A Lightweight Batch Authenticated Key Agreement Scheme Based on Fog Computing for VANETs

Lihui Li , Huacheng Zhang , Song Li , Jianming Liu * , Chi Chen *

Posted Date: 2 July 2025

doi: 10.20944/preprints202507.0134.v1

Keywords: VANETs; Authenticated Key Agreement; Fog Nodes; Lagrange Interpolation Formula

 doi:10.20944/preprints202507.0134.v1

Article

A Lightweight Batch Authenticated Key Agreement Scheme Based on Fog Computing for VANETs

Lihui Li [1], Huacheng Zhang [1], Song Li [1], Jianming Liu [1,*] and Chi Chen [2,*]

[1] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
[2] School of data science and artifical intelligence, Wenzhou University of Technology, Wenzhou 325027, China
* Correspondence: Jianming Liu jmliu@guet.edu.cn; Chi Chen chen_chi@189.cn

Abstract

In recent years, fog-based vehicular ad-hoc networks (VANETs) has become a hot research topic. Due to the inherent insecurity of open wireless channels between vehicles and fog nodes, establishing session keys through authenticated key agreement (AKA) protocols is critically important for securing communications. However, existing AKA schemes face several critical challenges: (1) When a large number of vehicles initiate AKA requests within a short time window, existing schemes that process requests one by one individually incur severe signaling congestion, resulting in significant quality of service degradation. (2) Many AKA schemes incur excessive computational and communication overhead due to the adoption of computationally intensive cryptographic primitives (e.g., bilinear pairings and scalar multiplications on elliptic curve groups) and unreasonable design choices, making them unsuitable for the low-latency requirements of VANETs. To address these issues, we propose a lightweight batch AKA scheme based on fog computing. In our scheme, when a group of vehicles requests AKA sessions with the same fog node within the set time interval, the fog node aggregates these requests and, with assistance from the traffic control center, establishes session keys for all vehicles by a round of operations. It has significantly reduced the operational complexity of the entire system. Moreover, our scheme employs Lagrange interpolation and lightweight cryptographic tools, thereby significantly reducing both computational and communication overhead. Additionally, our scheme supports conditional privacy preservation and includes a revocation mechanism for malicious vehicles. Security analysis demonstrates that the proposed scheme meets the security and privacy requirements of VANETs. Performance evaluation indicates that our scheme outperforms existing state-of-the-art solutions in terms of efficiency.

Keywords: VANETs; authenticated key agreement; fog nodes; lagrange interpolation formula

1. Introduction

With the rapid proliferation of private vehicles and increasingly complex road traffic networks, traffic management faces significant challenges, resulting in frequent accidents and substantial losses of life and property. Meanwhile, modern vehicles have transcended their traditional role as mere transportation tools, evolving into mobile intelligent spaces that integrate travel, leisure, entertainment, and work functions. These factors collectively highlight the critical importance of establishing an intelligent traffic management platform. In recent years, the swift advancement of information technology has positioned vehicular ad-hoc networks (VANETs) as an emerging research focus [1,2]. By enabling wireless communication among multiple entities, VANETs provide crucial technical support for traffic control, autonomous driving, safety warnings, and vehicular services, thereby becoming a cornerstone of modern intelligent transportation systems.

The foundational architecture of conventional VANETs typically comprises three core components: traffic control center (TCC), roadside units (RSU), and vehicles equipped with on-board unit (OBU) [3]. Central to this architecture, the TCC serves as both the system's centralized control hub and primary

 doi:10.20944/preprints202507.0134.v1

computation component. Its responsibilities encompass identity management, secure authentication, and security policy formulation for the entire network.

However, with the rapid advancement of VANETs and the exponential growth of smart vehicles, network data traffic has surged dramatically. Due to its centralized nature, traditional vehicular network architecture struggles to meet the requirements of high concurrency and real-time communication, often resulting in communication bottlenecks and single points of failure.

To overcome these challenges, cloud computing and fog computing-based vehicular architectures have gained significant attention in recent years [4,5]. As an extension of cloud computing, fog computing deploys computing and storage resources closer to the network edge, enabling more localized data processing. This approach effectively reduces communication latency and enhances network efficiency [6]. Furthermore, given the limited coverage of RSUs, introducing fog nodes (FNs) between TCCs and RSUs can significantly improve network stability and overall communication quality in vehicular networks.

In VANET environments, information exchange primarily relies on wireless communication technologies, such as WiFi and dedicated short-range communications (DSRC) [7,8]. However, the inherent security vulnerabilities in wireless technologies expose the entire system to both internal and external security threats, including man-in-the-middle attacks, replay attacks, and impersonation attacks. If VANETs suffer malicious attacks, they could result in serious casualties and substantial property damage. Therefore, ensuring communication security is a fundamental prerequisite for the widespread deployment of VANETs.

To ensure the security of VANETs, symmetric or asymmetric encryption is typically employed for secure data transmission. While asymmetric encryption offers robust security, its high computational and communication overhead makes it difficult to satisfy VANETs' requirements for high-frequency, low-latency communication. In contrast, the authenticated key agreement (AKA) mechanism enables the establishment of a secure session key between two communicating entities. This allows subsequent data transmission using symmetric encryption based on the session key, thereby reducing the computational and communication overhead while ensuring security.

At present, a variety of AKA schemes have been proposed [9–26], However, these schemes commonly face the following challenges: (1) When a large number of vehicles request key agreement from TCC in a short period of time, each request is processed individually between the vehicle and TCC. This paradigm incurs two critical limitations: on the one hand, TCC has too much processing pressure in a short period of time; on the other hand, the communication overhead of the whole system increases greatly, and the processing efficiency of the whole system is relatively low; (2) Some schemes rely on complex cryptographic operations, which is difficult to meet the real-time requirements in the VANETs environment.

To solve the above problems, we propose an efficient batch AKA scheme based on fog computing (BAKAF), which can improve the overall efficiency of the system on the premise of ensuring security. Our key contributions are summarized as follows:

(1) For the scenario where a group of vehicles in a local area request AKA sessions within a short time frame, we propose a comprehensive AKA scheme based on the concepts of fog computing and Lagrange interpolation. In our scheme, the FN aggregates all AKA requests and, with the assistance of the TCC, establishes session keys for all vehicles by a round of operations. This approach effectively alleviates operational complexity of the entire system. During the process of generating multiple session keys via Lagrange interpolation, a shared random point is used across the straight lines corresponding to different vehicles at the FN, thereby reducing both computational and communication overhead.

(2) Our scheme incorporates a conditional privacy preserving mechanism and supports the revocation of malicious vehicles. The storage and computational overhead associated with tracking and revoking malicious vehicles is centralized at TCC, which is beneficial for resource-constrained vehicles.

 doi:10.20944/preprints202507.0134.v1

(3) Our scheme employs cryptographic tools with low computational complexity, further reducing the overall computational overhead.

(4) Security analysis demonstrates that our scheme satisfies the security and privacy requirements of VANETs. Extensive experiments show that compared to existing advanced schemes, our approach achieves superior performance.

The remainder of this paper is organized as follows. Section 2 describes the related works. The preliminaries are presented in Section 3. Following this, the proposed scheme is detailed in Section 4. Subsequently, the security analysis and performance evaluation are presented in Sections 5 and 6 respectively. Finally, there is the conclusion of the paper.

2. Related Works

To meet the high-security requirements for identity authentication and session key exchange, numerous AKA protocols have been proposed over the past decade.

In early schemes for foundational peer-to-peer communication scenarios, AKA mechanisms often relied on public key infrastructure (PKI). Islam et al. [9] proposed a device-to-device (D2D) AKA protocol based on elliptic curve cryptography (ECC) and self-certified public keys, which simplifies certificate management while achieving lightweight authentication. However, the lack of a complete mutual authentication mechanism makes this scheme vulnerable to typical threats such as replay and blocking attacks. To further simplify key management, Dang et al. [10] introduced a more efficient identity-based AKA protocol for VANETs, using an identity-based key generation model to reduce reliance on certificates. However, Deng et al. [11] pointed out that this scheme fails to ensure forward security in key leakage scenarios and proposed an improved version to enhance robustness under asymmetric attack models.

As privacy protection and practicality demands rise, Recent studies have incorporated multi-factor authentication, anonymous authentication, Chebyshev polynomials, and elliptic curve-based lightweight cryptographic primitives into AKA protocols. For instance, Xie et al. [12] proposed a dynamic ID-based anonymous two-factor AKA scheme combining smart cards and dynamic identities, supporting anonymity and password updates. However, Li et al. [13] revealed that the scheme [12] is vulnerable to offline guessing attacks and smart card loss scenarios. Liu et al. [14] proposed a reputation-based conditional privacy-preserving AKA scheme for VANETs, which enhances authentication credibility and supports dynamic trust management. However, it suffers from high computational overhead due to its use of bilinear pairing. Lee et al. [15] introduced an improved two-factor AKA scheme based on extended chaotic maps, constructing a timestamp verification mechanism to resist replay attacks, but the protocol still faces risks of key leakage and temporary data exposure. Jiang et al. [22] proposed a cloud-based three-factor authentication and key agreement protocol (CT-AKA) integrating passwords, biometrics, and smart cards to secure access between cloud systems and vehicles. Dua et al. [16] designed a two-tier AKA scheme based on ECC, where cluster head vehicles are authenticated by a central authority in the first tier, and ordinary vehicles are authenticated by cluster head vehicles in the second tier to achieve efficient V2V communication.

With rising device density and access demands in VANETs, traditional authentication methods struggle with communication latency and key synchronization inefficiencies. To overcome these issues, researchers have designed AKA protocols supporting batch verification and one-to-many authentication. Vijayakumar et al. [17] proposed an efficient batch AKA scheme for 6G-enabled VANETs. However, their scheme requires authentication prior to key agreement and supports only batch authentication rather than batch key agreement. Sun et al. [18] combined certificateless signatures with the Chinese Remainder Theorem to design an AKA protocol with batch processing capabilities, claiming it effectively resists man-in-the-middle, impersonation, and replay attacks. Madandi et al. [19] developed a binary-tree-based AKA protocol to share authentication loads among nodes and enhance structural scalability. It should be noted that while these one-to-many AKA protocols reduce the complexity of operations, they also introduce high-computational operations such as pairing and

 doi:10.20944/preprints202507.0134.v1

modular exponentiation, which impose significant computational burdens on resource-constrained VANET environments. As a result, increasing attention has been paid to designing lightweight AKA protocols tailored for VANET environments. Researchers have recognized that vehicle terminals have limited resources, while RSUs, Trusted Authorities (TAs), and FNs possess greater computational and storage capabilities, leading to fog computing-based AKA design models. Wazid et al. [24] proposed a secure AKA protocol for fog computing-based VANETs deployments, offering resource-aware advantages. However, some studies pointed out potential vulnerabilities in the scheme [24] under simulated attacks [25]. Ma et al. [26] designed a fog-based AKA scheme proven to provide session key security and protect V2I communication without using bilinear pairings. Wei et al. [20] proposed a symmetric encryption-based conditional privacy-preserving AKA scheme for fog-based VANETs, significantly reducing communication and computational costs. Qiao et al. [21] proposed a lightweight anonymous three-party AKA protocol using Chebyshev chaotic map operations to generate a shared session key while preserving anonymity, however, due to its focus on healthcare IoT, it is not suitable for VANETs. Cui et al. [23] proposed a robust and scalable VANET authentication scheme enabling vehicles to register with a trusted authority (TA) once and subsequently achieve rapid and efficient authentication with cloud service providers. Considering the limited computational and storage capacities of vehicles and drones, Cui et al. [27] further introduced a lightweight and provably secure two-factor AKA scheme based on chaotic maps for UAV-assisted VANETs, featuring fuzzy verifiers and honeywords to resist offline guessing and honeypot attacks. Zhou et al. [28] designed a vehicle-attribute-based AKA scheme supporting multi-user simultaneous authentication, offering formal security under the Canetti-Krawczyk (eCK) model and enabling session key agreement and anonymous identity updates. However, its use of bilinear pairing results in significant computational overhead.

