Scientific Reports

https://doi.org/10.1038/s41598-025-31114-x

Article in Press

SOBV-FEF: secure lightweight data offloading base in blockchain technology for internet of vehicles enabled handover UAVs within a Fog-Edge federation

Received: 4 May 2025

Accepted: 29 November 2025

Published online: 04 December 2025

Cite this article as: Salami Y. SOBV-FEF: secure lightweight data offloading base in blockchain technology for internet of vehicles enabled handover UAVs within a Fog-Edge federation. *Sci Rep* (2025). https://doi.org/10.1038/s41598-025-31114-x

Yashar Salami

SOBV-FEF: Secure lightweight Data Offloading Base in Blockchain Technology for Internet of Vehicles Enabled Handover UAVs within a Fog-Edge Federation

[a]Yashar Salami *
[a] Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

[a] Faculty of Computer and Information Technologies, Cappadocia University, Ürgüp, Nevsehir, Turkey

yashar.salami@kapadokya.edu.tr

Abstract— *The Internet of Things (IoT) has improved efficiency and quality of life by connecting devices to the internet. It has seen success in areas such as smart vehicles and Unmanned Aerial Vehicles (UAVs), but faces processing limitations due to the need to send large amounts of data to other devices for processing. When heavy processing is required, it uses offloading techniques to send the data to other devices for processing. Secure data offloading transmission remains a fundamental challenge in this field. This paper presents an innovative authentication and key exchange method that uses Elliptic Curve Cryptography (ECC) and incorporates Handover for secure offloading, offering a safe, lightweight solution within a blockchain network. To evaluate the resistance of the proposed scheme against active and passive attacks, we employed the AVISPA tool to apply both formal and informal methods. Subsequently, to demonstrate the scheme's lightweight nature, we examined it with respect to computation and communication costs, the number of bits used, and security requirements. Additionally, we simulated the proposed scheme using the NS3 tool in two scenarios: urban and highway, with varying numbers of vehicles. The results indicate that the proposed scheme performs acceptably in urban and highway scenarios.*

Keywords— IoT, Blockchain, UAVs, Secure, Fog-Edge

1. Introduction

The IoT has revolutionized how humans interact with the world around them, enabling seamless communication between devices and providing unprecedented convenience[1]. IoT simplifies daily tasks from smart homes to healthcare, allowing devices to automate processes, gather real-time data, and make intelligent decisions without human intervention [2]. The need for secure, efficient, scalable infrastructures becomes critical as IoT grows[3].

Blockchain technology, a transformative solution, can fundamentally redefine security and trust in the IoT [4]*. By decentralizing data management, blockchain provides a transparent, tamper-proof ledger that ensures the integrity of communications between IoT devices* [5][6]*. Without relying on a central authority, blockchain makes IoT networks more resistant to cyberattacks and single points of failure*[7]*. Additionally, blockchain's smart contracts allow for the automation of secure device transactions, further enhancing efficiency* [8], [9]*. This potential for transformation inspires us to explore its integration within a fog-edge computing consortium.*

Blockchain integration within a fog-edge computing consortium can transform offloading in the Internet of Vehicles (IoV) context, where intelligent vehicles rely on rapid, secure data exchange[10], [11]*. Fog and edge computing bring data processing closer to the source, reducing latency and enhancing performance*[5], [12]*. By combining blockchain with these technologies, offloading tasks such as vehicle diagnostics, real-time navigation, and traffic management becomes more secure and efficient* [13]*.*

UAVs are crucial in connecting and coordinating with intelligent vehicles, offering significant advantages in navigation, traffic management, and environmental data collection[14] [15]*. These UAVs can transmit real-time information about road conditions, traffic, and obstacles to intelligent vehicles, enabling them to make better decisions and choose optimal routes*[16], [17]*. In emergencies, UAVs can also be a backup for innovative vehicles, providing real-time critical information*[18]*. This interaction not only enhances safety but also improves the efficiency of transportation systems, contributing to reduced traffic congestion and pollution*[19]*. Integrating UAVs with intelligent vehicles can create an innovative and sustainable transportation ecosystem*[20]*.*

Yashar et al. proposed a secure method for payload offloading in 2024, suitable for smart vehicles [21]*. However, it did not support UAV environments. In 2024, Vahid et al. introduced a new secure payload offloading method, which was also suitable for IoT environments but could not support UAV environments.* [22]*. Zinali et al.*[23] *Proposed a key exchange and mutual authentication method that, despite providing strong security, did not include support for data offloading. In 2022, Kwon et al. proposed a handover-based communication scheme for interactions between smart vehicles and UAVs; however, the scheme did not provide support for data offloading.*[24]*. In 2022, Kumar et al.*[25] *Proposed a 5G-based handover authentication scheme that incorporated blockchain and 6G technologies; however, it did not support payload offloading.*

Wang et al. proposed the B-TSCA scheme, which is based on blockchain technology and operates in a UAV environment [26]*; However, this scheme cannot support offloading and 6G technology. Zhou et al. proposed a scheme based on Conditional Privacy-Preserving Authentication and Key Agreement for roaming services in VANETs* [27]*; however, this scheme did not support blockchain and offloading. ZakeriKia et al. proposed the Robust and Anonymous Handover Authentication Scheme without Key Escrow Problem in vehicular sensor networks* [28].*In 2023, Sharma et al. proposed a lightweight privacy-preserving scheme tailored for IoT-based smart home environments. However, the scheme was specifically designed for general IoT applications and does not address the unique requirements of secure data offloading in intelligent vehicular networks* [29].*In 2024, Dhiman et al. proposed a mutual authentication scheme for smart IoT home networks. While effective in the context of smart homes, the scheme was not designed for vehicular environments and does not support secure data offloading in intelligent transportation systems*[30]*. In 2025, Dhiman et al. proposed another mutual authentication and addressing scheme for IoT networks. Although this work introduced improvements in addressing and authentication mechanisms for IoT environments, it remained focused on static smart home contexts and did not address the requirements of secure data offloading in dynamic vehicular networks*[31]*. Neha et al. In 2025, a multifactor remote user authentication mechanism was proposed for IoT networks. While the scheme enhances user anonymity and authentication in general IoT scenarios, it does not address the specific requirements of secure data offloading in intelligent vehicular environments*[32].*In 2024, Salami proposed the SOBT-UF scheme for secure offloading in blockchain-based intelligent transportation systems using 5G-enabled UAVs within a fog–edge infrastructure. While the scheme introduced an innovative integration of technologies, it lacked comprehensive resistance against active and passive attacks—being effective only in limited scenarios—and did not qualify as a lightweight solution*[33]*.*

Although this scheme supported authentication and key exchange, it could not support blockchain and offloading. This paper proposes a secure offloading approach to address this challenge, as prior methods have not closed the gap in offloading between vehicle-to-vehicle communication via UAVs in fog and cloud environments. The proposed scheme provides secure offloading resilient to active and passive attacks. It also offers real-time data processing and a robust, scalable infrastructure for future transportation systems.

1.1 Motivated

The explosive growth of IoV applications—from autonomous driving and cooperative perception to real-time traffic orchestration—places unprecedented demands on in-vehicle and

edge-layer computing platforms. However, these platforms are inherently limited by their constrained battery life and processing capabilities. Secure offloading of heavy computational tasks to nearby nodes has emerged as a promising remedy; yet, extant schemes typically assume a perpetually secure environment and rely on stationary fog nodes or Roadside Units. Such infrastructures not only incur high deployment and maintenance costs but also suffer connectivity disruptions when vehicles move beyond their coverage radius. UAVs offer a transformative alternative: their mobility ensures continuous, line-of-sight coverage, eliminating blind spots and maintaining uninterrupted supervision. Despite this advantage, previous research has largely neglected drone-assisted secure offloading and the complex security threats inherent to real-world vehicular networks—namely, man-in-the-middle, replay, and message-forgery attacks. Ensuring mutual authentication under these dynamic conditions is therefore paramount. To address these gaps, we propose SOBV-FEF, a lightweight, blockchain-enabled secure offloading framework that harnesses Elliptic Curve Cryptography for robust authentication and key exchange, and seamlessly integrates UAV-based handover support. We rigorously evaluate its resilience to active and passive attacks through formal and informal analyses using the AVISPA tool, and demonstrate its efficiency through detailed assessments of computational and communication overhead, bit-length requirements, and security guarantees. NS-3 simulations across urban and highway scenarios with varying vehicle densities confirm that SOBV-FEF not only thwarts sophisticated adversarial attacks but also substantially reduces system overhead while preserving high performance in realistic deployments.

1.2 Paper contribution

This paper presents an innovative authentication and key exchange method for the Internet of Things (IoT). It is based on Elliptic Curve Cryptography (ECC) and uses Handover to securely offload data in a blockchain network.

The main contributions of this paper are as follows:

- ***Propose** a novel security method: Develop a tailored approach to enhance data transmission security in IoT environments, addressing unique vulnerabilities.*
- ***Evaluate** security comprehensively: Apply formal and informal methods to assess the proposed scheme's resilience against active and passive attacks, strengthening its validity and reliability.*

- ***Analyze*** *computational efficiency: Conduct a detailed cost analysis of computation, communication, and bit usage, demonstrating the proposed scheme's efficiency and lightweight design.*
- ***Demonstrate*** *performance through simulation: Utilize the NS3 tool to conduct operational simulations in urban and highway scenarios, analyzing latency, packet loss, packet delivery, and throughput to validate the method's effectiveness in real-world conditions.*

1.3 Organization paper

The organization of this paper is as follows: The second section introduces the network model utilized, along with the assumptions of the attack model and the problem statement. The third section presents the proposed scheme and its various components for secure offloading. Section 4 presents a formal security analysis of the proposed scheme using the AVISPA security tool and examines the informal attacks defined in the attack model. The fifth section evaluates the proposed scheme regarding processing, communication, storage costs, and security requirements. The sixth section discusses the simulation of the proposed scheme using the NS3 tool across different scenarios. Finally, the last section presents the conclusions.

2. Network Architecture

This section presents the network architecture, details the underlying assumptions, defines the problem statement, and reviews ECC and the associated threat model.

2.1 Network Architecture

We utilize a three-tier network architecture consisting of cloud, fog, and edge layers. The cloud layer is positioned at the highest level, has superior processing capabilities compared to the other layers, and communicates with the underlying fog layer. This cloud layer processes a large volume of data and employs advanced analytical and predictive models to optimize the system's performance and accuracy. As a central hub, this layer stores and analyzes complex and big data to facilitate decision-making across the network.

