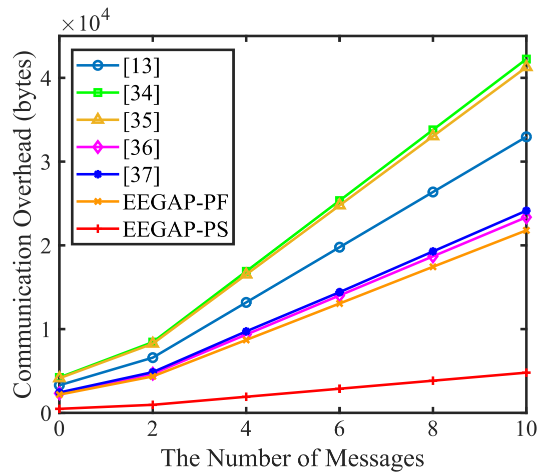


Fig. 6. Communication overheads comparison.

in the scenarios (EEGAP-PF and EEGAP-PS) involved in our scheme are $T_{gen} \approx 3.7620ms$ and $T_{gen} \approx 1.2550ms$, respectively. Let n denote vehicle density ($200 \sim 400$), p the message send probability, v the vehicle speed ($5m/s \sim 10m/s$), and r the communication range of platoon. Thus, platoon serving capability $P_{ser} = \frac{p \cdot r}{v \cdot T_{gen} \cdot n}$.

Fig. 7 indicates that the service capacity of platoon gradually declines with increasing vehicle density and speed. Nevertheless, EEGAP is capable of supporting 240 messages within every $300ms$ interval, indicating a very low packet loss rate even under high-density traffic conditions.

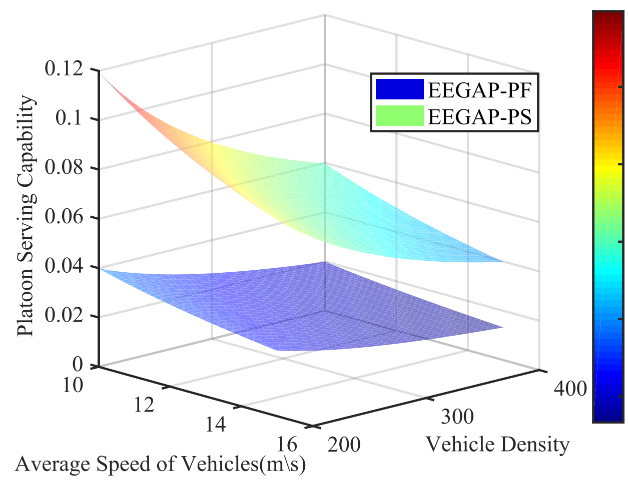


Fig. 7. Service capacity of platoon in the EEGAP for different vehicle densities and different average vehicle speeds, platooning scope $r = 300m$.

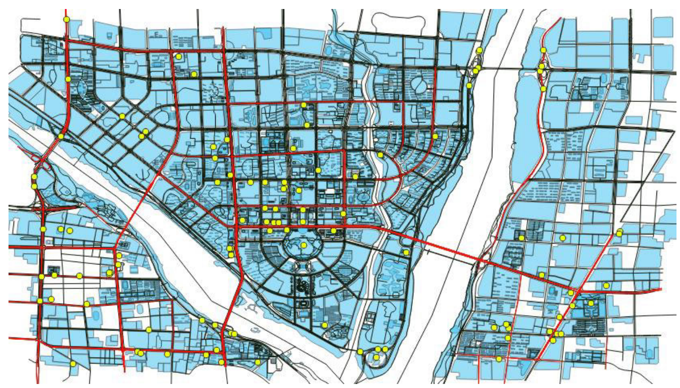


Fig. 8. Simulation area map with a range of $9km \times 5km$ in Linyi, China. Vehicular platoons of fixed size are represented by yellow dots.

TABLE V

CRYPTOGRAPHIC OPERATION AND EXECUTION TIMES

Parameter	**Value**
MAC Layer	IEEE 802.11p
Routing Protocol	AODV
Radio Range	$300m$
Data Rate	$6Mbps$
Number of Platoons	$20 \sim 100$
Simulation Time	527/300 Second per each run
Area	$9km \times 5km$
Vehicle Speed	$20m/s$

B. Simulation Results Analysis

The network performance of all protocols is simulated using the open-source simulation platforms NS-3 3.27 and SUMO 1.8. Road topology and vehicular platoon movement traces are generated by Open Street Map (OSM) and SUMO respectively. Fig. 8 shows a map of the simulated area with a range of $9km \times 5km$, which is a real traffic environment located in Linyi, China. The communication model adopts the IEEE 802.11p standard with a channel capacity of $6Mbps$ and a communication range of $300m$. The vehicular platoon is assumed to have a fixed size, and the total number of platoons varies from 20 to 100. The simulation parameters used are summarized in Table V. Moreover, to evaluate the network performance, we consider the following metrics.