In summary, although existing AKA protocols have made significant progress in improving VANET communication security, several key challenges remain. First, many schemes rely on complex cryptographic operations such as bilinear pairing and group signatures, leading to high computational and communication overheads that fail to meet the latency requirements of vehicular applications. Second, most of these protocols do not support batch AKA operations, which results in high operational complexity and poor service quality when facing large-scale AKA requests in localized and short-time scenarios.

3. Preliminaries

In this section, we introduce some basic knowledge, including system model, pseudo random function, elliptic curve cryptography, security and privacy requirements. The important symbols and definitions used in subsequent sections are shown in Table 1.

 doi:10.20944/preprints202507.0134.v1

Table 1. Definition Basic symbols

Symbol	Definition
TCC	Traffic control center
RSU	Road-side unit
FN_f	The f-th fog node
V_i	The i-th vehicle
λ	Security parameter
$H_i(1 \leq i \leq 7)$	Cryptographic hash functions
G, q, P	An additive elliptic curve group G with order q and generator P
s / P_{pub}	System master secret key / system public key
ID_i / PID_i	Real identity / pseudonym of V_i
ID_f / PID_f	Real identity / pseudonym of FN_f
(x_i^v, y_i^v)	The point selected by V_i
(x_f, y_f)	The point selected by FN_f
(x_i^t, y_i^t)	The point that TCC selects for vehicle V_i.
t_i / t_f / t_t	Timestamps generated by V_i / FN_f / TCC_t
M_i	Request message generated by vehicle V_i
MAC_i	Message authentication code of vehicle V_i
M_{agg}	Aggregated data of all vehicles' request messages
MAC_{agg}	Aggregated data of all vehicles' MAC
c, α, β	Hash values
r_i / R_i	Random number / random point generated by V_i
$\oplus$ / $\|\|$	XOR operation / concatenation of strings

3.1. System Model

Our BAKAF scheme is based on the new architecture with FNs, as illustrated in Figure 1. This architecture mainly comprises of four entities, i.e., traffic control center (TCC), fog node (FN), roadside unity (RSU), and vehicle. The functions and features of each entity are as follows.

TCC: TCC as a fully trusted entity, is controlled by government agencies. It possesses robust storage capacity and computing power. TCC is responsible for the registration of FNs, RSUs, and vehicles, broadcasting system parameters, and tracking and revoking malicious vehicles. By leveraging technologies such as artificial intelligence and big data, TCC can provide data support for services like road navigation, real-time information sharing, public security investigation, traffic improvement, and urban data analysis.

FN: FNs have strong storage and processing capabilities, primarily responsible for collecting and processing data within their domains. they are generally arranged at the edges of the fogs, so that the vehicles can handle various tasks nearby, thereby enhancing overall system efficiency and meeting the low-latency requirements of the VANETs. FNs are connected to TCC and RSUs via wired links for various interactions. they are semi-trusted entities that, on one hand, faithfully execute protocol commands, and on the other hand, monitor and collect data from other entities.

RSU: RSUs typically feature small signal coverage areas and limited storage and computing capabilities. In our scheme, we assume that RSUs are only used for collecting and forwarding the information they received.

Vehicle: Vehicles are equipped with wireless sensing devices and tamper-proof devices (TPD). They are untrusted entities with weak computing and storage capabilities, making them vulnerable to various attacks. Vehicles collect surrounding information through sensing devices and transmit it to nearby vehicles or RSUs via wireless communication technology.

 doi:10.20944/preprints202507.0134.v1

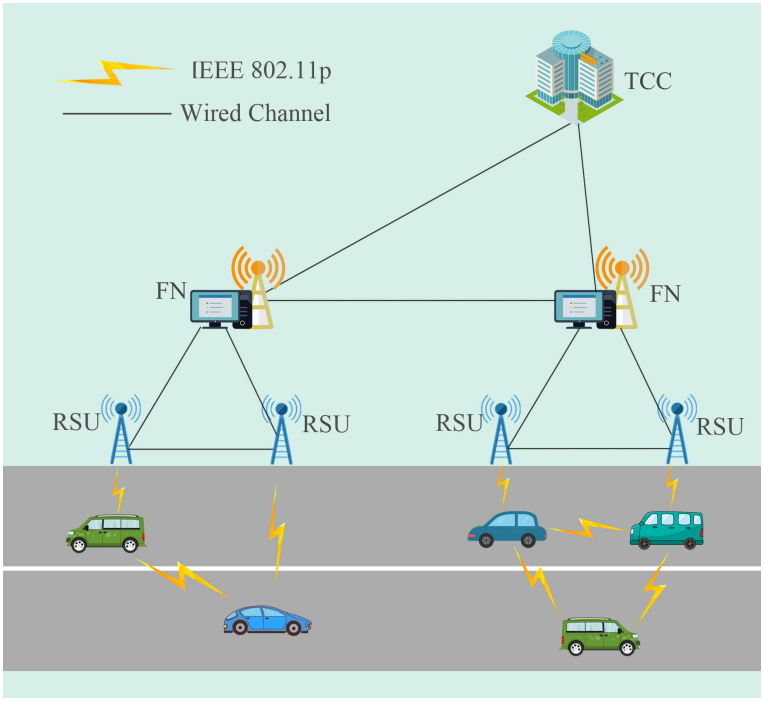

Figure 1. System model.

3.2. Pseudo Random Function (PRF)

Pseudo-random function (PRF) is an important cryptographic primitive that behaves similarly to a truly random function. Specifically, given an input consisting of a key $k \in \mathcal{K}$ and a binary string $x \in \mathcal{X}$, a PRF generates an output string $y \in \mathcal{Y}$ via the computation $y = \text{PRF}(k, x)$. A secure PRF must ensure that its output is computationally indistinguishable from the output of a truly random function for any probabilistic polynomial time (PPT) adversary. PRF typically have the following characteristics: (1) Given the same key and input, they will generate the same output. (2) For different keys and inputs, the output should appear random and unpredictable.

3.3. Elliptic Curve Cryptography (ECC)

Since ECC was built by Kobilitz, it has been widely applied to encryption and other safety-related areas [29].

Let F_p denote a finite field with a large prime number p as its order. We choose an elliptic curve E defined as $y^2 = x^3 + ax + b \pmod{p}$, where $a, b \in F_p$. Then, an additive cyclic elliptic curve group G on E with order q and a generator P can be generated, which contains the point at infinity O. The properties of group G are as follows:

Scalar multiplication: Let $\alpha \in Z_q^*$, the scalar multiplication on E is defined as $\alpha P = P + P + \cdots + P$ (α times).

Elliptic curve discrete logarithm (ECDL) assumption: For randomly chosen $P, Q \in G$ satisfying $Q = \alpha P$, where $\alpha \in Z_q^*$ is unknown, there exists no efficient algorithm that can determine α from Q in PPT with non-negligible advantage.

3.4. Security and Privacy Requirements

Mutual authentication: In order to ensure identity legitimacy of the communication entity during AKA communication, mutual authentication should be performed among vehicles, FNs, and the TCC.

Confidentiality: The session key established through the AKA process can be kept confidential from any other entity except for the participating entities.

Data integrity: If a message is maliciously forged during transmission, the receiver should be capable of detecting the forgery.

 doi:10.20944/preprints202507.0134.v1

Unlinkability: Unlinkability guarantees that no observable connection exists between different messages sent by the same vehicle. This security mechanism effectively prevents any entity from deducing whether two intercepted messages were transmitted by the identical sender.

Conditional privacy-preserving: During communication, the vehicle's real identity remains hidden from all entities except TCC, which can retrieve any vehicle's true identity when necessary.

Revocability: Once malicious vehicles are detected, TCC can prevent them from initiating further AKA sessions.

Forward and backward secrecy: From an adversarial perspective, session keys established across different sessions must be computationally indistinguishable. Knowledge of one session key provides no advantage in deriving another.

Resistance to various attacks: Our scheme must withstand various well-known attacks, such as impersonation attack, replay attack.

4. Prposed Scheme

In this section, we will describe our BAKAF scheme in detail, which consists of three phases, namely: setup phase, registration phase, and authentication and key agreement phase. In setup phase, TCC initializes the system and selects public parameters. In registration phase, all vehicles and fog nodes need to register with TCC, respectively. In authentication and key agreement phase, within the set time interval, a batch of vehicles simultaneously establishes corresponding session keys with the fog nodes.

4.1. Setup Phase

Given a security parameter λ, TCC defines F_p as a finite field with a large prime number p as its order. TCC then generates an elliptic curve E defined by the equation $y^2 \equiv x^3 + ax + b \pmod{p}$, where $a, b \in F_p$. Subsequently, TCC defines a additive cyclic elliptic curve group G with order q and generator P, which includes a point at infinity denoted as O. Finally, TCC selects a random number $s \in Z_q^*$ as the system's master secret key and retains the corresponding system public key $P_{pub} = sP$.

TCC constructs a revocation list, denoted as $RevList$, which is used to store the real identities of malicious vehicles. Then chooses 7 cryptographic hash functions: $H_1 : Z_q^* \times \{0,1\}^{l_i} \rightarrow Z_q^*$, $H_2 : G \rightarrow \{0,1\}^{l_i}$, $H_3 : Z_q^* \times \{0,1\}^{l_i} \times Z_q^* \times G \rightarrow \{0,1\}^{l_a}$, $H_4 : Z_q^* \times \{0,1\}^{l_i} \times Z_q^* \times Z_q^* \times \{0,1\}^* \rightarrow Z_q^*$, $H_5 : Z_q^* \times \{0,1\}^{l_i} \times Z_q^* \times Z_q^* \times Z_q^* \rightarrow Z_q^*$, $H_6 : \{0,1\}^* \rightarrow Z_q^*$, $H_7 : Z_q^* \times \{0,1\}^{l_i} \times \{0,1\}^{l_i} \times Z_q^* \times Z_q^* \times Z_q^* \rightarrow Z_q^*$, where l_i represents the length of the real identity or pseudonym, l_a represents the length of message authentication code.