The middle layer, the fog layer, operates with less processing power than the cloud. However, this layer is crucial for collecting data from the edge layer. The fog layer can perform local analysis using edge computing methods and enable real-time processing close to the data source. It includes various devices, with those closest to the edge layer recognized as fog nodes. These fog nodes act as intermediaries, connecting the edge layer to the cloud. A

blockchain network is also embedded within the fog layer, interacting with devices to ensure data security and integrity through decentralized verification. A set of fog nodes can communicate with each other within the fog layer. If these nodes are located near one another, they can form a region and cover intelligent vehicles in edge-layer communication technology. The UAV can act as a substitute when there are no roadside units (RSUs) and can communicate with vehicles equipped with communication technology to transfer data to a higher layer. Furthermore, UAVs in each area can perform "Handover " operations through a UAV and communicate with other UAVs in that region. These capabilities significantly enhance communication and coordination between devices and vehicles, creating an integrated and intelligent network. At the base of this architecture lies the edge layer, which includes the Internet of Vehicles (IoV). Although these vehicles have less processing power compared to the higher layers, they can locally collect and process data and, if necessary, transfer critical information to nearby fog nodes. Through this connection, these vehicles can exchange important real-time data, facilitating quick decision-making in crucial situations. If needed, these vehicles can also communicate with the cloud layer through the fog nodes. Figure 1 illustrates the Network architecture, comprising edge, fog, and cloud layers, with UAV-based coordination.

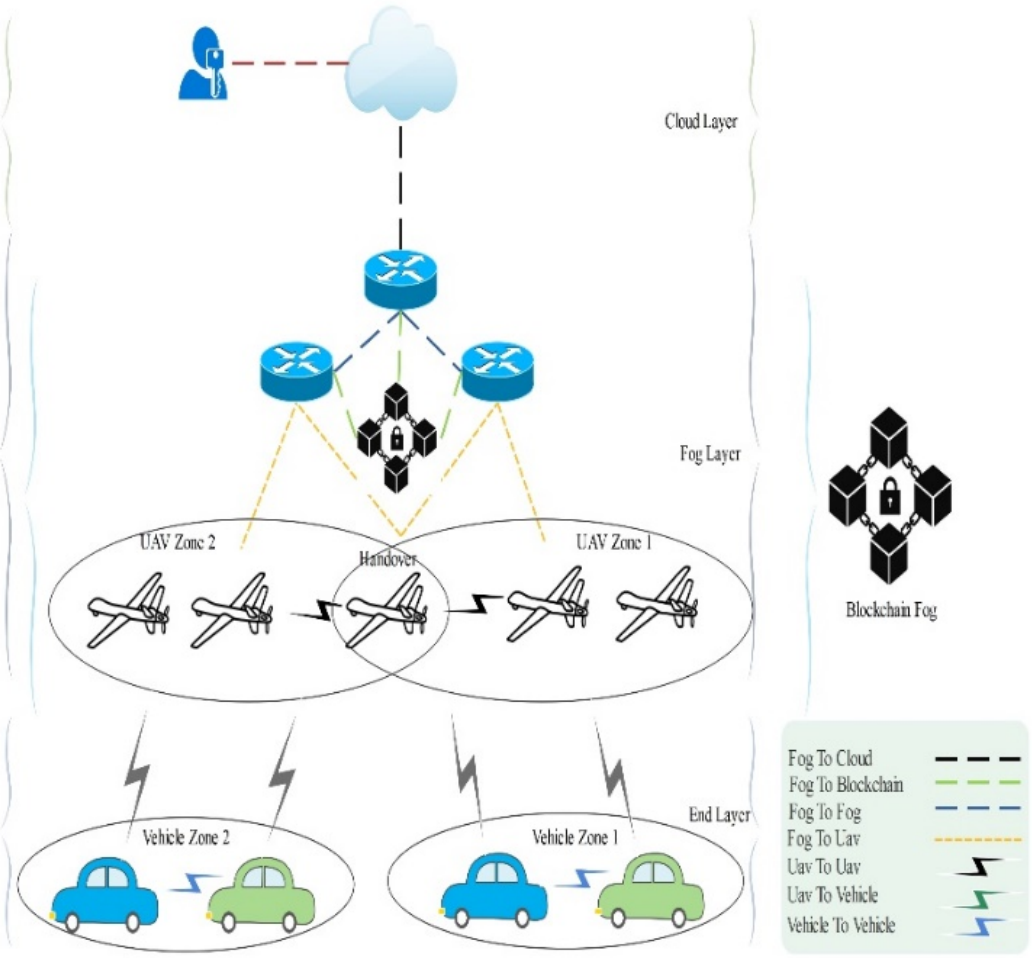

Fig 1. Network Architecture Model.

2.2 Assumptions

- ***Unsecured Communication Channels at the Edge Layer:*** *In the network architecture's edge layer, communication channels remain unprotected, and any data transferred in this layer is exposed to threats such as unauthorized access and security breaches.*
- ***Synchronized Device Scheduling:*** *All devices throughout the network operate on a unified and synchronized schedule, ensuring that each device follows a fixed and predetermined timing configuration across the entire system.*
- ***Heterogeneity of Network Devices:*** *The devices present in the network are heterogeneous in terms of their capabilities, functionalities, and potentially the*

communication protocols they use, which introduces complexities in network management.

- ***Secure Communication Between Fog and Cloud Layers:*** *The fog and cloud layers recognize each other and establish a secure communication channel. This ensures that all data exchanged between these two layers is protected from unauthorized access or tampering.*
- ***Secure Communication Between Fog Nodes and Roadside Units:*** *Fog nodes and roadside units, recognized entities in the network, maintain a secure communication channel between themselves, ensuring that data is transmitted securely and without risk.*
- ***Communication of UAVs :****Several UAVs close to one another can cover a specific area. In this case, each location has a unique number called the area ID. This ID helps UAVs more accurately identify their positions and the assigned tasks.*
- ***Handover:*** *Between two areas, the airborne unit can act as a handover, maintaining the communication of active UAVs in both regions.*
- ***Connection****: In the end layer, all devices utilize 6G technology for their communications. This technology facilitates unprecedented data exchange by providing extremely high speeds and low latency. Additionally, 6G enables the simultaneous connection of many devices, significantly contributing to the Internet of Things (IoT) expansion.*
- ***Permissioned Blockchain:*** *In this type of blockchain, with the used architecture, users must have specific permissions to access the network, and depending on their access type, they can receive different roles.*

2.3 Threat model

In this paper, the Dolev-Yao (D&Y) adversarial model is employed to assess potential security vulnerabilities within the communication infrastructure systematically. This model assumes the presence of a powerful adversary capable of intercepting, observing, and manipulating the exchange of messages between legitimate parties. Operating as a man-in-the-middle, the adversary may obtain unauthorized access to confidential information, often without detection by the communicating entities. The D&Y framework characterizes two principal classes of adversarial behavior: passive eavesdropping, wherein the attacker discreetly monitors message exchanges to extract sensitive information without modifying the content; and active interference, whereby the attacker deliberately alters, injects, or blocks messages with the intent of disrupting communication or deceiving one or more participants.

The goal is to identify security vulnerabilities and evaluate the design's resilience against potential threats, including impersonation, man-in-the-middle attacks, replay attacks, and other related adversarial strategies. For this reason, we continue to conduct both formal and informal security testing to assess the resistance of our proposed design against various attacks, including:

- *Denial of Service (DoS)*
- *Impersonation*
- *Eavesdropping*
- *Traffic Analysis*
- *Data Leakage*
- *Verification Table Leakage*
- *Privileged Insider Attacks*
- *Session-specific Random Number Leakage*
- *Forward Secrecy*
- *Brute-force / Offline Password Guessing*
- *Stolen-Verifier Attacks*
- *Modification Attacks*

This comprehensive evaluation approach ensures a thorough assessment of the scheme's security posture, identifying potential weaknesses and verifying its ability to withstand diverse attack vectors.

2.4 Review ECC

ECC is a public key encryption method based on the algebraic structure of elliptic curves over finite fields. The equations of these curves are represented in the form: $y + axy + by = x3 + cx2 + dx + e$ *In this equation, R {a,b,c,d,e} denotes real numbers that must satisfy specific conditions. In these curves, a point may be considered zero or a point at infinity. For further information, you can refer to [12].*

2.5 Problem statement

Considering the limitations of processing power and energy in edge-layer vehicles, when a ***vehicle*** *encounters data requiring significant computational resources, it must transfer it to another vehicle with higher processing power and sufficient energy reserves. This process, known as offloading, allows the more capable vehicle to perform the computational tasks and return the results. However, offloading may pose security risks, as malicious entities can intercept, manipulate, or monitor the transferred data. To counter these threats, both parties*

involved in the data exchange must verify each other's identities before any transfer occurs. One of the main challenges in this context is ensuring secure offloading between vehicles. Protecting the offloading process involves preventing unauthorized access to the data and ensuring it reaches its destination without alteration or damage. This requires strong authentication mechanisms and encryption techniques to secure the data during transmission. Achieving secure and efficient offloading is crucial for maintaining the integrity and reliability of the network, primarily when vehicles must rely on external processing resources to perform complex tasks. This paper presents a novel authentication and key exchange method utilizing Elliptic ECC, which integrates the Handover to facilitate secure offloading.

3. Proposed schema

This section provides a detailed presentation of the proposed scheme and its steps.

3.1Notations

Table 1 shows the Notations used in the scheme.

Table 1. Notations.

No.	***Description***	***Notations***
1	*The ID of Vehicle i*	Vi
2	*The ID of Vehicle j*	Vj
3	*The ID of the UAV fi*	Ufi
4	*ID of UAV fj*	Ufj
5	*The ID of the UAV handover*	Uh
6	*ID of fog*	F
7	*ID Blockchain*	B
8	*Timestamp of Vi, Vj, Ufi, Ufj, Uh, F, B*	T_i ,T_j ,T_{Fi} ,T_{Fj} ,T_{Uh} ,T_F T_B
9	*KUV*	*session key*
10	*Rf*	*Request offloading*
11	*Pow*	*Power requirement*
12	*Expire time*	*ΔT*
13	*Hash function*	(.)
14	*XOR function*	⊕

3.2 Register and login phase

Figure 2 shows the register and login flowchart.