- **End-to-end Packet Delay**

The end-to-end delay reflects the time taken for a packet to be transmitted from the source vehicular platoon to the destination platoon. We compared the end-to-end delay of the EEGAP with that of existing protocols [13], [34], [35], [36], [37] under varying platoons densities, maintaining a constant vehicle velocity of $20m/s$. As shown in Fig. 9, the end-to-end packet delay increases with the number of platoons. This is because, as the platoons grows, the packet size transmitted within vehicular platoon systems also increases, leading to higher transmission delays. The experimental results in Fig. 9 indicate that EEGAP achieves the lowest end-to-end delay. This is primarily due to the reduced computation and communication overhead in both packet generation and reception, making our scheme more efficient than the alternatives.

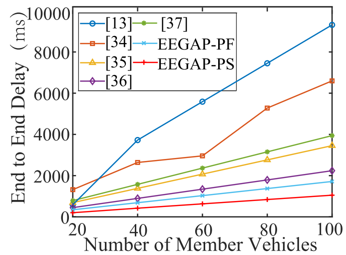


Fig. 9. Comparison of end to end delay under different vehicular platoons density.

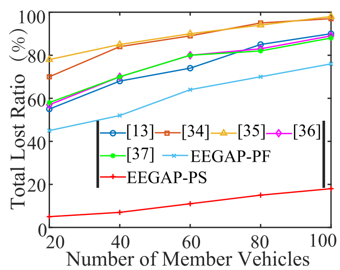


Fig. 10. Comparison of total lost ratio under different vehicular platoons density.

- **Packet Loss Ratio**

The packet loss ratio is defined as the proportion of lost packets relative to the total number of packets transmitted during vehicular platoon communications. A higher packet loss ratio reflects increased transmission failures, typically resulting from channel contention, collisions, or network congestion. As depicted in Fig. 10, the packet loss ratio increases proportionally with the number of platoons for all evaluated protocols. This trend is attributable to the increased packet transmission volume, which raises the probability of collision and contention in the shared communication medium. Nevertheless, EEGAP exhibits a significantly lower packet loss ratio compared to other schemes [13], [34], [35], [36], [37]. This improvement is primarily due to EEGAP's lightweight authentication mechanism and efficient message processing, which collectively reduce network congestion and minimize transmission delays.

VIII. Conclusion and Outlooks

This paper proposes an efficient ECC-based group authentication protocol (EEGAP), specifically designed for dynamic vehicular platoon environments. EEGAP addresses critical challenges associated with platoon structure adjustments, including platoon fusion and splitting, thereby enabling secure, scalable, and seamless intra-platoon communication in dynamic scenarios. Unlike existing schemes, EEGAP combines anonymous authentication with a robust group key agreement mechanism that guarantees message integrity, non-repudiation and traceability, while significantly reducing communication and computation overhead. Simulation results obtained using NS-3 and SUMO demonstrate that EEGAP achieves over 7.08% reduction in computation cost and more than 6.85% reduction in communication overhead compared to state-of-the-art schemes. Comprehensive security analysis and performance evaluations confirm that EEGAP meets stringent security and practicality requirements for vehicular platoon communication systems. Future work will focus on further improving the protocol's scalability and responsiveness to support large-scale deployments, with an emphasis on lightweight communication mechanisms for enhanced real-time performance.

References

[1] Y. Li, Y. Zhao, and S. Tong, "Adaptive fuzzy control for heterogeneous vehicular platoon systems with collision avoidance and connectivity preservation," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 11, pp. 3934–3943, Nov. 2023.

[2] Y. Zhou, Z. Cao, X. Dong, and J. Zhou, "BLDSS: A blockchain-based lightweight searchable data sharing scheme in vehicular social networks," *IEEE Internet Things J.*, vol. 10, no. 9, pp. 7974–7992, May 2023.

[3] J.-W. Kwon and D. Chwa, "Adaptive bidirectional platoon control using a coupled sliding mode control method," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2040–2048, Oct. 2014.

[4] J. Boo and D. Chwa, "Integral sliding mode control-based robust bidirectional platoon control of vehicles with the unknown acceleration and mismatched disturbance," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10881–10894, Oct. 2023.

[5] Q. Kong, L. Su, and M. Ma, "Achieving privacy-preserving and verifiable data sharing in vehicular fog with blockchain," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4889–4898, Aug. 2021.

[6] Z. Qiao, Q. Yang, Y. Zhou, and M. Zhang, "Improved secure transaction scheme with certificateless cryptographic primitives for IoT-based mobile payments," *IEEE Syst. J.*, vol. 16, no. 2, pp. 1842–1850, Jun. 2022.