TCC publishes the system parameters $parmas = \{p, q, a, b, P, P_{pub}, H_1, H_2, H_3, H_4, H_5, H_6, H_7\}$ and secretly saves the system's master secret key s.

4.2. Registration Phase

4.2.1. Registration for Vehicles

Assuming vehicle V_i requests registration, TCC selects a specific string ID_i as the vehicle V_i's real identity, calculates the certificate $c_i = H_1(s, ID_i)$, and sends (ID_i, c_i) to vehicle V_i through a secure channel.

4.2.2. Registration for Fog Node

Assuming fog node FN_f requests registration, TCC selects a specific string ID_f as the fog node FN_f's real identity, calculates the certificate $c_f = H_1(s, ID_f)$, and sends (ID_f, c_f) to fog node FN_f through a secure channel.

The certificate c_i (or c_f) is owned exclusively by the vehicle V_i (or fog node FN_f) itself and the TCC, and will play a vital role in both authentication and message integrity assurance.

 doi:10.20944/preprints202507.0134.v1

4.3. Authentication and Key Agreement Phase

Assume that there are n vehicles requesting key agreement with the fog node FN_f in a short period of time. For the convenience of explanation, we will take vehicle V_i as an example to elaborate. The main steps of this stage are shown in Figure 2.

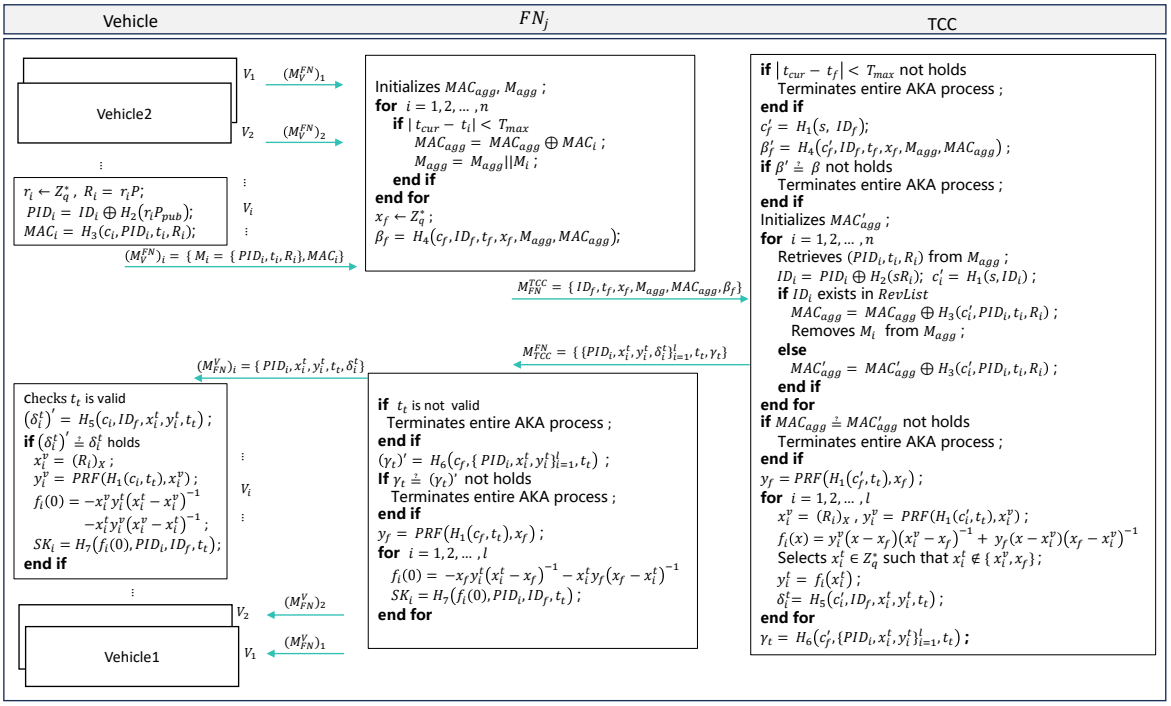

Figure 2. The main steps of authentication and key agreement phases.

4.3.1. Vehicle Requests to Generate a Session Key

Vehicle V_i selects a random number $r_i \in Z_q^*$ and computes $R_i = r_i P$ to obtain a random point on the group G.

Vehicle V_i retains its pseudonym $PID_i = ID_i \oplus H_2(r_i P_{pub})$. (The vehicle uses a different pseudonym each time it sends the requested information, helping to ensure conditional privacy-preserving.)

Generates the message authentication code $MAC_i = H_3(c_i, PID_i, t_i, R_i)$, where t_i is the timestamp, sets message $(M_V^{FN})_i = \{M_i = \{PID_i, t_i, R_i\}, MAC_i\}$.

Finally, vehicle V_i sends the message $(M_V^{FN})_i$ to fog node FN_f. It notes that the subscript V denotes that the sender of message is a vehicle, while the superscript FN denotes that the receiver of message is a FN, and the superscripts and subscripts of some symbols below have similar meanings.

4.3.2. FN Aggregates the Request Data from Each Vehicle

Upon receiving key agreement request information $\{M_i, MAC_i\}_{i=1}^n$ from n vehicles within the predetermined time interval, FN_f initializes MAC_{agg} (a binary string of length $|MAC_i|$, all bits set to 0) and M_{agg} (an empty string) for storing the aggregated MAC and message, respectively.

For each vehicle's information M_i, checks whether the inequality $|t_{cur} - t_i| < T_{max}$ holds? where t_{cur} and T_{max} denote the current timestamp and max valid time interval, respectively. If not, discard the information. Otherwise, proceed to calculate $MAC_{agg} = MAC_{agg} \oplus MAC_i$ and $M_{agg} = M_{agg} || M_i$. In this way, the information from all vehicles is aggregated into the variables M_{agg} and MAC_{agg}.

FN_f selects a random number $x_f \in Z_q^*$, such that $x_f \notin \{(R_1)_X, (R_2)_X, \cdots, (R_n)_X\}$, where $(R_i)_X$ represents the X-coordinate of the point R_i on the elliptic curve E. Calculates verification message $\beta_f = H_4(c_f, ID_f, t_f, x_f, M_{agg}, MAC_{agg})$, where t_f is the timestamp of FN_f. Then, FN_f sends the message $M_{FN}^{TCC} = \{ID_f, t_f, x_f, M_{agg}, MAC_{agg}, \beta_f\}$ to TCC.

 doi:10.20944/preprints202507.0134.v1

4.3.3. TCC Authenticates and Processes the FN's Requests

Upon receiving the batch key agreement request message M_{FN}^{TCC} from FN_f, TCC first checks whether the inequality $|t_{cur} - t_f| < T_{max}$ holds? If not, the entire AKA process will be terminated.

Next, TCC calculates $c'_f = H_1(s, ID_f)$ and $\beta'_f = H_4(c'_f, ID_f, t_f, x_f, M_{agg}, MAC_{agg})$, and checks whether the equation $\beta' \stackrel{?}{=} \beta$ holds. If it holds, two conclusions can be drawn: the identity legality of ID_f has been validated by TCC, and the integrity of M_{FN}^{TCC} can be guaranteed; Otherwise, the entire AKA process will be terminated.

Next, TCC initializes an auxiliary variable MAC'_{agg}, which is a binary string with the length equal to $|MAC_i|$ and each bit of which is 0, and MAC'_{agg} is used to re-record the information of legitimate vehicles. Then, TCC performs the following procedures for each vehicle V_i to verify its identity legitimacy and information integrity.

- Retrieves the i-th tuple (PID_i, t_i, R_i) from M_{agg} and obtains the real identity of V_i by computing $ID_i = PID_i \oplus H_2(sR_i)$.
- Calculates the vehicle V_i's certificate $c'_i = H_1(s, ID_i)$.
- Verifies whether vehicle V_i is revoked by checking if its real identity ID_i exists in the revocation list $RevList$.
  - If it is not revoked, it indicates that the vehicle is legal, then calculates $MAC'_{agg} = MAC'_{agg} \oplus H_3(c'_i, PID_i, t_i, R_i)$.
  - If it is revoked, calculates $MAC_{agg} = MAC_{agg} \oplus H_3(c'_i, PID_i, t_i, R_i)$, where the message authentication code MAC_i of the illegal vehicle V_i is removed from MAC_{agg}, at the same time, the message M_i of the illegal vehicle V_i should be removed from M_{agg}.

TCC checks whether the equation $MAC_{\text{agg}} \stackrel{?}{=} MAC'_{agg}$ holds. If it does not hold, it indicates that the messages from some legitimate vehicles have been tampered with, and the entire AKA process is terminated. Otherwise, it confirms that the information from all legitimate vehicles remains intact, and TCC proceeds with the following steps.

Computes $y_f = PRF(H_1(c'_f, t_t), x_f)$, where t_t is the timestamp of TCC. Then, based on the message $\{PID_i, t_i, R_i, ID_i\}$ of each legitimate vehicle V_i, TCC performs the following procedures.

- TCC sets $x_i^v = (R_i)_X$, and $y_i^v = PRF(H_1(c'_i, t_t), x_i^v)$.
- Using the Lagrange interpolation formula, TCC can obtain the equation $f_i(x) = y_i^v(x - x_f)(x_i^v - x_f)^{-1} + y_f(x - x_i^v)(x_f - x_i^v)^{-1}$ of a straight line passing through points (x_i^v, y_i^v) and (x_f, y_f). These straight lines corresponding to all legitimate vehicles pass through a common point (x_f, y_f), which can significantly reduce the computational and communication overhead.
- TCC selects a random number $x_i^t \in Z_q^*$ for vehicle ID_i, such that $x_i^t \notin \{x_i^v, x_f\}$, then substitutes x_i^t into the straight line equation $f_i(x)$ to compute the corresponding Y-coordinate, i.e., evaluates $y_i^t = f_i(x_i^t)$. Therefore, in addition to the points (x_i^v, y_i^v) and (x_f, y_f), TCC has now obtained the third point (x_i^t, y_i^t) on the straight line $f_i(x)$.
- TCC obtains verification message by computing $\delta_i^t = H_5(c'_i, ID_f, x_i^t, y_i^t, t_t)$.

Upon completing the above operations for each vehicle, TCC computes $\gamma_t = H_6(c'_f, \{PID_i, x_i^t, y_i^t\}_{i=1}^l, t_t)$, where l represents the number of legitimate vehicles, sets message $M_{TCC}^{FN} = \{\{PID_i, x_i^t, y_i^t, \delta_i^t\}_{i=1}^l, t_t, \gamma_t\}$, and transmits the M_{TCC}^{FN} to the fog node FN_f. Since the information of illegitimate vehicles has been deleted by TCC, the pseudonyms of legitimate vehicles need to be returned to the fog node FN_f.

4.3.4. FN Generates Session Keys

Upon receiving message M_{TCC}^{FN}, fog nodes FN_f checks whether t_t is valid. If not, the entire AKA process is terminated. If it is, the following process continues.