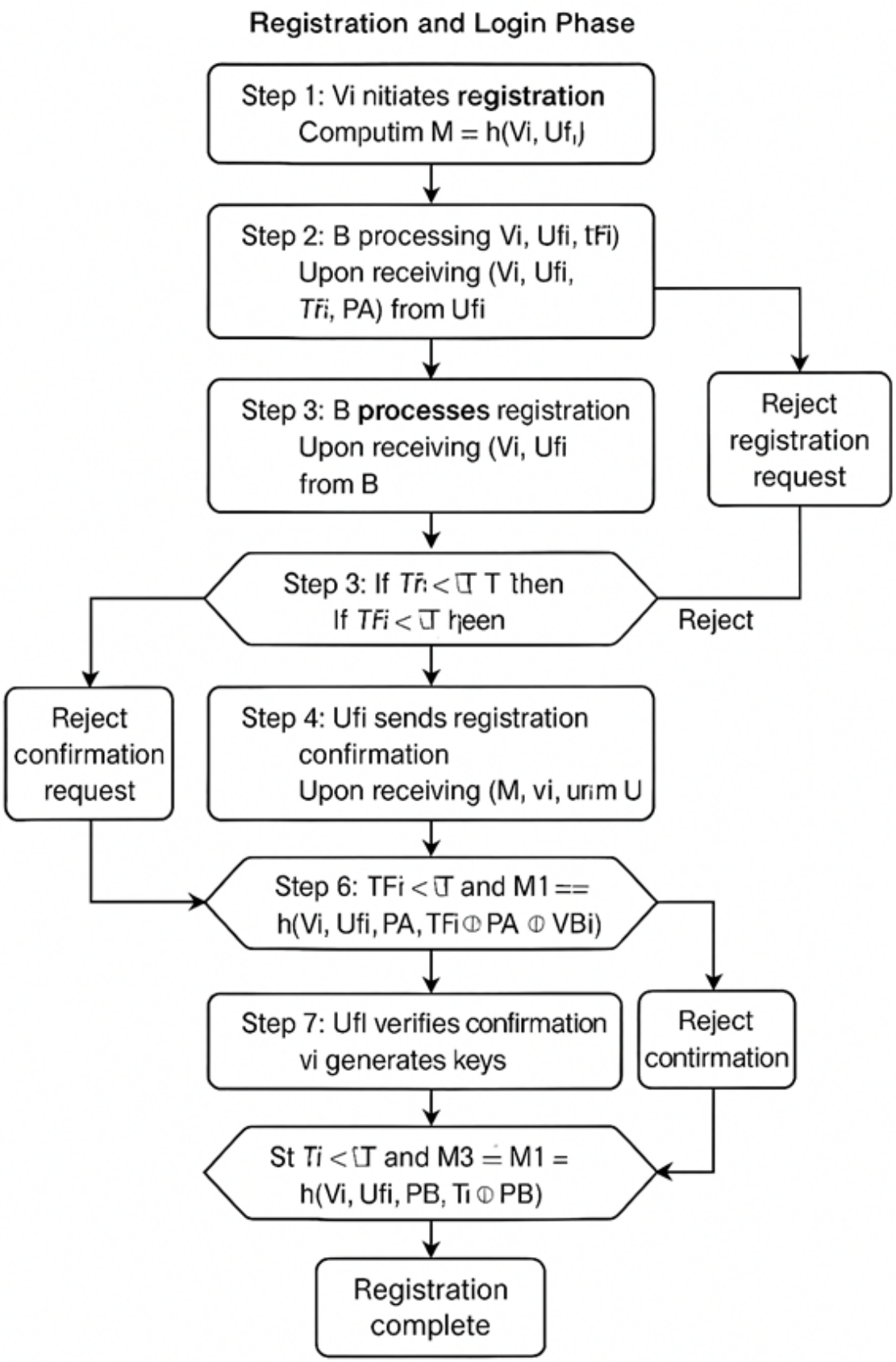

Fig 2. Register and login flowchart.

***Step 1**: Initiation of Registration Request*

The user entity (Vi) initiates the registration process by generating a secure message to the intermediate entity (Ufi).

- ***Computation**: Vi computes a hash value M = h (Vi, Ufi, T_i), where:*
  - *h denotes a cryptographic hash function (e.g., SHA-256).*
  - *Vi is the unique identifier of the user.*
  - *Ufi is the identifier of the intermediate entity.*
  - *T_i is a timestamp ensuring message freshness.*
- ***Transmission**: Vi sends {M, Vi, Ufi, T_i} to Ufi.*

Step 2: Verification and Request Forwarding by Ufi

Upon receiving Vi's message, Ufi validates the request and prepares it for further processing by generating a public key and forwarding it to entity B.

- ***Validation**:*
  - ***Timestamp Check**: Ufi verifies that T_i < ΔT, where ΔT is a predefined time window for message validity.*
  - ***Hash Verification**: Ufi recomputes M' = h (Vi, Ufi, T_i) and checks if M = M'. If true, the message is authentic.*
- ***Key Generation**: If verification succeeds, Ufi selects a private key UA (a scalar value) and computes the corresponding public key PA = UA * G, where G is the base point of an elliptic curve.*
- ***Transmission**: Ufi sends a registration request {Vi, Ufi, T_Fi, PA} to entity B, where T_Fi is a new timestamp generated by Ufi.*

Step 3: Processing and Token Generation by B

Entity B, typically a backend server or trusted authority, processes the request from Ufi, generates a registration token, and forwards relevant data to entity F.

- ***Validation:** B checks if T_Fi < ΔT to confirm the request's freshness.*
- ***Computation:** B computes a registration token VBi = h (Vi ⊕ Ufi ⊕ B), where:*
  - *⊕ denotes the bitwise XOR operation.*
  - *B is the identifier or secret of entity B.*
- ***Storage and Transmission:***
  - *B stores VBi locally.*

- *B sends VBi to Ufi.*
- *B forwards {VBi, Vi, Ufi} to entity F.*

***Step 4**: Data Storage by F*

Entity F, acting as a storage or federation server, receives and stores the registration data from B.

- ***Action**: F stores {VBi, Vi, Ufi} in its secure database.*

Step 5: Response Preparation by Ufi

After receiving VBi from B, Ufi constructs a response to Vi to continue the registration process.

- ***Computation**: Ufi computes a hash M1 = h (Vi, Ufi, PA, T_Fi ⊕ PA ⊕ VBi), incorporating the public key PA, timestamp T_Fi, and token VBi.*
- ***Transmission**: Ufi sends {M1, Vi, Ufi, PA, T_Fi, VBi} to Vi.*

Step 6: Verification and Key Exchange by Vi

Vi receives Ufi's response, verifies its authenticity, and generates cryptographic keys to establish a shared secret.

- ***Validation:***
  - ***Timestamp Check:** Vi verifies that T_Fi < ΔT.*
  - ***Hash Verification:** Vi recomputes M1' = h (Vi, Ufi, PA, T_Fi ⊕ PA ⊕ VBi) and checks if M1 = M1'.*
- ***Key Generation and Storage:***
  - *If verification succeeds, Vi stores VBi.*
  - *Vi selects a private key VA and computes the public key PB = VA * G.*
  - *Vi computes the shared secret KUV = VA * PA (using ECC Diffie-Hellman key exchange) and stores KUV.*
- ***Message Preparation:** Vi computes M3 = h (Vi, Ufi, PB, T_i ⊕ PB) and sends {M3, Vi, Ufi, PB, T_i} to Ufi.*

4. Step 7: Final Verification and Registration Completion

Ufi receives the message from Vi, verifies it, and finalizes the registration by computing and storing the shared secret.

- ***Validation:***

- ***Timestamp Check:*** *Ufi verifies that T_i < ΔT.*
- ***Hash Verification:*** *Ufi recomputes M3' = h (Vi, Ufi, PB, T_i ⊕ PB) and checks if M3 = M3'.*

- ***Key Computation:*** *If verification succeeds, Ufi computes KUV = UA * PB and stores KUV.*
- ***Completion:*** *The registration process is complete, with Vi and Ufi sharing the secret key KUV.*

Figure 3. Show Register and login phase

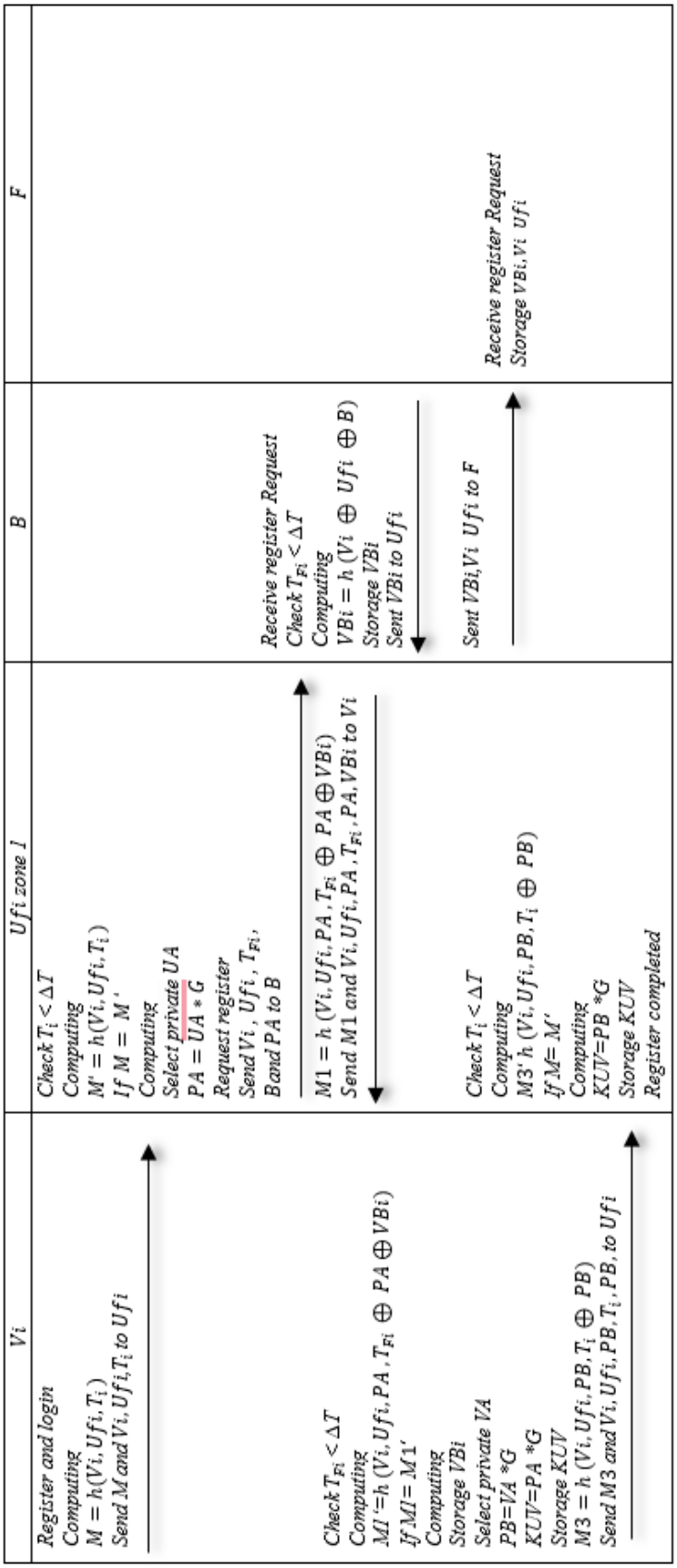

Fig 3. Register and login phase.