[7] B. L. Nguyen, D. T. Ngo, N. H. Tran, M. N. Dao, and H. L. Vu, "Dynamic V2I/V2V cooperative scheme for connectivity and throughput enhancement," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1236–1246, Feb. 2022.

[8] L.-W. Chen and H.-M. Chen, "Driver behavior monitoring and warning with dangerous driving detection based on the Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7232–7241, Nov. 2021.

[9] Y. Xia, Y. Liu, S. Dong, M. Li, and C. Guo, "SVCA: Secure and verifiable chained aggregation for privacy-preserving federated learning," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 18351–18365, May 2024.

[10] J. Chen et al., "Industrial blockchain threshold signatures in federated learning for unified space-air-ground-sea model training," *J. Ind. Inf. Integr.*, vol. 39, May 2024, Art. no. 100593.

[11] C.-K. Wu, "A privacy-preserving group encryption scheme with identity exposure," *Frontiers Comput. Sci.*, vol. 16, no. 5, Oct. 2022, Art. no. 165823.

[12] X. Li and X. Yin, "Blockchain-based group key agreement protocol for vehicular ad hoc networks," *Comput. Commun.*, vol. 183, pp. 107–120, Feb. 2022.

[13] X. Feng, Q. Shi, Q. Xie, and L. Wang, "P2BA: A privacy-preserving protocol with batch authentication against semi-trusted RSUs in vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3888–3899, 2021.

[14] J. Wang, N. Wang, H. Wang, K. Cao, and A. M. El-Sherbeeny, "GCP: A multi-strategy improved wireless sensor network model for environmental monitoring," *Comput. Netw.*, vol. 254, Dec. 2024, Art. no. 110807.

[15] B. Harishma et al., "Safe is the new smart: PUF-based authentication for load modification-resistant smart meters," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 663–680, Jan. 2022.

[16] A. Mansour, K. M. Malik, A. Alkaff, and H. Kanaan, "ALMS: Asymmetric lightweight centralized group key management protocol for VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1663–1678, Mar. 2021.

[17] X. Zhang, H. Zhong, C. Fan, I. Bolodurina, and J. Cui, "CBACS: A privacy-preserving and efficient cache-based access control scheme for software defined vehicular networks," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1930–1945, 2022.

[18] J. Zhang, H. Zhong, J. Cui, M. Tian, Y. Xu, and L. Liu, "Edge computing-based privacy-preserving authentication framework and protocol for 5G-enabled vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7940–7954, Jul. 2020.

[19] J. Cui, X. Zhang, H. Zhong, J. Zhang, and L. Liu, "Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1654–1667, 2020.

[20] A. Yang, J. Weng, K. Yang, C. Huang, and X. Shen, "Delegating authentication to edge: A decentralized authentication architecture for vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1284–1298, Feb. 2022.

[21] N. Alangudi Balaji, R. Sukumar, and M. Parvathy, "Enhanced dual authentication and key management scheme for data authentication in vehicular ad hoc network," *Comput. Electr. Eng.*, vol. 76, pp. 94–110, Jun. 2019.

[22] P. Liu, A. Kurt, and U. Ozguner, "Distributed model predictive control for cooperative and flexible vehicle platooning," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1115–1128, May 2019.

[23] Q. Zhang et al., "A group key agreement protocol for intelligent Internet of Things system," *Int. J. Intell. Syst.*, vol. 37, no. 1, pp. 699–722, Jan. 2022.

[24] Q. Zhang et al., "A hierarchical group key agreement protocol using orientable attributes for cloud computing," *Inf. Sci.*, vol. 480, pp. 55–69, Apr. 2019.

[25] L. Wei, J. Cui, H. Zhong, Y. Xu, and L. Liu, "Proven secure tree-based authenticated key agreement for securing V2V and V2I communications in VANETs," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3280–3297, Sep. 2022.

[26] Y. Zhao, Y. Wang, Y. Liang, H. Yu, and Y. Ren, "Identity-based broadcast signcryption scheme for vehicular platoon communication," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7814–7824, Jun. 2023.

[27] S. K. Dwivedi, R. Amin, S. Vollala, and A. K. Das, "Design of blockchain and ECC-based robust and efficient batch authentication protocol for vehicular ad-hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 1, pp. 275–288, Jan. 2024.

[28] S. Yu et al., "Efficient ECC-based conditional privacy-preserving aggregation signature scheme in V2V," *IEEE Trans. Veh. Technol.*, vol. 72, no. 11, pp. 15028–15039, Nov. 2023.