FN_f computes $(\gamma_t)' = H_6(c_f, \{PID_i, x_i^t, y_i^t\}_{i=1}^l, t_t)$ and checks whether the equation $\gamma_t \stackrel{?}{=} (\gamma_t)'$ holds. If holds, it indicates that the legality of the vehicles $\{V_i\}_{i=1}^l$ and the integrity of M_{TCC}^{FN} and M_V^{FN} are guaranteed. Then, FN_f performs the following steps.

 doi:10.20944/preprints202507.0134.v1

- The fog node FN_f sends the messages $\{M_{FN}^V\}_{i=1}^l = \{PID_i, x_i^t, y_i^t, t_t, \delta_i^t\}_{i=1}^l$ to the corresponding vehicles $\{V_i\}_{i=1}^l$ respectively.
- Computes $y_f = PRF(H_1(c_f, t_t), x_f)$, and performs the following operations for each legal vehicle.
  - Substitute $x = 0$ into the straight line equation $f_i(x)$ to obtain the Y-coordinate of the intersection point between the line $f_i(x)$ and the Y-axis: $f_i(0) = -x_f y_i^t (x_i^t - x_f)^{-1} - x_i^t y_f (x_f - x_i^t)^{-1}$. The line $f_i(x)$ passes through points (x_i^t, y_i^t) and (x_f, y_f), where point (x_f, y_f) is set by fog node FN_f and point (x_i^t, y_i^t) is set by TCC.
    Note: It's unnecessary to first solve for the explicit expression of the linear equation here.
  - Gets the session key between the vehicle V_i and the fog node FN_f by computing $SK_i = H_7(f_i(0), PID_i, ID_f, t_t)$.
  - Fog node FN_f stores the session key SK_i.

4.3.5. Vehicle Generates Session Key

After receiving message $(M_{FN}^V)_i$, vehicle ID_i checks whether t_t is valid. If not, the entire AKA process is terminated. If it is, the following processes continue.

Computes $(\delta_i^t)' = H_5(c_i, ID_f, x_i^t, y_i^t, t_t)$ and checks whether the equation $(\delta_i^t)' \stackrel{?}{=} \delta_i^t$ holds. If holds, it indicates that the legality of the vehicles V_i and the integrity of M_{TCC}^{FN} and M_V^{FN} are guaranteed. Then, FN_f performs the following steps.

- Sets $x_i^v = (R_i)_X$, and computes $y_i^v = PRF(H_1(c_i, t_t), x_i^v)$.
- Substitute $x = 0$ into the linear equation $f_i(x)$ to obtain the Y-coordinate of the intersection point between the line $f_i(x)$ and the Y-axis: $f_i(0) = -x_i^v y_i^t (x_i^t - x_i^v)^{-1} - x_i^t y_i^v (x_i^v - x_i^t)^{-1}$. The line $f_i(x)$ passes through points (x_i^t, y_i^t) and (x_i^v, y_i^v), where point (x_i^v, y_i^v) is set by vehicle V_i and point (x_i^t, y_i^t) is set by TCC.
  Note: For a specific vehicle V_i, the three points (x_i^t, y_i^t), (x_i^v, y_i^v), and (x_f, y_f) lie on the same straight line. Therefore, the line determined by any two of these points is identical, denoted as $f_i(x)$.
- Gets the session key between the vehicle V_i and the fog node FN_f by computing $SK_i = H_7(f_i(0), PID_i, ID_f, t_t)$.
- Vehicle ID_i stores the session key SK_i.

Remark 1: Within a fixed time interval, if n vehicles request AKA sessions with the same FN, the FN aggregates all AKA session requests, and the system then performs batch AKA processing. This approach offers at least three advantages: (1) reducing the overall system complexity; (2) lowering the communication overhead between the FN and TCC; and (3) decreasing the computational overhead for both the FN and TCC.

Remark 2: On TCC side, even if some vehicles are found to be illegal, the remaining legal vehicles can still perform the batch AKA process, which greatly enhances the flexibility of batch AKA.

5. Security Analysis

In this section, we will analyze the security performance of our scheme from both informal and formal aspects.

5.1. Informal Security Analysis

Mutual authentication: Our scheme comprises four phases of information transmission, all employing identical authentication mechanisms. We illustrate this using the TCC-to-FN message transmission as an example. TCC first computes $\gamma_t = H_6(c_f', \{PID_i, x_i^t, y_i^t\}_{i=1}^l, t_t)$, where $c_f' = H_1(s, ID_f)$ is the certificate of FN_f. Subsequently, TCC transmits the message $M_{TCC}^{FN} = \{\{PID_i, x_i^t, y_i^t, \delta_i^t\}_{i=1}^l, t_t, \gamma_t\}$ to FN_f. Upon receiving message M_{TCC}^{FN}, FN_f computes $(\gamma_t)' = H_6(c_f, \{PID_i, x_i^t, y_i^t\}_{i=1}^l, t_t)$ and checks whether the equation $\gamma_t \stackrel{?}{=} (\gamma_t)'$ holds. If the equation holds, it confirms that the message must

 doi:10.20944/preprints202507.0134.v1

originate from TCC, since only FN_f and TCC possess the certificate c_f. Therefore, the authenticity is guaranteed.

Confidentiality: In our scheme, the method for calculating the session key is $SK_i = H_7(f_i(0), PID_i, ID_f, t_t)$. To obtain the session key between vehicle V_i and FN_j, we first need to obtain $f_i(0)$, which represents the Y-coordinate of the line $f_i(x)$ at $x = 0$. At least two points are required to determine a straight line. Even if a malicious entity intercepts a point on the straight line transmitted from TCC, since it cannot obtain the certificates of the vehicle V_i or the FN FN_j, it cannot obtain the other point on the required straight line. Consequently, it cannot construct the linear equation and calculate $f_i(0)$. Therefore, the session key between vehicle V_i and FN_j remains confidential to any malicious entity.

Data integrity: Our scheme comprises four phases of information transmission, all employing identical message integrity protection mechanisms. We illustrate this using the FN-to-TCC message transmission as an example. FN_f first sends the message $M_{FN}^{TCC} = \{ID_f, t_f, x_f, M_{agg}, MAC_{agg}, \beta_f\}$ to TCC, where $\beta_f = H_4(c_f, ID_f, t_f, x_f, M_{agg}, MAC_{agg})$, and $c_f = H_1(s, ID_f)$ is the certificate of FN_f, which is owned exclusively by the fog node FN_f itself and the TCC. If any field in M_{FN}^{TCC} is tampered with (e.g., t_f altered to t_f^*), the malicious attacker must recompute the verification value β_f. Without FN_f's certificate, the attacker can only arbitrarily choose c_f^*, yielding: $\beta_f = H_4(c_f^*, ID_f, t_f^*, x_f, M_{agg}, MAC_{agg})$.

Upon receiving the message M_{FN}^{TCC} from FN_f, TCC first recalculates certificate $c_f' = H_1(s, ID_f)$ and verification value $\beta_f' = H_4(c_f', ID_f, t_f^*, x_f, M_{agg}, MAC_{agg})$. Then, checks whether the equation $\beta_f' \stackrel{?}{=} \beta_f$ holds. Because c_f' is not equal to c_f^*, the verification equation will not hold. In this way, the integrity of M_{FN}^{TCC} can be guaranteed.

Unlinkability: For each AKA session request, the vehicle generates a fresh pseudonym, timestamp, and random ECC point, where neither the timestamp nor the ECC point correlates with the vehicle's identity or pseudonym, and no other entity except the TCC and the vehicle itself can track the true identity through pseudonym. Therefore, these mechanism effectively prevents any malicious entity from deducing whether two intercepted messages were transmitted by the identical sender.

Conditional privacy-preserving: Prior to initiating an AKA session, vehicle V_i generates a pseudonym $PID_i = ID_i \oplus H_2(r_i P_{pub})$, then uses PID_i for all subsequent communications. Once message is disputed, The TCC can recover the real identity via $ID_i = PID_i \oplus H_2(sR_i)$, where s is the master secret key, R_i is a random ECC point.

To derive the real identity from PID_i, any entity must calculate either $ID_i = PID_i \oplus H_2(sR_i)$ or $ID_i = PID_i \oplus H_2(r_i P_{pub})$. s is the system's master private key, owned only by TCC, while r_i is a random number, owned only by the vehicle V_i. Solving for r_i from R or s from P_{pub} requires solving the difficult ECDL problem. Therefore, except TCC, no other entities can obtain the real identity.

Revocability: TCC maintains a revocation list $RevList$. Upon detection of a malicious vehicle V_i, its real identity ID_i is added to $RevList$. If V_i attempts to initiate subsequent AKA sessions, TCC will detect that its real identity ID_i is already in the revocation list $RevList$ and terminate the AKA procedure. All revocation operations are concentrated on TCC, which has abundant computing and storage resources, making it relatively friendly for vehicle terminals with limited resources.

Forward and backward secrecy: In our scheme, the session key is computed as $SK_i = H_7(f_i(0), PID_i, ID_f, t_t)$, where: $f_i(0)$ depends on two fresh random nonces, PID_i is a randomly generated ECC point. The single-use and randomness of these parameters ensure no correlation between session keys sk_i across different sessions, thus guaranteeing both forward and backward secrecy.

Resistance to impersonation Attack: In our scheme, any passed message contains a verification value, which is bound to the certificate of the entity (vehicle or FN) through a hash function. Since these certificates are exclusively held by TCC and the originating entity itself, no other party can forge valid verification values. Consequently, any impersonation attempt by malicious attackers will be detected during the recipient's verification process.

 doi:10.20944/preprints202507.0134.v1

Resistance to replay attack: In our scheme, each transmitted message incorporates a timestamp. For instance, during the FN-to-TCC message transmission, FN_f sends the message $M_{FN}^{TCC} = \{ID_f, t_f, x_f, M_{agg}, MAC_{agg}, \beta_f\}$ to TCC, where t_f denotes the timestamp. As demonstrated in the "**Data integrity**" section, the timestamp t_f is immutable. Upon receiving the message M_{FN}^{TCC}, TCC verifies its freshness by checking whether the inequality $|t_c - t_f| < \Delta T$ holds, where t_c represents the current time. If not, TCC treats the message as expired. Therefore, this timestamp validation mechanism, combined with the immutability guarantee, ensures that our scheme can effectively defend against replay attack.

5.2. Formal Security Proof

In this section, we formally prove that our scheme satisfies session key security under the Real-Or-Random (ROR) model [30].

5.2.1. Security Model

We first establish a security model to define the adversaries' capabilities and the interaction rules between the challenger and the adversaries, and the model includes the following definitions:

Participants: Let Π_V^i, Π_{FN}^f, and Π_{TCC}^t denote the i-th instance of a vehicle, the f-th instance of a FN, the t-th instance of a TCC, respectively. The concrete instance of these participants can also be represented as Π_Λ^χ, where Λ indicates the instance type and χ represents the index. Notably, RSUs are excluded from being considered as participating entities, as they merely function as conventional base stations for message forwarding.