3.3 ***Request offloading phase***

Figure 4 shows the Request offloading flowchart.

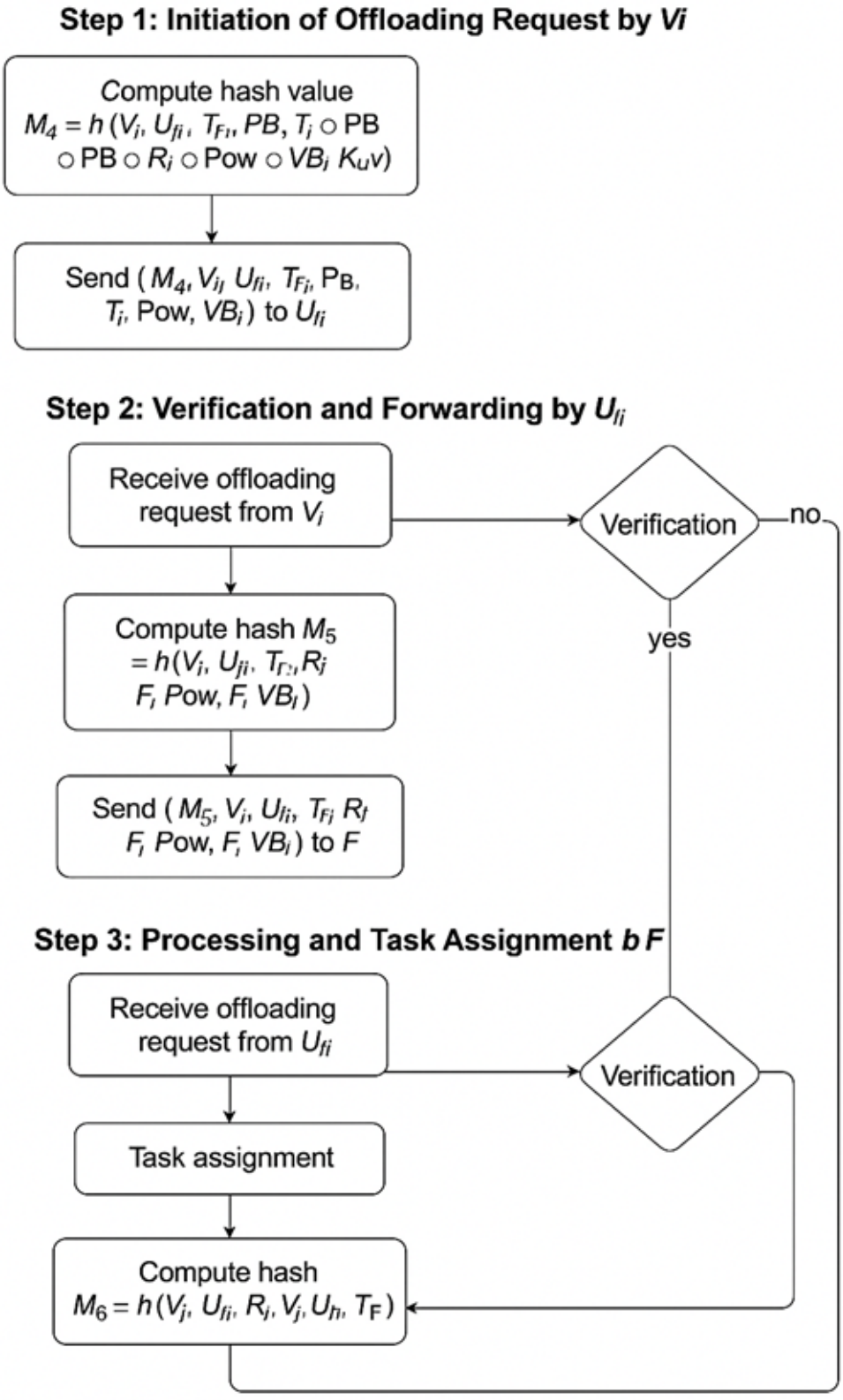

Fig 4. Request offloading flowchart.

Step 1: Initiation of Offloading Request by Vi

The user entity (Vi) initiates the offloading process by generating a secure request for task delegation that incorporates the required computational power and specifications.

- ***Computation:*** *Vi computes a hash value M4 = h (Vi, Ufi, T_Fi, PB, T_i ⊕ PB ⊕ Rf ⊕ Pow ⊕ VBi ⊕ KUV), where:*
  - *h is a cryptographic hash function (e.g., SHA-256).*
  - *Vi is the user's unique identifier.*
  - *Ufi is the identifier of the intermediate entity.*
  - *T_Fi is a timestamp generated by Ufi during the registration phase, reused here for consistency.*
  - *PB is the public key of Vi (computed as PB = VA * G during registration).*
  - *T_i is a new timestamp generated by Vi.*
  - *Rf represents the task or resource identifier for the offloading request.*
  - *Pow denotes the computational power required for the task.*
  - *VBi is the registration token computed during the registration phase.*
  - *KUV is the secret key shared between Vi and Ufi.*
  - *⊕ denotes the bitwise XOR operation.*
- ***Transmission:*** *Vi sends {M4, Vi, Ufi, T_Fi, PB, T_i, Rf, Pow, VBi} to Ufi.*

Step 2: Verification and Forwarding by Ufi

Upon receiving Vi's offloading request, the intermediate entity (Ufi) validates the message and prepares a new request to forward to the federation server (F).

- ***Validation:***

Timestamp Check: Ufi verifies that T_i < ΔT, where ΔT is a predefined time window for message validity, ensuring the request is not stale.
Hash Verification: Ufi recomputes M4' = h (Vi, Ufi, T_Fi, PB, T_i ⊕ PB ⊕ Rf ⊕ Pow ⊕ VBi ⊕ KUV) using the received parameters and the shared key KUV. If M4 = M4', the request is authentic and untampered.

- ***Computation:*** *If verification succeeds, Ufi computes a new hash M5 = h (Vi, Ufi, T_Fi, Rf, Pow, F, VBi), where F is the federation server's identifier.*
- ***Transmission:*** *Ufi sends {M5, Vi, Ufi, T_Fi, Rf, Pow, F, VBi} to F.*

Step 3: Processing and Task Assignment by F

The federation server (F) receives the offloading request from Ufi, validates it, and assigns the task to an appropriate IoV node based on the required computational power.

- ***Validation:***
  - *Timestamp Check: F verifies that T_Fi < ΔT to confirm the request's freshness.*
  - *Hash Verification: F recomputes M5' = h (Vi, Ufi, T_Fi, Rf, Pow, F, VBi) and checks if M5 = M5'. If true, the message is valid.*
- ***Task Assignment:***
  - *Storage: F stores the power requirement (Pow) for the task.*
  - *Node Selection: F selects an IoV node (Vj) from its database that meets the computational power requirement (Pow).*
- ***Computation:*** *F computes a hash M6 = h (Vi, Ufi, Rf, Vj, Uh, T_F), where:*
  - *Vj is the identifier of the selected IoV node.*
  - *Uh is the identifier of the entity responsible for handling the IoV node (e.g., a coordinator or gateway).*
  - *T_F is a new timestamp generated by F.*
- ***Transmission:*** *F sends {M6, Vi, Ufi, Rf, Vj, Uh, T_F} to Uh.*

. Figure 5 shows the Request offloading phase.

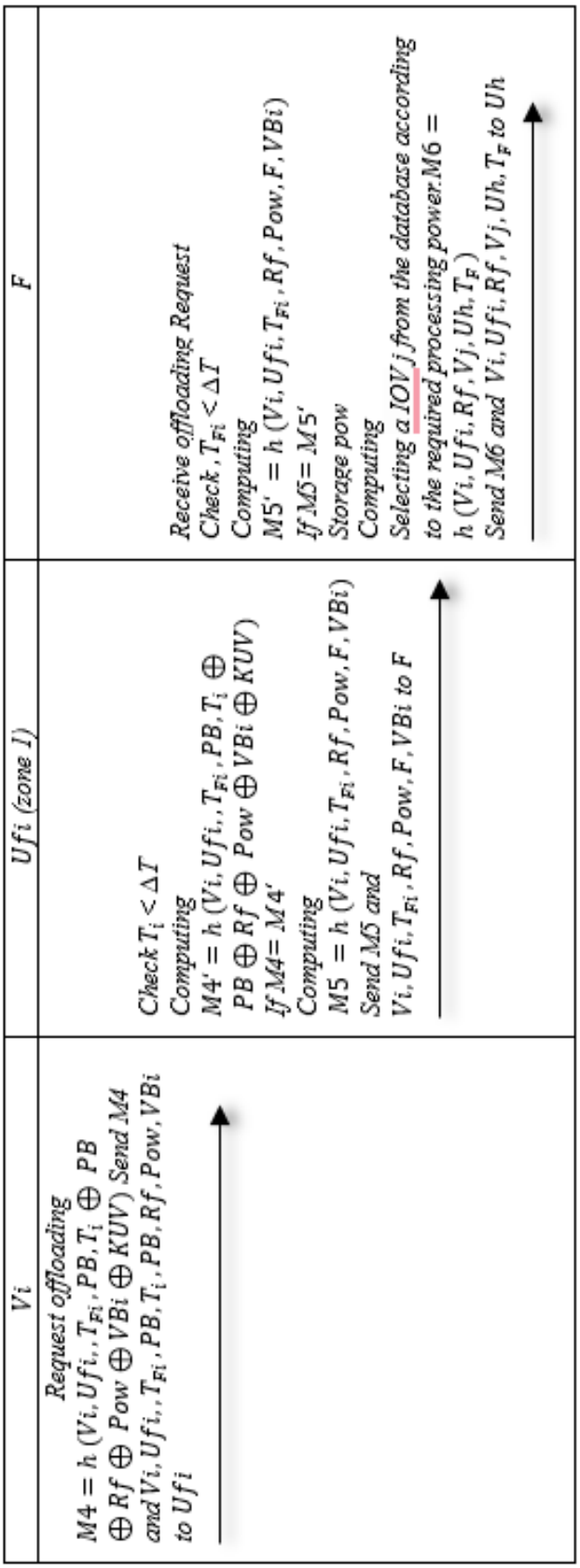

Fig 5. Request offloading phase.

3.4 ***Completed offloading phase***

Figure 6 shows the completed offloading flowchart.