[29] I. Ali, Y. Chen, N. Ullah, R. Kumar, and W. He, "An efficient and provably secure ECC-based conditional privacy-preserving authentication for vehicle-to-vehicle communication in VANETs," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1278–1291, Feb. 2021.

[30] D. Boneh and V. Shoup, "A graduate course in applied cryptography," IETF, Internet-Draft draft 0.5, Jan. 2020.

[31] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[32] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Apr. 2002, pp. 337–351.

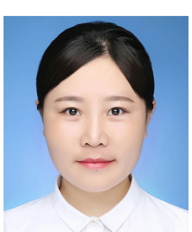
[33] H. Cheng, X. Zhang, J. Yang, and Y. Liu, "PPRT: Privacy preserving and reliable trust-aware platoon recommendation scheme in IoV," *IEEE Syst. J.*, vol. 17, no. 3, pp. 4922–4933, Sep. 2023.

[34] P. Vijayakumar, M. Azees, S. A. Kozlov, and J. J. P. C. Rodrigues, "An anonymous batch authentication and key exchange protocols for 6G enabled VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1630–1638, Feb. 2022.

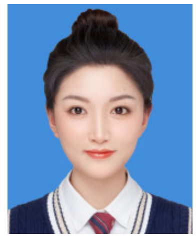
[35] C. Wang, J. Shen, J.-F. Lai, and J. Liu, "B-TSCA: Blockchain assisted trustworthiness scalable computation for V2I authentication in VANETs," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1386–1396, Jul. 2021.

[36] J. Miao, Z. Wang, X. Ning, A. Shankar, C. Maple, and J. J. P. C. Rodrigues, "A UAV-assisted authentication protocol for Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 8, pp. 10286–10297, Aug. 2024.

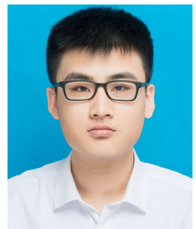
[37] C. Maurya and V. K. Chaurasiya, "Efficient anonymous batch authentication scheme with conditional privacy in the Internet of Vehicles (IoV) applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 9670–9683, Sep. 2023.



Hongyuan Cheng received the Ph.D. degree from the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China, in 2023. She is currently an Associate Professor with the School of Computer Science and Engineering, Linyi University, Linyi, China. Her research primarily focuses on reputation evaluation and privacy preservation in IoV.



Zihan Wang received the B.E. degree from the School of Medical Information Engineering, Shandong First Medical University, Tai'an, China, in 2022, and the master's degree from the School of Information Science and Engineering, Linyi University, Linyi, China, in 2025. Her main research interests include anonymous authentication and privacy preservation in vehicular networks.

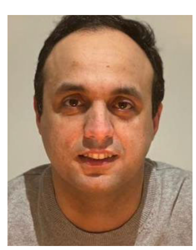


Jingcheng Song (Member, IEEE) received the Ph.D. degree from the School of Information and Communication, Guilin University of Electronic Technology, Guilin, China. In 2022, he joined the School of Information Science and Engineering, Linyi University. His research interests include data privacy protection, information security, and federated learning.

Qi Zhong received the Ph.D. degree from Deakin University, Australia, in 2023. She is currently an Assistant Professor at the City University of Macau. Her research focuses on deep learning model copyright protection, machine learning backdoor attacks and defenses, and data privacy protection.

Zahra Pooranian (Senior Member, IEEE) received the Ph.D. degree in computer science from the Sapienza University of Rome, Italy, in February 2017. She is currently a Lecturer in computer science at the University of Reading, U.K. She has contributed to PASSCOINFOG (funded by Huawei), AUTOTRUST (Funded by European Space Agency), and National Privacy Project (funded by Alan Turing), U.K.

Fabio Martinelli (Senior Member, IEEE) received the M.Sc. degree from the University of Pisa, Pisa, Italy, in 1994, and the Ph.D. degree from the University of Siena, Siena, Italy, in 1999. He is currently the Research Director of the Consiglio Nazionale delle Ricerche (CNR), Pisa, where he leads the Cyber Security Project area. He is involved in several steering committees of international WGs/conferences and workshops. His main research interests include security/privacy in distributed and mobile systems and the foundations of security/trust.



Mohammad Shojafar (Senior Member, IEEE) received the Ph.D. degree (Hons.) from the Sapienza University of Rome, Rome, Italy, in 2016. He is an Associate Professor in network security and an Intel Innovator, a Professional ACM Member, a fellow of the Higher Education Academy, and a Marie Curie alumnus, working at the 5G and 6G Innovation Centre (5G/6GIC), University of Surrey, U.K. He has secured approximately two million in research funding as the Principal Investigator for various EU- and U.K.-funded projects. He is an Associate Editor of IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and IEEE TRANSACTIONS ON CONSUMER ELECTRONICS.