Partnering: Two participants are considered as partners if they (a) belong to the same session, (b) successfully exchange messages in sequence, and (c) want to mutually authenticate each other.

Freshness: If the adversary $\mathcal{A}$ does not obtain the session key shared among Π_V^i, Π_{FN}^f, and Π_{TCC}^t, then these instances are considered fresh.

Adversary: The adversary $\mathcal{A}$ can participate in interactions of Π_V^i, Π_{FN}^f, and Π_{TCC}^t by adopting the following oracle queries.

- $Execute(\Pi_V^i, \Pi_{FN}^f, \Pi_{TCC}^t)$: This query simulates the passive adversary $\mathcal{A}$ to intercept messages exchanged among Π_V^i, Π_{FN}^f, and Π_{TCC}^t.
- $Send(\Pi_\Lambda^\chi, m)$: The query models an active adversary $\mathcal{A}$ sending message m to Π_V^i, Π_{FN}^f, and Π_{TCC}^t. Upon receiving this query, these instances return corresponding response messages to $\mathcal{A}$.
- $Test(\Pi_\Lambda^\chi)$: When challenger $\mathcal{C}$ receives this query from adversary $\mathcal{A}$, it randomly selects a bit $b \in \{0,1\}$, if $b = 1$, $\mathcal{C}$ sends the real session key of Π_Λ^χ to $\mathcal{A}$; if $b = 0$, $\mathcal{C}$ sends a random key of the same length as the session key to the $\mathcal{A}$. If the session key of Π_Λ^χ is undefined, or if a Test query has been made to Π_Λ^χ or its partners, $\mathcal{A}$ receives $\perp$ as an invalid value.

Semantic security: Adversary $\mathcal{A}$ first perform $Test(\Pi_\Lambda^\chi)$ query, and guess the random value of b in $Test(\Pi_\Lambda^\chi)$ query, denotes as b', if $b' = b$, $\mathcal{A}$ wins, We define our BAKAF scheme as $\mathcal{P}$, and the advantage of $\mathcal{A}$ to break the $\mathcal{P}$'s semantic security based on ROR model within PPT as $Adv_\mathcal{A}^\mathcal{P} = |2Pr[b' = b] - 1|$, where $Pr[E]$ denotes the probability that the event E occurs, if $Adv_\mathcal{A}^\mathcal{P}$ is negligible, our scheme is regarded as secure under the ROR model.

5.2.2. Security Proof

In this subsection, we prove that our BAKAF scheme satisfies the semantic security of the session key.

Theorem 1. *Let $N_i (i = 1, 2, \ldots, 7)$, $|hash_i| (i = 1, 2, \ldots, 7)$, N_s, N_e, N_t, and $Adv_\mathcal{A}^{PRF}$ represent the maximum number of hash $H_i (i = 1, 2, \ldots, 7)$ queries, the space range of hash function $H_i (i = 1, 2, \ldots, 7)$, the number of* Send *oracle queries, the number of* Execute *oracle queries, the number of* Test *oracle queries, and the advantage*

 doi:10.20944/preprints202507.0134.v1

of adversary $\mathcal{A}$ in breaking the PRF, respectively. Thus, the advantage $Adv_{\mathcal{A}}^{\mathcal{P}}$ of adversary $\mathcal{A}$ in breaking the semantic security of the session keys in our BAKAF scheme within PPT, can be calculated as follows:

$$Adv_{\mathcal{A}}^{\mathcal{P}} \leq \frac{N_s + N_e}{q} + \sum_{i=1}^{7} \frac{N_i^2}{|hash_i|} + 2Adv_{\mathcal{A}}^{PRF} \tag{1}$$

Proof: Similar to [31,32], to prove **Theorem 1**, we construct a sequence of games Game_i ($i = 0, 1, 2, 3$). These games involve interactions between adversary $\mathcal{A}$ and challenger $\mathcal{C}$. We define succ_i as the event that adversary $\mathcal{A}$ wins in Game_i.

Game$_0$: This game serves as the starting point of the entire proof process. It simulates a realistic environment in which adversary $\mathcal{A}$ launches actual attacks. A random number b is predetermined before the game begins. Based on the semantic security of the session key, we have

$$Adv_{\mathcal{A}}^{\mathcal{P}} = 2|Pr[succ_0] - 1/2| \tag{2}$$

Game$_1$: In this game, the adversary $\mathcal{A}$ is permitted to eavesdrop on communications among parties Π_V^i, Π_{FN}^f and Π_{TCC}^t by executing the Execute operation. Subsequently, $\mathcal{A}$ performs the Test operation to distinguish the real session key SK_i from a random value. And because of $SK_i = H_7(f_i(0), PID_i, ID_f, t_t)$, where $f_i(0) = -x_i^v y_i^t (x_i^t - x_i^v)^{-1} - x_i^t y_i^v (x_i^v - x_i^t)^{-1}$. Since determining the equation of a straight line requires at least two distinct points on the line, but $\mathcal{A}$ can obtain at most one point, the actual session key cannot be computed. Consequently, the probability of $\mathcal{A}$ winning Game_1 is identical to that of winning Game_0, as shown below:

$$Pr[succ_1] = Pr[succ_0] \tag{3}$$

Game$_2$: Based on *Game*$_1$, the adversary $\mathcal{A}$ can perform Send and hash oracle queries. In this game, $\mathcal{A}$ launches spoofing attacks by forging and sending malicious requests to Π_V^i, Π_{FN}^f, and Π_{TCC}^t. Our scheme allows $\mathcal{A}$ to modify messages M_V^{FN}, M_{FN}^{TCC}, M_{TCC}^{FN}, and M_{FN}^V, but these messages contain random elements (r_i, x_f, x_i^t) and independent timestamps (t_i, t_f, t_t). Hence, $\mathcal{A}$ must execute Send queries without causing collisions. By the birthday paradox, we derive the following probability bound:

$$|Pr[succ_2] - Pr[succ_1]| \leq \frac{N_s + N_e}{2q} + \sum_{i=1}^{7} \frac{N_i^2}{2|hash_i|} \tag{4}$$

Game$_3$: In this game, the $\mathsf{Send}(\Pi_{\mathsf{TCC}}^t, M_{FN}^{TCC})$ oracle is slightly modified compared to *Game*$_2$. Specifically, when processing queries from $\mathcal{B}$, the challenger $\mathcal{C}$ now substitutes the PRF outputs with truly random numbers. Since this modification affects exactly two PRF operations, namely $PRF(H_1(c_f', t_t), x_f)$ and $PRF(H_1(c_i', t_t), x_i^v))$, we can derive the following probability bound:

$$|Pr[succ_3] - Pr[succ_2]| \leq Adv_{\mathcal{A}}^{PRF} \tag{5}$$

After that, the adversary $\mathcal{A}$ has utilized all available oracles to challenge the semantic security of protocol $\mathcal{P}$. Adversary $\mathcal{A}$ attempts to win the game solely by guessing the value of b. All session keys SK_i used to respond to Test queries in this game are independently and uniformly distributed. As a result, no information regarding the hidden bit b used by the Test oracle is leaked to the adversary. Consequently, we obtain:

$$Pr[succ_3] = 1/2 \tag{6}$$

By combining the above formulas: (2), (3), (4), (5), (6), we can obtain:

 doi:10.20944/preprints202507.0134.v1

$$\begin{aligned} Adv_{\mathcal{A}}^{\mathcal{P}} &= 2|Pr[succ_0] - 1/2| \\ &= 2|Pr[succ_1] - Pr[succ_3]| \\ &= 2|Pr[succ_1] - Pr[succ_2] + Pr[succ_2] - Pr[succ_3]| \\ &\leq 2(|Pr[succ_2] - Pr[succ_1]| \\ &+ |Pr[succ_3] - Pr[succ_2]|) \\ &\leq \frac{N_s + N_e}{q} + \sum_{i=1}^{7} \frac{N_i^2}{|hash_i|} + 2Adv_{\mathcal{A}}^{PRF} \end{aligned} \tag{7}$$

Since q and $|Hash_i|$ are typically sufficiently large, and $Adv_{\mathcal{B}}^{PRF}$ is small enough and infinitely close to 0, the adversary $\mathcal{A}$'s advantage $Adv_{\mathcal{A}}^{\mathcal{P}}$ in breaking the semantic security of our BAKAF scheme is negligible. Thus, our scheme is semantically secure under the ROR model.

6. Peformance Evaluation

In this subsection, we evaluate the proposed scheme's performance through two key metrics: computational overhead and communication overhead. We compare our BAKAF scheme with state-of-the-art provably secure AKA schemes [20–23]. The selection of these comparative schemes is based on two key criteria: (1) Our scheme employs three-party entity collaboration for key agreement, and all selected comparison schemes similarly adopt a three-party framework. (2) Since our scheme eliminates computationally expensive bilinear pairing operations, pairing-based AKA schemes are excluded from the comparison to ensure fairness.

As far as we know, this scheme is the first batch three-party AKA scheme in VANETs. Therefore, if other schemes perform n AKA sessions, the computational and communication overheads of those schemes are n times the respective overheads of a single AKA session.

6.1. Computational Overhead Comparison

Before comparing the computational overhead, we use the MIRACL [33] library to measure all basic operations. All experiments are conducted on a laptop equipped with an Intel Core i5-12500 processor, 16 GB of memory, and the Windows 11 operating system. We omitted normal Z_q^* operations (addition, multiplication) and string operations (concatenation, XOR) due to their negligible execution times. The experimental results for the average time consumption of basic operations are shown in Table 2. It is noted that each operation's execution time was averaged across 1000 repetitions.

Table 2. The execution time of basic operations.

Operation	Description	Time (ms)
T_{sm}	Scale multiplication based on ECC	0.562
T_{la}	Lagrange interpolation	0.011
T_h	One-way hash	0.005
T_{aes}	AES-256 encryption/decryption	0.016
T_{prf}	Pseudo random function	0.015
T_{cm}	Extended Chebyshev chaotic map	0.381

In a complete AKA session process of the Lu et al.'s scheme [20], a vehicle needs to execute 2 scalar multiplication operations based on ECC, 1 Lagrange interpolation operation, 1 PRF operation and 5 one-way hash operations, so the computational overhead of a vehicle is $2T_{sm} + T_{la} + T_{prf} + 5T_h$; A fog node needs to execute 1 PRF operation, 1 Lagrange interpolation operation and 4 hash operations, so the computational overhead of the fog node is $T_{la} + T_{prf} + 4T_h$; A sub-TA needs to execute 1 Lagrange interpolation operation, 1 scalar multiplication operation based on ECC, 2 PRF operations and 9 one-way hash operations, so the computational overhead of a sub-TA is $T_{sm} + T_{la} + 2T_{prf} + 9T_h$, in scheme [20], a lookup operation for real ID was performed on the Cuckoo Filter, however, since this

 doi:10.20944/preprints202507.0134.v1

functionality is not implemented in our proposed scheme, we omitted this operation to ensure fairness, the role of sub-TA is similar to that of TCC in our scheme. Thus, the total computational overhead of scheme [20] is $n(3T_{sm} + 3T_{la} + 4T_{prf} + 18T_h)$.