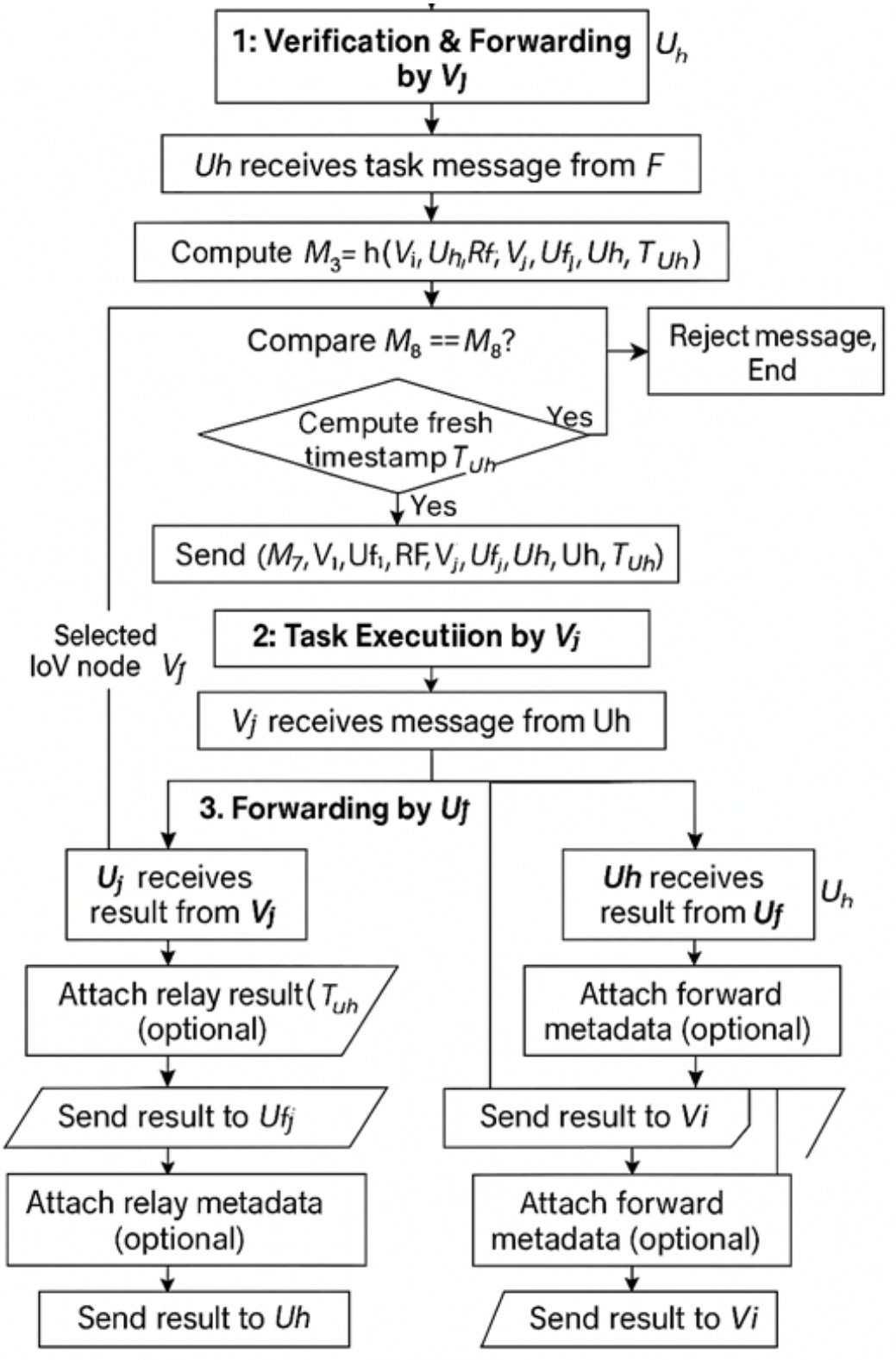

Fig 6. Completed offloading flowchart.

Step 1: Verification and Task Forwarding by Uh

The coordinating entity (Uh), responsible for managing the IoV node, receives the task assignment message from the federation server (F) and forwards it to the selected IoV node (Vj).

- **Validation:**
  - *Hash Verification: Uh recomputes the hash M6' = h (Vi, Ufi, Rf, Vj, Uh, T_F) using the received parameters, where:*
  - *Vi is the user's unique identifier.*
  - *Ufi is the identifier of the intermediate entity.*
  - *Rf is the task or resource identifier.*
  - *Vj is the identifier of the selected IoV node.*
  - *Uh is the identifier of the coordinating entity.*
  - *T_F is the timestamp generated by F.*
  - *h is a secure cryptographic hash function (e.g., SHA-256).*
  - *If M6 = M6', the message is authentic and untampered.*
- **Task Forwarding:**
  - *Node Identification: Uh identifies the IoV node Vj based on the received Vj identifier.*
  - *Message Preparation: Uh computes a new hash M7 = h (Vi, Ufi, Rf, Vj, Ufj, Uh, T_Uh), where:*
  - *Ufj is the identifier of the entity responsible for relaying results (potentially another intermediate entity or a gateway).*
  - *T_Uh is a fresh timestamp generated by Uh.*
  - **Transmission:** Uh sends {M7, Vi, Ufi, Rf, Vj, Ufj, Uh, T_Uh} to Vj.

Step 2: Task Execution and Result Generation by Vj

- *The IoV node (Vj) receives the task message from Uh, validates it, executes the computational task, and sends the results to the result-relaying entity (Ufj).*

- **Validation:**
  - *Timestamp Verification: Vj checks whether T_Uh < ΔT, where ΔT is a predefined time window for message validity, ensuring the request is not stale. (Note: The original text references T_Fj, which appears to be a typo; T_Uh is used here for consistency with the received timestamp.)*
  - *Hash Verification: Vj recomputes the hash M7' = h (Vi, Ufi, Rf, Vj, Ufj, Uh, T_Uh) and checks if M7 = M7'. (Note: The original text references M8 and M8', which seem to be errors; M7 and M7' are used here to align with the hash computed by Uh.) If true, the message is valid.*
- **Task Execution:**
  - *Upon successful validation, Vj executes the computational task identified by Rf.*
  - *Vj generates the task results, including computed data, outputs, or status information***.**
- **Transmission:** *Vj sends the offloading results to Ufj.*

Step 3: *Receive and send the Result offloading to* Uh*.*

Step 4: *Receive and send the Result offloading* Vi*. Figure 7. Show completed offloading phase.*

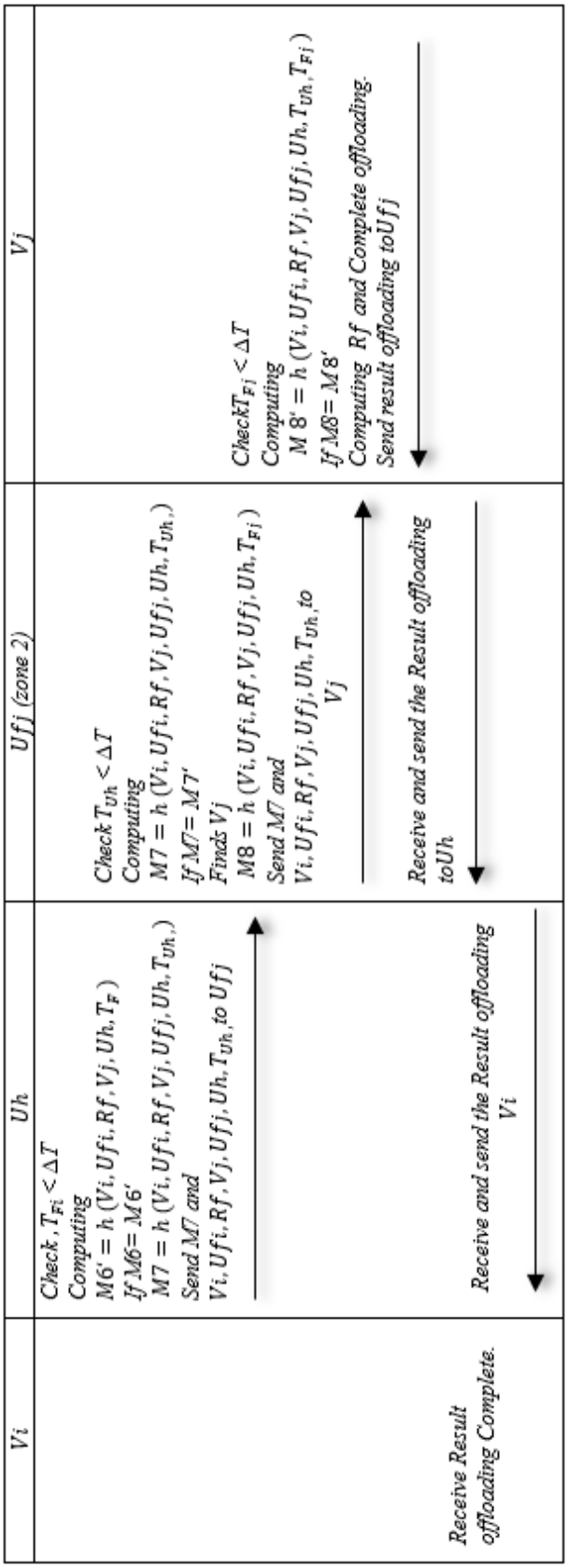

Fig 7. Completed offloading phase.

4 Formal and informal security analysis

This section provides a formal security analysis using AVISPA and an informal analysis of the attacks defined in the attack model for the proposed scheme.

4.1Avispa

AVISPA utilizes an IF (Intermediate Format) translator, which is essential for converting protocols specified in HLPSL into a low-level language for further analysis[34][35]*. The IF is then processed by AVISPA's integrated analysis tools: the On-the-Fly Model Checker (OFMC) and the Constraint Logic-based Attack Searcher (CL-AtSe)*[36][37]*. Each tool provides varied methods to assess the protocol's robustness across different scenarios, making AVISPA a powerful and versatile framework for protocol security analysis* [38]*.*

4.2Simulation result

The results shown in Figure 8 demonstrate that the proposed method utilizing the OFMC is robust against known attacks. It effectively constrained the number of sessions and conducted a thorough analysis of the target protocol. Furthermore, the processing and search times were minimal, with a satisfactory number of visited nodes and analysis depth. Overall, these findings confirm the security of the proposed method against known threats. The results shown in Figure 9 indicate that the proposed method employing CL-AtSe is resilient against known attacks. This approach successfully established boundaries on the number of sessions while thoroughly analyzing the target protocol. Additionally, the quantities of analyzed and reachable states and the efficiency of translation and computation times were all satisfactory. These outcomes confirm the security of the proposed method against known threats.