In a complete AKA session process of the Qiao et al.'s scheme [21], a user needs to execute 2 Extended Chebyshev chaotic map operations, 7 one-way hash operations, so the computational overhead of a user is $2T_{cm} + 7T_h$, and the role of user is similar to that of vehicle in our scheme; A fog node needs to execute execute 3 Extended Chebyshev chaotic map operations, 4 one-way hash operations, so the computational overhead of the fog node is $3T_{cm} + 4T_h$; A server needs to execute 3 Extended Chebyshev chaotic map operations, 4 symmetric encryption operations, and 13 one-way hash operations, so the computational overhead of a cloud needs is $3T_{cm} + 4T_{aes} + 13T_h$, and the role of server is similar to that of TCC in our scheme. Thus, the total computational overhead of scheme [21] is $n(8T_{cm} + 24T_h + 4T_{aes})$.

In a complete AKA session process of the Jiang et al.'s scheme [22], a user needs to execute 5 scalar multiplication operations based on ECC, 7 one-way hash operations, so the computational overhead of a user is $5T_{sm} + 7T_h$, and the role of user is similar to that of vehicle in our scheme; A autonomous vehicle needs to execute 4 one-way hash operations, and the role of autonomous vehicle is similar to that of fog node in our scheme; A cloud needs to execute 5 scalar multiplication operations based on ECC and 13 one-way hash operations, so the computational overhead of a cloud needs is $5T_{sm} + 13T_h$, and the role of cloud is similar to that of TCC in our scheme. Thus, the total computational overhead of scheme [22] is $n(10T_{sm} + 24T_h)$.

In a complete AKA session process of the Cui et al.'s scheme [23], a vehicle needs to execute 8 one-way hash operations and 3 scalar multiplication operations based on ECC, so the computational overhead of a vehicle is $8T_h + 3T_{sm}$; A cloud service needs to execute 7 one-way hash operations and 3 scalar multiplication operations based on ECC, so the computational overhead of the cloud service is $7T_h + 3T_{sm}$, and the role of cloud service is similar to that of fog node in our scheme; A TA needs to execute 10 one-way hash operations and 2 scalar multiplication operations based on ECC, so the computational overhead of a TA is $10T_h + 2T_{sm}$, The role of TA is similar to that of TCC in our scheme. Thus, the total computational overhead of scheme [23] is $n(25T_h + 8T_{sm})$.

In our BAKAF scheme, assume that AKA is executed in a batch for n vehicles, a vehicle needs to execute 2 scalar multiplication operations based on ECC, 1 Lagrange interpolation operation, 1 PRF operation and 5 one-way hash operations, so the computational overhead of a vehicle is $2T_{sm} + T_{la} + T_{prf} + 5T_h$, and the computational overhead of n vehicle is $2nT_{sm} + nT_{la} + nT_{prf} + 5nT_h$; A fog node needs to execute 1 PRF operation, n Lagrange interpolation operation and $3 + n$ hash operations, so the computational overhead of the fog node is $nT_{la} + T_{prf} + (3 + n)T_h$ for n vehicles and $T_{la} + T_{prf} + 4T_h$ for $n = 1$ vehicle; TCC needs to execute n Lagrange interpolation operations, n scalar multiplication operations based on ECC, $n + 1$ PRF operations and $5n + 4$ one-way hash operations, so the computational overhead of TCC is $nT_{sm} + nT_{la} + (n + 1)T_{prf} + (5n + 4)T_h$ for n vehicles and $T_{sm} + T_{la} + 2T_{prf} + 9T_h$ for $n = 1$ vehicle. Thus, the total computational overhead is $3nT_{sm} + 3nT_{la} + 2(n + 1)T_{prf} + (11n + 7)T_h$ for n vehicles and $3T_{sm} + 3T_{la} + 4T_{prf} + 18T_h$ for $n = 1$ vehicle.

Consequently, the comparison results of computational overhead between our scheme and schemes [20–23] are presented in Table 3.

 doi:10.20944/preprints202507.0134.v1

Table 3. Computational overhead for n AKA session (ms).

Scheme	Vehicle	FN	TCC	Total
[20]	$n(2T_{sm}+T_{la}+T_{prf}+5T_h)$ =1.185n	$n(T_{la}+T_{prf}+4T_h)$ =0.056n	$n(T_{sm}+T_{la}+2T_{prf}+9T_h)$ =0.668n	$n(3T_{sm}+3T_{la}+4T_{prf}+18T_h)$ =1.909n
[21]	$n(2T_{cm}+7T_h)$ =0.797n	$n(3T_{cm}+4T_h)$ =1.163n	$n(3T_{cm}+4T_{aes}+13T_h)$ =1.272n	$n(8T_{cm}+24T_h+4T_{aes})$ =3.232n
[22]	$n(5T_{sm}+7T_h)$ =2.845n	$n(4T_h)$ =0.02n	$n(5T_{sm}+13T_h)$ =2.875n	$n(10T_{sm}+24T_h)$ =5.74n
[23]	$n(8T_h+3T_{sm})$ =1.726n	$n(7T_h+3T_{sm})$ =1.721n	$n(10T_h+2T_{sm})$ =1.174n	$n(25T_h+8T_{sm})$ =4.621n
Our	$2nT_{sm}+nT_{la}+nT_{prf}+5nT_h$ =1.185n	$nT_{la}+T_{prf}+(3+n)T_h$ =0.016n+0.04	$nT_{sm}+nT_{la}+$ $(n+1)T_{prf}+(5n+4)T_h$ =0.623n +0.045	$3nT_{sm}+3nT_{la}+$ $2(n+1)T_{prf}+(11n+7)T_h$ =1.824n+0.085

In schemes [20–23], executing n AKA sessions incurs n the overhead of a single session. Since the entities in schemes [20–23] have similar functions to their counterparts in our scheme, this table only uses the entity names from our scheme.

And we use Figure 3 to show the comparison results for a single AKA session, according to Figure 3, our scheme maintains identical computational overhead to Scheme [20], while demonstrating significant advantages over Schemes [21–23] in terms of both per-entity and total computational overhead. Meanwhile, Figure 4 compares the computational overhead (both per-entity and total) for all schemes when n AKA sessions are conducted within a fixed time interval. The results demonstrate that as the number of vehicles n increases, our scheme achieves lower computational overhead than baseline schemes.

The low computational overhead of our scheme primarily stems from two key factors: (1) Our scheme employs lightweight cryptographic primitives, while the more computationally intensive ECC-based scalar multiplication operations are exclusively used for anonymous credential generation; (2) The batch AKA approach significantly reduces the overall computational operations. Beyond reducing per-entity computational overhead, our scheme substantially decreases the interaction between FN and TCC through batch AKA session processing.

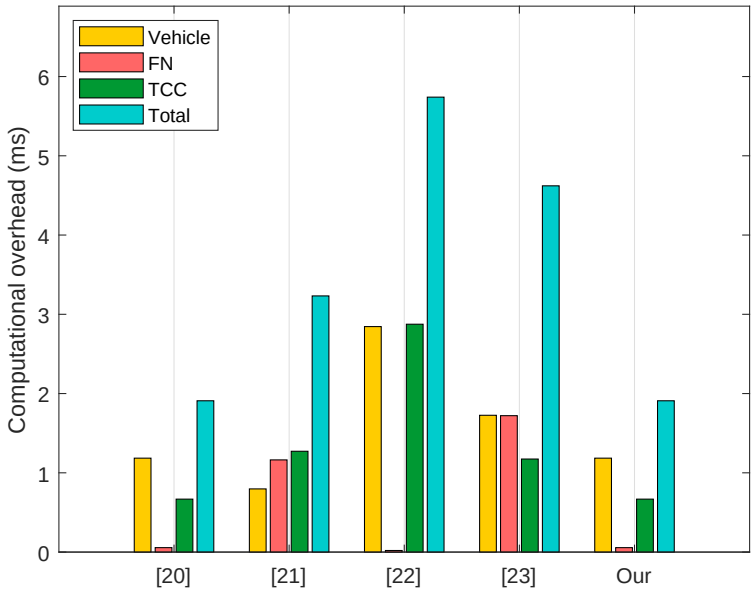

Figure 3. Computational overhead comparison for a **single** AKA session.

 doi:10.20944/preprints202507.0134.v1

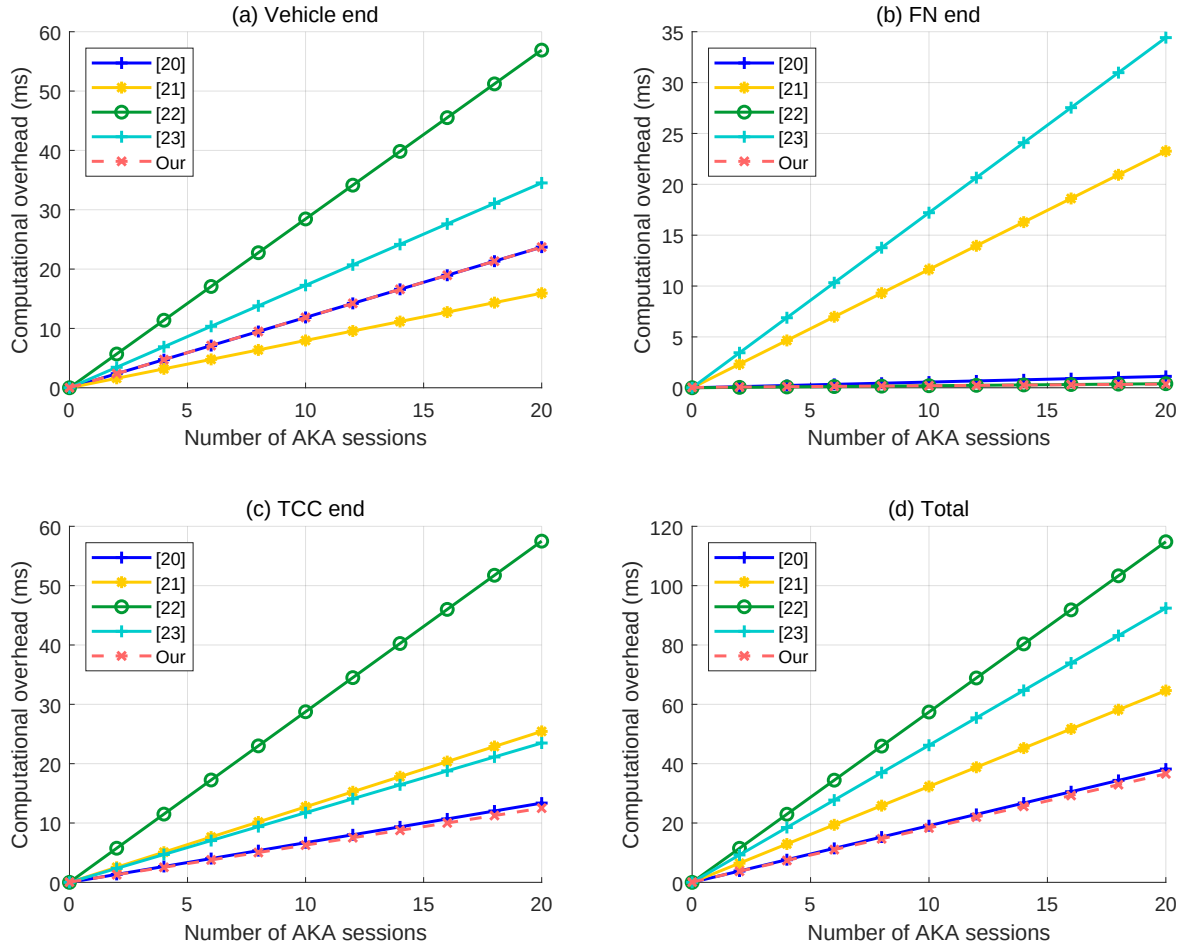

Figure 4. A comparison of the computational overhead for n AKA sessions.