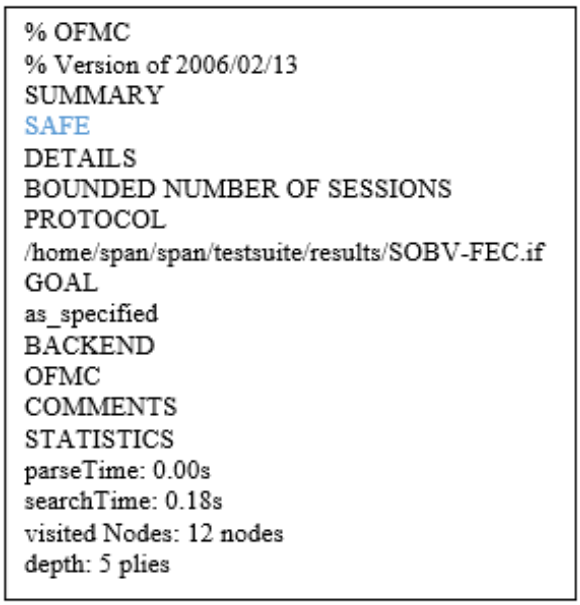

Fig 8. Result of OFMC backend.

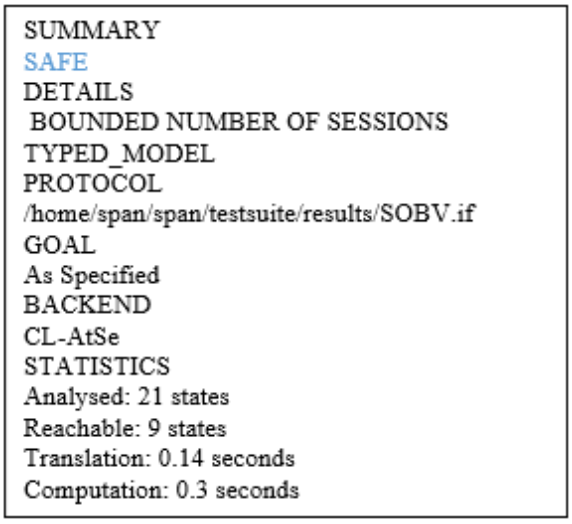

Fig 9. Result of CL-AtSe backend.

4.3 Informal Security Analysis

- ***Modification Attack***

  Every message includes a hash value that depends on sensitive inputs (IDs, timestamps, keys). Any modification will result in a hash mismatch during the verification step (e.g., $M \neq M'$), causing the protocol to reject the altered message.

- ***Denial of Service (DoS)***

  Timestamps are verified at every critical step ($T_i < \Delta T$), effectively filtering out replayed or delayed messages. Randomness and session-specific values make message reuse or duplication impractical.

- ***Impersonation***

  *All identities are linked to values derived from private keys (e.g., PA = UA * G). An attacker cannot forge a valid PA or PB without knowing the corresponding private key. Hashes used for authentication depend on these values, making impersonation infeasible.*

- ***Eavesdropping***

  Although messages are transmitted over a potentially insecure channel, their contents are protected via hash functions combined with nonces, session keys, and private values. Even if intercepted, the attacker cannot derive useful information without solving the hash function, which is computationally infeasible.

- ***Traffic Analysis***

  Including random values (e.g., Rf) and dynamic timestamps ensures that repeated protocol runs produce completely different message structures. This obfuscates traffic patterns and prevents attackers from drawing meaningful conclusions via timing or volume analysis.

- ***Data Leakage***

  Sensitive values such as KUV, Rf, or VBi are either encrypted, hashed, or protected by cryptographic transformations. Even if stored data is compromised, it cannot be directly mapped back to meaningful input without solving hard cryptographic problems.

- ***Verification Table Leakage***

  The value VBi is stored securely and only meaningful in combination with other private values. Without access to Vi, Ufi, and B, the verifier is useless to an attacker.

- ***Privileged Insider Attack***

  No single party (e.g., Ufi, F, or B) can access all the cryptographic inputs or keys required to reconstruct sensitive data. Insider access is therefore insufficient to impersonate other parties or decrypt session keys.

- ***Session-specific Random Number Leakage***

  Values like Rf are embedded in hash functions along with session-dependent parameters and cannot be extracted by reverse-engineering the hash. The use of strong one-way hash functions ensures leakage is computationally infeasible.

- ***Forward Secrecy***

  *Session keys (e.g., KUV) are derived from ephemeral public/private key pairs using elliptic curve operations (PB = VA * G, KUV = PB * UA). Compromise of long-term keys does not affect past sessions, ensuring forward secrecy.*

- ***Brute-force / Offline Password Guessing***

No plaintext passwords are transmitted or stored. Even if an attacker intercepts messages, they would need to guess multiple inputs (IDs, random values, timestamps) to verify their guesses, rendering offline attacks infeasible.

- ***Stolen-Verifier Attack***

  The verifier VBi is generated using secure hashing of combined private inputs and is of no use. The protocol does not expose password equivalents, thus mitigating risks even in database compromise.

- ***Confidentiality***

  All exchanged data is protected either through hash encapsulation or ECC-based key derivation. Without the appropriate private key or knowledge of session parameters, the attacker cannot retrieve the actual content of messages.

- ***Session Key Agreement***

  Session keys (KUV) are established securely through elliptic curve operations on independently generated private/public keys. Both parties contribute to the key, ensuring perfect forward secrecy and protection from man-in-the-middle attacks.

- ***Mutual Authentication***

  At every interaction point, the recipient verifies the hash value (M, M1, M3, etc.) to confirm message authenticity. These verifications involve identity, public key, and time-bound parameters, ensuring both sides are authenticated at every phase.

5. Performance Evaluation

This section analyzes the proposed scheme in terms of computational cost, communication cost, the number of bits used, and security requirements.

We calculate the computational cost using the MIRACL library [34] simulation used in the paper. This library simulates and estimates the execution times of essential cryptographic operations. Additionally, we utilized the E3C tool [39]*to enhance accuracy and minimize computational errors in our calculations. The following shows the different notations for the cryptographic operations, including ECC multiplication (EM), ECC addition (EA), AES encryption (SE), AES decryption (SD), modular exponentiation (ME), bilinear pairing (BP), bilinear pairing multiplication (BPM), and hash function (H). The table provides the execution times for these operations on a desktop computer and a Raspberry Pi platform. On the desktop computer, the ECC multiplication (EM) takes 2.598 milliseconds, the ECC addition (EA) takes 0.012 milliseconds, the AES encryption (SE) and decryption (SD) both take 0.001 milliseconds, the modular exponentiation (ME) takes 0.196 milliseconds, the bilinear pairing (BP) takes 6.490 milliseconds, the bilinear pairing multiplication (BPM) takes 0.813 milliseconds, and the*

hash function (H) takes 0.003 milliseconds. On the Raspberry Pi platform, the ECC multiplication (EM) takes 2.862 milliseconds, the ECC addition (EA) takes 0.017 milliseconds, the AES encryption (SE) takes 0.013 milliseconds, the AES decryption (SD) takes 0.014 milliseconds, the modular exponentiation (ME) takes 0.311 milliseconds, the bilinear pairing (BP) takes 9.244 milliseconds, the bilinear pairing multiplication (BPM) takes 1.014 milliseconds, and the hash function (H) takes 0.006 milliseconds.

5.1 Computation cost

Table 2 compares the computational costs and total execution time of the proposed scheme and five other existing schemes ([25], [26], [27],[28], [24]*. The computational costs are divided into Device, Infrastructure 1, and Infrastructure 2. Each cell in the table represents the number and type of cryptographic operations required in that particular component. The Device requires 1 ECC multiplication (Em) and five hash operations (H) for the proposed scheme. Infrastructure 1 requires 1 ECC multiplication (Em) and eight hash operations (H). Infrastructure 2 requires two hash operations (H). The total execution time for the proposed scheme is 5.778 milliseconds, which is significantly lower compared to the other schemes:* [25]*: 41.502 milliseconds,* [26]*: 19.197 milliseconds,* [27]*: 43.279 milliseconds,* [28]*: 35.057 milliseconds,* [24]*: 11.001 milliseconds The proposed scheme's lower computational costs and total execution time demonstrate its efficiency and suitability for resource-constrained environments, such as IoT and mobile devices, where computational power and energy consumption are critical factors.*

Table 2. Comparison of computation cast.

Schemes	*Device*	*Infrastructure 1 (Zone 1)*	*Infrastructure 2 (Zone 2)*	*Total Costs (ms)*
[25]	*4Em + 4H + 1BP*	*1EM + 1BP*	*2EM + 1BP + 2H*	*41.502*
[26]	*1ME + 1BPM + 1BP*	*3ME + 1BPM + 1BP*	*2ME*	*19.197*
[27]	*6EM + 2EA + 3H*	*7EM+ 2EA + 5H*	*3EM + 2EA + 4H*	*43.279*
[28]	*4Em + 4EA + 7H*	*4EM + 3EA + 6H*	*5EM + 3EA + 9H*	*35.057*
[24]	*2Em+7H*	*2Em+8H*	*5H*	*11.001*
Proposed	*1Em+5H*	*1Em+8H*	*2H*	*5.778*

5.2 *Communication and storage cost*

In this section, we define the number of bits used for identity, hash function, random number, ECC point, timestamp, and modular exponentiation as 160 bits, 160 bits, 160 bits, 320 bits, 64 bits, and 1024 bits, respectively, to calculate the number of bits used. The analysis of the

communication costs in Table 3 reveals that the proposed scheme offers a significant advantage over the other schemes. While the total communication costs for the schemes in [24],[25], [26], [27] *and* [28] *The proposed scheme, which ranges from 2560 to 3200 bits, significantly reduces this cost to 1480 bits, even using 14 messages. This reduction enhances efficiency and suggests a more optimized communication protocol, which can lead to lower latency and resource consumption in practical applications. Thus, the proposed scheme is a more efficient solution for secure communications.*

Table 3: Comparison of communication cost and number of bits.

Schemes	*Number of Bits*	*Number of Communication*
[25]	*3200 bits*	*4*
[26]	*3072 bits*	*3*
[27]	*2880 bits*	*4*
[28]	*2944 bits*	*4*
[24]	*2560 bits*	*4*
Proposed	*1480 bits*	*14*

5.3Security Requirement

As shown in Table 4, the proposed scheme surpasses other approaches in fulfilling security requirements. It offers comprehensive protection against various threats, including replay attacks, man-in-the-middle attacks, impersonation attacks, insider threats, and verification table leakage. Moreover, the scheme effectively incorporates critical features such as anonymity, untraceability, and perfect forward secrecy. Additionally, it provides notable advantages over competing schemes in terms of computational efficiency, secure offloading, and a lightweight design. SR1: Resistance to replay attack SR2: Resistance to man-in-the-middle attack SR3: Resistance to impersonation attack SR4: Resistance to privileged insider attack SR5: Resistance to verification table leakage attack SR6: Resistance to DoS attack SR7: Resistance to session-specific random number leakage attack SR8: Anonymity SR9: Untrace ability SR10: Perfect forward secrecy SR11: Mutual authentication SR12: Provide a computationally efficient communication, SR13:secure offloading, SR14 lightweight.

Table 4 comparison of security requirements.

Security Requirement	[25]	[26]	[27]	[28]	[24]	*Proposed*
SR1	*o*	*o*	*o*	*o*	*o*	*o*
SR2	*o*	*o*	*×*	*o*	*o*	*o*
SR3	*×*	*×*	*×*	*o*	*o*	*o*
SR4	*o*	*o*	*o*	*o*	*×*	*o*
SR5	*×*	*o*	*o*	*o*	*o*	*o*
SR6	*×*	*o*	*×*	*×*	*o*	*o*
SR7	*o*	*×*	*×*	*o*	*o*	*o*
SR8	*×*	*o*	*×*	*o*	*o*	*o*
SR9	*o*	*o*	*o*	*o*	*o*	*o*
SR10	*o*	*o*	*o*	*o*	*o*	*o*
SR11	*o*	*o*	*o*	*o*	*o*	*o*
SR12	*×*	*×*	*×*	*×*	*o*	*o*
SR13	*×*	*×*	*×*	*×*	*×*	*o*
SR14	*×*	*×*	*×*	*×*	*×*	*o*

O: supported, X: unsupported.

6. Simulation

This section presents the proposed scheme's simulation using the NS3 tool. It discusses the environmental parameters and then examines the simulation results in urban and highway scenarios. Finally, the findings are discussed.

6.1 Simulation parameter

This paper evaluated the proposed scheme's performance through simulations conducted in two distinct scenarios: an urban environment characterized by dense traffic and lower vehicle speeds and a highway setting representing a less congested, high-speed scenario. These scenarios were designed to test the scheme's adaptability and effectiveness under diverse network conditions. Platform and Hardware: The simulations were run on Ubuntu 20.04 using a Dell workstation equipped with an Intel Core i5 processor, 4 GB of RAM, and a 1 TB hard drive. The primary simulation tool was the NS3 simulation tool (Version 40) [40]*. Two scenarios were simulated: Urban (vehicles moving at 20 m/s with no pause) and Highway (vehicles moving at 100 m/s with no pause). The simulations involved 30, 50, 70, or 100 vehicles and 10 UAVs, with a communication range of 100 m for both UAV-to-vehicle and UAV-to-UAV interactions. Vehicles followed the Random Waypoint Mobility Model, while fog and cloud nodes were stationary. The simulation area was set to 300 m × 1500 m. The Friis Propagation Loss Model was used for signal attenuation, with a transmit power of 7.5 dB. The IEEE 802.11b wireless protocol and IEEE 802.11 DCF with CSMA/CA were employed for medium access control. The AODV routing protocol and UDP*

transmission protocol were utilized. Each simulation ran for 300 seconds. As shown in Table 5, we could simulate realistic urban and highway environments by configuring these parameters. This allows for a thorough assessment of the proposed scheme's scalability, reliability, and efficiency across varying vehicle densities and mobility patterns.

Table 5 shows parameter simulation.

Parameters	*Description*
Platform	*Ubuntu-24.04*
Hardware platform	*Dell 5110, Intel Core i5, 4 GB RAM, 1 TB HDD*
Tool used	*NS3. Ver 40*
Scenario types	*Urban, Highway*
Number of Vehicles	*30,50,70,100*
Number of UAVs	*10*
Vehicles speed Urban	*20 m/s (no pause)*
Vehicles speed Highway	*100 m/s (no pause)*
Mobility model	*Random Waypoint mobility model*
Mobility of Fog	*0*
Mobility of Cloud	*0*
Simulation environment area	*300*1500*
Loss model	*Friis propagation loss model*
Transmit power	*7.5 db*
Routing protocol	*AODV*
Medium access control type	*IEEE 802.11 DCF with mechanism CSAMA/CA*
Wireless protocol	*802.11 b*
Transmit protocol	*UDP*
Communication range of UAV to Vehicles	*100 m*
Communication range of UAV to UAV	*100 m*
Simulation time	*300 s*

6.2 Simulation result

Urban Simulation results: In this section, the analysis of network performance metrics is presented in vehicle traffic simulation scenarios with a constant speed of 100 km/h and varying numbers of vehicles: 30, 50, 70, and 100. The evaluation metrics include Throughput, End-to-End Delay, Packet Delivery Ratio (PDR), and Packet Loss Ratio. The data is examined at two levels: Aggregate flow Monitor Metrics and the Distribution of metrics for each flow Aggregate.

6.3Urban Simulation Results

This section presents the analysis of network performance metrics in vehicle traffic simulation scenarios with a constant speed of 100 km/h and varying numbers of vehicles: 30, 50, 70, and 100. The evaluation metrics include Throughput, End-to-End Delay, Packet Delivery Ratio

(PDR), and Packet Loss Ratio. The data is examined at two levels: Aggregate FlowMonitor Metrics and the Distribution of metrics for each flow.

6.3.1 FlowMonitor Metrics Analysis

Figures 10 to 13 show the distribution of performance metrics for each flow in scenarios with 30, 50, 70, and 100 vehicles, with the number of flows (Flow IDs) being 175, 500, 1250, and 1750, respectively.

Throughput Distribution: *30-vehicle Scenario: In Figure 10, Most flows have a throughput below 200 kbps, with a maximum of 600 kbps. The dispersion is low, but a few flows reach higher throughputs.50-vehicle Scenario: In Figure 11, Throughput for most flows remains below 200 kbps, with a maximum of 1200 kbps. Dispersion increases slightly compared to the 30-vehicle scenario, with more flows achieving higher throughputs.70-vehicle Scenario: In Figure 12, Most flows still have throughput below 200 kbps, with a maximum of 1000 kbps. Dispersion is similar to the 50-vehicle scenario, but increased flows (1250) lead to a broader distribution.100-vehicle Scenario: In Figure 13, Throughput for most flows is below 200 kbps, with a maximum of 1400 kbps. Dispersion is higher than in previous scenarios, and more flows achieve throughputs above 1000 kbps, though they remain a minority.*

End-to-End Delay Distribution: *30-vehicle Scenario: In Figure 10, Most flows have delays below 500 ms, with a maximum of 2000 ms. The dispersion is notable, reflecting varied communication conditions. 50-vehicle Scenario: In Figure 11, Delays for most flows are below 500 ms, but the maximum reaches 3000 ms. Dispersion increases, likely due to greater network congestion. 70-vehicle Scenario: In Figure 12, Most flows have delays below 500 ms, with a maximum of 1750 ms. Dispersion is similar to the 50-vehicle scenario, but the larger number of flows results in a broader distribution. 100-vehicle Scenario: In Figure 13, Delays for most flows are below 500 ms, with a maximum of 4000 ms. Dispersion is the highest, indicating significant congestion impacts at high density.*

Packet Delivery Ratio (PDR) Distribution: 30-vehicle Scenario: *In Figure 10, PDR for most flows ranges between 60% and 100%, consistent with the overall average (87.80%). Some flows have low PDR (below 40%), indicating delivery issues.* ***50-vehicle Scenario****: In Figure 11, PDR for most flows is between 60% and 100%, averaging 85.94%. The number of flows with low PDR (below 40%) increases slightly.* ***70-vehicle Scenario:*** *In Figure 12, the PDR for most flows remains between 60% and 100%, with an average of 84.91%. Compared to previous scenarios, more flows have low PDR (below 40%).* ***100-vehicle Scenario****: In Figure 13, PDR for most flows is between 60% and 100%, with an average of 89.59%. The number of flows with low PDR (below 40%) decreases, suggesting improved packet delivery.*

Packet Loss Ratio Distribution: 30-vehicle Scenario*: In Figure 10, Packet loss for most flows is between 0% and 40%, consistent with the average (12.20%). Some flows experience high loss (up to 100%), possibly due to connection drops.* ***50-vehicle Scenario****: In Figure 11, Packet loss for most flows is between 0% and 40%, with an average of 14.06%. The number of flows with high loss (up to 100%) increases slightly.* ***70-vehicle Scenario:*** *In Figure 12, packet loss for most flows is between 0% and 40%, with an average of 15.09%. Compared to previous scenarios, more flows have high loss rates.****100-vehicle Scenario:*** *In Figure 13, Packet loss for most flows is between 0% and 40%, with an average of 10.41%. The number of flows with high loss decreases, indicating a reduction in packet loss.*

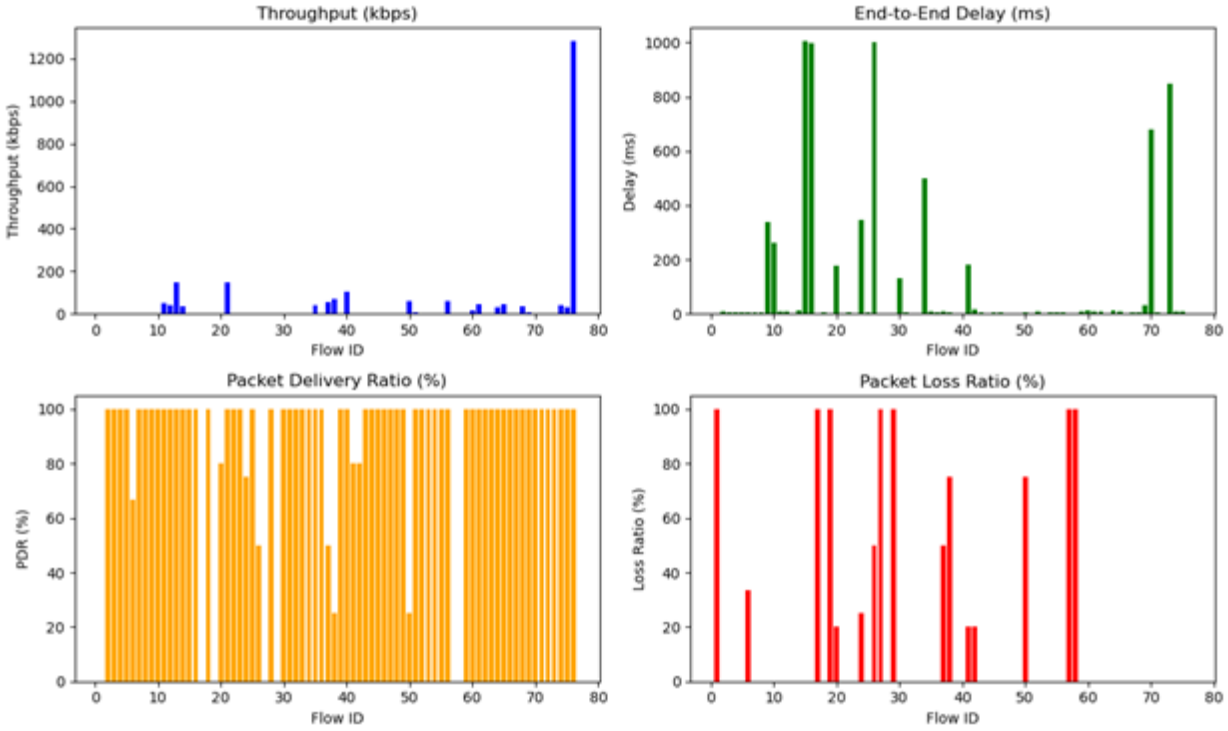

Fig10. Performance metrics under a scenario with 30 vehicles.