6.2. Communication Overhead Comparison

Before comparing the communication overhead, we assume that the security parameters of all schemes are 128 bits in length. The element types and lengths involved in information transmission for each scheme are listed in Table 4. To ensure fairness and simplicity, we adopt AES as the symmetric encryption algorithm across all schemes, with ciphertext length identical to plaintext length.

Table 4. The sizes of each element (byte).

Symbol	Description	Size (byte)
$\|G\|$	The size of element in elliptic curve addition group	64
$\|t\|$	The size of timestamp	4
$\|ID\|$	The size of real identity or pseudonym	20
$\|M\|$	the size of message authentication code	20
$\|Z_q^*\|$	The size of element in Z_q^*	32
$\|E\|$	The size of the output generated by extended Chebyshev polynomial	32

In Scheme [20], a vehicle transmits $\{pid_i, t_i, R_i, \alpha_i\}$ to the FN in the first stage, where pid_i is the pseudonym, whose length is identical to $|ID|$, t_i is the timestamp, R_i is an element in elliptic curve group G, $\alpha_i \in Z_q^*$, so the communication overhead in the first stage is $|ID| + |t| + |G| + |Z_q^*| =$ 120 bytes. In the second stage, the FN transmits $\{id_j, t_j, n_j, pid_i, t_i, R_i, \alpha_i, \beta_j\}$ to TCC, where id_j is the real identity, t_j is the timestamp, $n_j, \beta_j \in Z_q^*$, so the communication overhead in this stage is $2|ID| + 2|t| + |G| + 3|Z_q^*| = 208$ bytes. In the third stage, TCC transmits $\{n_k, fn_k, t_k, \gamma_k, \delta_k\}$ to the FN, where $n_k, fn_k, \gamma_k, \delta_k \in Z_q^*$, and t_k is the timestamp, so the communication overhead in this stage is

 doi:10.20944/preprints202507.0134.v1

$|t| + 4|Z_q^*| = 132$ bytes. In the fourth stage, the FN transmits $\{n_k, fn_k, t_k, \delta_k\}$ to the vehicle, so the communication overhead in this stage is $|t| + 3|Z_q^*| = 100$ bytes. Suppose n vehicles conducting AKA sessions during a fixed time interval, the total communication overhead is $(120 + 208 + 132 + 100)n = 560n$ bytes.

In Scheme [21], a user transmits $MS_1 = \{NID_i, PID_i, W_i, T_1, T_u(x)\}$ to the fog node FN_j in the first stage, where $NID_i = E_s(ID_i||O_i)$ is obtained by symmetrically encrypting the concatenation of the identifier ID_i and a random number O_i, whose length is identical to $|ID| + |Z_q^*|$, T_1 is timestamp, PID_i and W_i are derived from one-way hash functions, since their data types are unspecified in the original paper, we adopt the 20-byte length defined therein, $T_u(x)$ is the extended Chebyshev polynomial, whose length is $|E|$. So the communication overhead in the first stage is $|MS_1| = |ID| + |Z_q^*| + 2 * 20 + |t| + |E| = 128$ bytes. In the second stage, the FN_j transmits $MS_2 = \{MS_1, NID_j, A, T_v(x), W_j, T_2\}$ to cloud server S, where $NID_j = E_s(ID_j||N_j)$ is obtained by symmetrically encrypting the concatenation of the identifier ID_j and a random number N_j, whose length is $|ID| + |Z_q^*|$, A and $T_v(x)$ are the extended Chebyshev polynomial, T_2 is timestamp, W_j is derived from one-way hash functions, so the communication overhead in this stage is $|MS_1| + |ID| + |Z_q^*| + |t| + 2|E| + 20 = 268$ bytes. In the third stage, cloud server S transmits $|MS_3| = \{V_i, V_j, Au_i, Au_j, B, C, T_3\}$ to the fog node FN_j, where V_i, V_j, Au_i, Au_j are derived from one-way hash functions, B and C are the extended Chebyshev polynomial, and T_3 is the timestamp, so the communication overhead in this stage is $4 * 20 + 2|E| + |t| = 148$ bytes. In the fourth stage, the fog node FN_j transmits $MS_4 = \{V_i, Au_i, A, B, C, T_3\}$ to the user, where T_3 is timestamp, so the communication overhead in this stage is $2 * 20 + 3|E| + |t| = 140$ bytes. Suppose n users conducting AKA sessions during a fixed time interval, the total communication overhead is $(128 + 268 + 148 + 140)n = 684n$ bytes.

Similarly, when executing n AKA sessions within a fixed time interval, the total communication overhead for Scheme [22] and Scheme [23] is calculated as $704n$ bytes and $656n$ bytes, respectively.

In our scheme, suppose there are n vehicles performing AKA sessions with the same fog node FN_f within a fixed time interval, vehicle V_i need to transmits $(M_V^{FN})_i = \{M_i = \{PID_i, t_i, R_i\}, MAC_i\}$ to the fog node FN_f, where PID_i is pseudonym, t_i is timestamp, R_i is an element in elliptic curve group G, and MAC_i is the message authentication code, so the communication overhead of n vehicles in the first stage is $(|ID| + |t| + |G| + |M|)n = 108n$ bytes. In the second stage, the fog node FN_f transmits the message $M_{FN}^{TCC} = \{ID_f, t_f, x_f, M_{agg}, MAC_{agg}, \beta_f\}$ to TCC, where ID_f is real identity, t_f is time timestamp, $x_f, \beta_f \in Z_q^*$, M_{agg} is formed by sequentially concatenating n messages $\{M_i\}_{i=1}^n$, whose length is $n(|ID| + |t| + |G|)$, and MAC_{agg} is message authentication code, so the communication overhead in this stage is $|ID| + |t| + 2|Z_q^*| + n(|ID| + |t| + |G|) + |M| = 108 + 88n$ bytes. In the third stage, TCC transmits the message $M_{TCC}^{FN} = \{\{PID_i, x_i^t, y_i^t, \delta_i^t\}_{i=1}^l, t_t, \gamma_t\}$ to the fog node FN_f, where t_t is timestamp, $x_i^t, y_i^t, \delta_i^t, \gamma_t \in Z_q^*$, here we assume $l = n$, implying that all vehicles' data are both correct and complete. so the communication overhead in this stage is $n(|ID| + 3|Z_q^*|) + |t| + |Z_q^*| = 36 + 116n$ bytes. In the fourth stage, the fog node FN_f sends the messages $\{M_{FN}^V\}_{i=1}^n = \{PID_i, x_i^t, y_i^t, t_t, \delta_i^t\}_{i=1}^n$ to the corresponding vehicles $\{V_i\}_{i=1}^n$ respectively,so the communication overhead in this stage is $n(|ID| + 3|Z_q^*| + |t|) = 120n$ bytes. Thus, for n vehicles executing the AKA sessions, the total communication overhead across all four stages amounts to $432n + 144$ bytes.

Figure 5 compares the communication overhead of different schemes as the number of AKAs varies. As shown in the figure, our scheme demonstrates a clear advantage as the number of AKAs increases. This is primarily because our scheme processes multiple AKA requests in batches within a fixed time interval.

 doi:10.20944/preprints202507.0134.v1

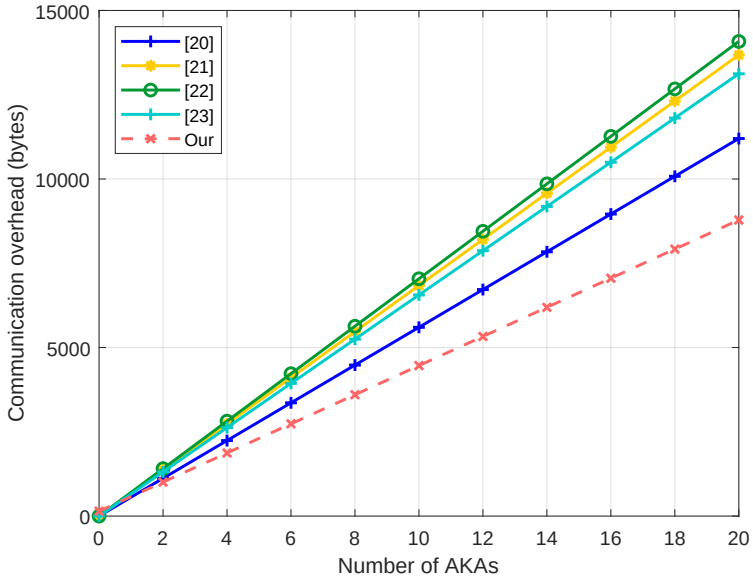

Figure 5. The comparision of communication overhead.

7. Conclusion and Remark

In this scheme, for a batch of vehicles initiating AKA requests to the same fog node within a fixed short time interval, We combine fog computing and Lagrange interpolation to complete these AKA processes in a round of operations. The interpolation points of the straight lines corresponding to each vehicle share a common point at the fog node. These mechanisms ensure that our scheme possesses very low computational and communication overhead while simplifying the overall system operations.

The authentication component of our scheme relies on the certificates of vehicles and fog nodes, as well as the system master secret key. It is assumed that the system master secret key and the certificate values remain perpetually secure. However, existing physical attack techniques (such as side-channel attacks) can obtain such sensitive information. Since existing technologies like physically unclonable functions and fuzzy extractor can effectively resist these physical attacks, our scheme does not focus on this aspect in-depth.

The batch AKA mechanism designed in our scheme requires participating vehicles to establish session keys with the same FN, which poses a certain limitation. How to perform batch AKA operations among different vehicles and different FNs within a short time interval is a key direction for our future research.

Author Contributions: Conceptualization, L.L., J.L. and C.C.; methodology, L.L. and H.Z. ; validation, C.C. and H.Z.; formal analysis, L.L.; Software, S.L.; writing—original draft preparation, L.L. and S.L.; writing—review and editing, L.L., C.C., S.L. and H.Z.; visualization, L.L. and H.Z.; project administration, J.L. and C.C. All authors have read and agreed to the published version of the manuscript.

Funding: National Natural Science Foundation of China (Grant No. 62072133), Major Scientific and Technological Innovation Project of Wenzhou (ZG2023028, ZG2024013).

Data Availability Statement: This study is based on the MIRACL library, which is openly available at [https://github.com/miracl/MIRACL].

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

References

1. Cao, J.; Feng, W.; Ge, N.; Lu, J. Delay Characterization of Mobile-Edge Computing for 6G Time-Sensitive Services. *IEEE Internet of Things Journal* **2021**, *8*, 3758–3773. https://doi.org/10.1109/JIOT.2020.3023933.

 doi:10.20944/preprints202507.0134.v1

2. Cui, J.; Wei, L.; Zhong, H.; Zhang, J.; Xu, Y.; Liu, L. Edge Computing in VANETs-An Efficient and Privacy-Preserving Cooperative Downloading Scheme. *IEEE Journal on Selected Areas in Communications* **2020**, *38*, 1191–1204. https://doi.org/10.1109/JSAC.2020.2986617.
3. Saleem, M.A.; Li, X.; Mahmood, K.; Shamshad, S.; Alenazi, M.J.F.; Das, A.K. A Cost-Efficient Anonymous Authenticated and Key Agreement Scheme for V2I-Based Vehicular Ad-Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems* **2024**, *25*, 12621–12630. https://doi.org/10.1109/TITS.2024.3383670.
4. Lin, C.C.; Deng, D.J.; Yao, C.C. Resource Allocation in Vehicular Cloud Computing Systems With Heterogeneous Vehicles and Roadside Units. *IEEE Internet of Things Journal* **2018**, *5*, 3692–3700. https://doi.org/10.1109/JIOT.2017.2690961.
5. Awais, S.M.; Yucheng, W.; Mahmood, K.; Alenazi, M.J.F.; Bashir, A.K.; Das, A.K.; Lorenz, P. Provably Secure and Lightweight Authentication and Key Agreement Protocol for Fog-Based Vehicular Ad-Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems* **2024**, *25*, 21107–21116. https://doi.org/10.1109/TITS.2024.3452928.
6. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog Computing and Its Role in the Internet of Things. In Proceedings of the ACM SIGCOMM International Conference on Mobile Cloud Computing, Helsinki, Finland, 2012; pp. 13–16.
7. Kenney, J.B. Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proceedings of the IEEE* **2011**, *99*, 1162–1182. https://doi.org/10.1109/JPROC.2011.2132790.
8. Jiang, D.; Delgrossi, L. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. In Proceedings of the VTC Spring 2008 - IEEE Vehicular Technology Conference, 2008, pp. 2036–2040. https://doi.org/10.1109/VETECS.2008.458.
9. Islam, S.H.; Biswas, G.P. Design of Two-Party Authenticated Key Agreement Protocol Based on ECC and Self-Certified Public Keys. *Wireless Personal Communications* **2015**, *82*, 2727–2750.
10. Dang, L.; Xu, J.; Cao, X.; Li, H.; Chen, J.; Zhang, Y.; Fu, X. Efficient identity-based authenticated key agreement protocol with provable security for vehicular ad hoc networks. *International Journal of Distributed Sensor Networks* **2018**, *14*, 155014771877254.
11. Deng, L.; Shao, J.; Hu, Z. Identity based two-party authenticated key agreement scheme for vehicular ad hoc networks. *Peer-to-Peer Networking and Applications* **2021**.
12. Xie, Q.; Wong, D.S.; Wang, G.; Tan, X.; Chen, K.; Fang, L. Provably Secure Dynamic ID-Based Anonymous Two-Factor Authenticated Key Exchange Protocol With Extended Security Model. *IEEE Transactions on Information Forensics and Security* **2017**, *12*, 1382–1392. https://doi.org/10.1109/TIFS.2017.2659640.
13. Li, X.; Yang, D.; Zeng, X.; Chen, B.; Zhang, Y. Comments on 'Provably Secure Dynamic Id-Based Anonymous Two-Factor Authenticated Key Exchange Protocol With Extended Security Model'. *IEEE Transactions on Information Forensics and Security* **2019**, *14*, 3344–3345. https://doi.org/10.1109/TIFS.2018.2866304.
14. Liu, Y.; Wang, Y.; Chang, G. Efficient Privacy-Preserving Dual Authentication and Key Agreement Scheme for Secure V2V Communications in an IoV Paradigm. *IEEE Transactions on Intelligent Transportation Systems* **2017**, *18*, 2740–2749. https://doi.org/10.1109/TITS.2017.2657649.
15. Lee, T.F.; Hsiao, C.H.; Hwang, S.H.; Lin, T.H. Enhanced smartcard-based password-authenticated key agreement using extended chaotic maps. *PLoS ONE* **2017**, *12*, e0181744.
16. Dua, A.; Kumar, N.; Das, A.K.; Susilo, W. Secure Message Communication Protocol Among Vehicles in Smart City. *IEEE Transactions on Vehicular Technology* **2018**, *67*, 4359–4373. https://doi.org/10.1109/TVT.2017.2780183.
17. Vijayakumar, P.; Azees, M.; Kozlov, S.A.; Rodrigues, J.J.P.C. An Anonymous Batch Authentication and Key Exchange Protocols for 6G Enabled VANETs. *IEEE Transactions on Intelligent Transportation Systems* **2022**, *23*, 1630–1638. https://doi.org/10.1109/TITS.2021.3099488.
18. Sun, Y.; Cao, J.; Ma, M.; Zhang, Y.; Li, H.; Niu, B. EAP-DDBA: Efficient Anonymity Proximity Device Discovery and Batch Authentication Mechanism for Massive D2D Communication Devices in 3GPP 5G HetNet. *IEEE Transactions on Dependable and Secure Computing* **2022**, *19*, 370–387.
19. Madanchi, M.; Abolhassani, B. Authentication and Key Agreement Based Binary Tree for D2D Group Communication. In Proceedings of the 2020 28th Iranian Conference on Electrical Engineering (ICEE), 2020, pp. 1–5. https://doi.org/10.1109/ICEE50131.2020.9260921.
20. Wei, L.; Cui, J.; Zhong, H.; Bolodurina, I.; Liu, L. A Lightweight and Conditional Privacy-Preserving Authenticated Key Agreement Scheme With Multi-TA Model for Fog-Based VANETs. *IEEE Transactions on Dependable and Secure Computing* **2023**, *20*, 422–436. https://doi.org/10.1109/TDSC.2021.3135016.

 doi:10.20944/preprints202507.0134.v1

21. Qiao, H.; Dong, X.; Jiang, Q.; Ma, S.; Liu, C.; Xi, N.; Shen, Y. Anonymous Lightweight Authenticated Key Agreement Protocol for Fog-Assisted Healthcare IoT System. *IEEE Internet of Things Journal* **2023**, *10*, 16715–16726. https://doi.org/10.1109/JIOT.2023.3270300.
22. Jiang, Q.; Zhang, N.; Ni, J.; Ma, J.; Ma, X.; Choo, K.K.R. Unified Biometric Privacy Preserving Three-Factor Authentication and Key Agreement for Cloud-Assisted Autonomous Vehicles. *IEEE Transactions on Vehicular Technology* **2020**, *69*, 9390–9401. https://doi.org/10.1109/TVT.2020.2971254.
23. Cui, J.; Zhang, X.; Zhong, H.; Zhang, J.; Liu, L. Extensible Conditional Privacy Protection Authentication Scheme for Secure Vehicular Networks in a Multi-Cloud Environment. *IEEE Transactions on Information Forensics and Security* **2020**, *15*, 1654–1667. https://doi.org/10.1109/TIFS.2019.2946933.
24. Wazid, M.; Bagga, P.; Das, A.K.; Shetty, S.; Rodrigues, J.J.P.C.; Park, Y. AKM-IoV: Authenticated Key Management Protocol in Fog Computing-Based Internet of Vehicles Deployment. *IEEE Internet of Things Journal* **2019**, *6*, 8804–8817. https://doi.org/10.1109/JIOT.2019.2923611.
25. Saleem, M.A.; Mahmood, K.; Kumari, S. Comments on "AKM-IoV: Authenticated Key Management Protocol in Fog Computing-Based Internet of Vehicles Deployment". *IEEE Internet of Things Journal* **2020**, *7*, 4671–4675. https://doi.org/10.1109/JIOT.2020.2975207.
26. Ma, M.; He, D.; Wang, H.; Kumar, N.; Choo, K.K.R. An Efficient and Provably Secure Authenticated Key Agreement Protocol for Fog-Based Vehicular Ad-Hoc Networks. *IEEE Internet of Things Journal* **2019**, *6*, 8065–8075. https://doi.org/10.1109/JIOT.2019.2902840.
27. Cui, J.; Liu, X.; Zhong, H.; Zhang, J.; Wei, L.; Bolodurina, I.; He, D. A Practical and Provably Secure Authentication and Key Agreement Scheme for UAV-Assisted VANETs for Emergency Rescue. *IEEE Transactions on Network Science and Engineering* **2024**, *11*, 1454–1468. https://doi.org/10.1109/TNSE.2023.3323972.
28. Zhou, Y.; Cao, L.; Qiao, Z.; Xu, R.; Han, Y.; Xing, J.; Yang, B.; Xia, Z.; Zhang, M. A Novel Cloud-Assisted Authentication Key Agreement Protocol for VANET. *IEEE Transactions on Vehicular Technology* **2024**, *73*, 13526–13541. https://doi.org/10.1109/TVT.2024.3390654.
29. Cui, J.; Wei, L.; Zhang, J.; Xu, Y.; Zhong, H. An Efficient Message-Authentication Scheme Based on Edge Computing for Vehicular Ad Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems* **2019**, *20*, 1621–1632. https://doi.org/10.1109/TITS.2018.2827460.
30. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password-Based Authenticated Key Exchange in the Three-Party Setting. In Proceedings of the Public Key Cryptography - PKC 2005; Vaudenay, S., Ed., Berlin, Heidelberg, 2005; pp. 65–84.
31. Liu, G.; Li, H.; Liang, Y.; Le, J.; Wang, N.; Mu, N.; Liu, Z.; Liu, Y.; Xiang, T. PSRAKA: Physically Secure and Robust Authenticated Key Agreement for VANETs. *IEEE Transactions on Vehicular Technology* **2024**, pp. 1–15. https://doi.org/10.1109/TVT.2024.3522666.
32. Han, Y.; Guo, H.; Liu, J.; Ehui, B.B.; Wu, Y.; Li, S. An Enhanced Multifactor Authentication and Key Agreement Protocol in Industrial Internet of Things. *IEEE Internet of Things Journal* **2024**, *11*, 16243–16254. https://doi.org/10.1109/JIOT.2024.3355228.
33. Miracl cryptographic SDK, 2019.