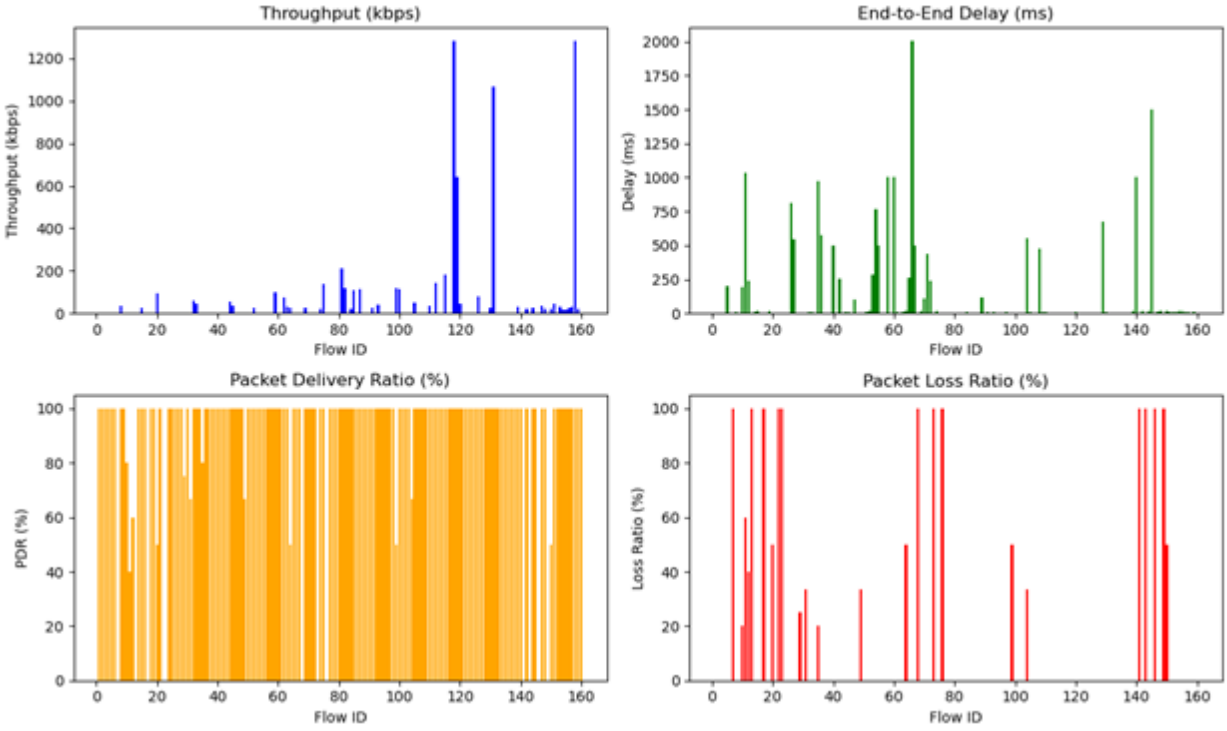

Fig11. Performance metrics under a scenario with 50 vehicles.

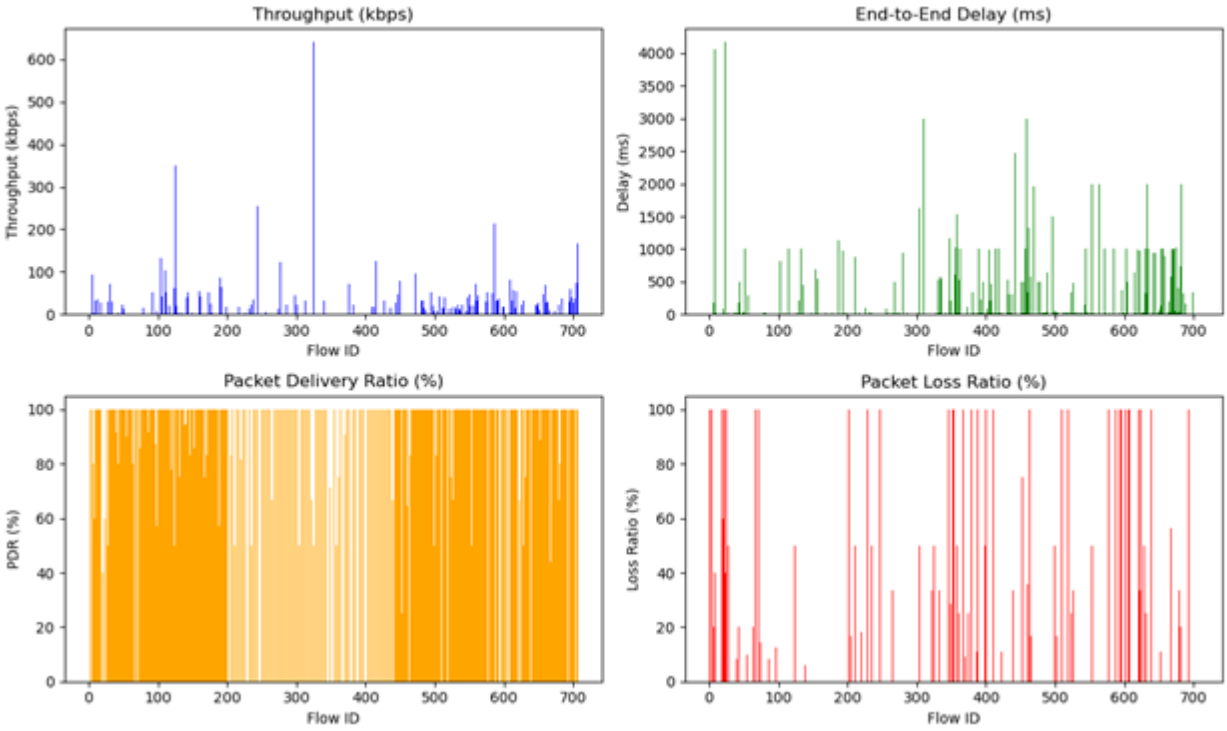

Fig12. Performance metrics under a scenario with 70 vehicles.

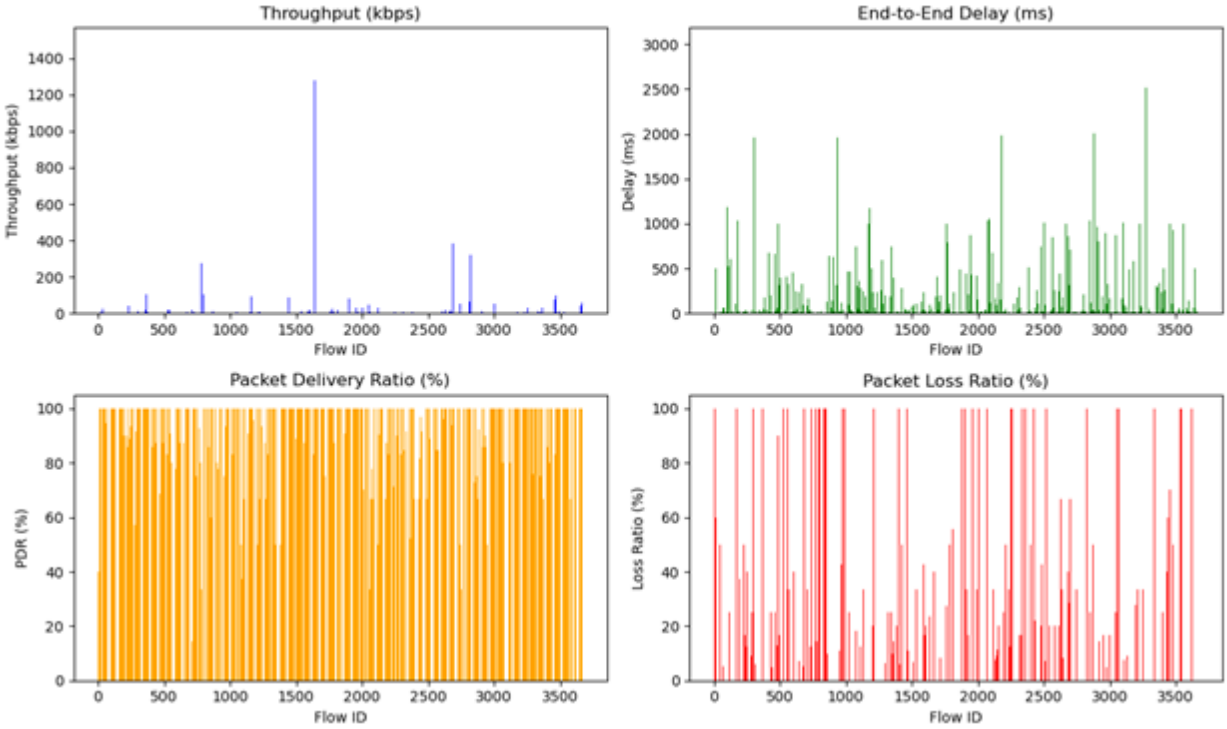

Fig13. Performance metrics under a scenario with 100 vehicles.

6.3.2 Aggregate FlowMonitor Metrics Analysis

Figures 14 to 17 display the aggregate flow metrics for the four scenarios with 30, 50, 70, and 100 vehicle counts.

Observations: *Throughput: Throughput increases from 6885.41 kbps (30 vehicles) to 23959.31 kbps (50 vehicles), a 3.5-fold increase, indicating better network utilization. However, it decreases to 9487.02 kbps (70 vehicles) and 4654.47 kbps (100 vehicles), likely due to network congestion and increased competition for channel access. End-to-End Delay (Avg Delay): The delay peaks at 202.41 ms in the 50-vehicle scenario, likely due to higher traffic. It decreases to 197.45 ms (70 vehicles) and 109.39 ms (100 vehicles), possibly due to reduced packet delivery under congestion. Packet Delivery Rate (Avg PDR): PDR is highest in the 100-vehicle scenario (89.59%) and lowest in the 70-vehicle scenario (84.91%). The variation (84.91% to 89.59%) suggests relative stability in packet delivery across scenarios. Packet Loss Ratio (Avg Loss Ratio): Packet loss is lowest in the 100-vehicle scenario (10.41%) and highest in the 70-vehicle scenario (15.09%). The variation (10.41% to 15.09%) reflects an inverse relationship with PDR.*

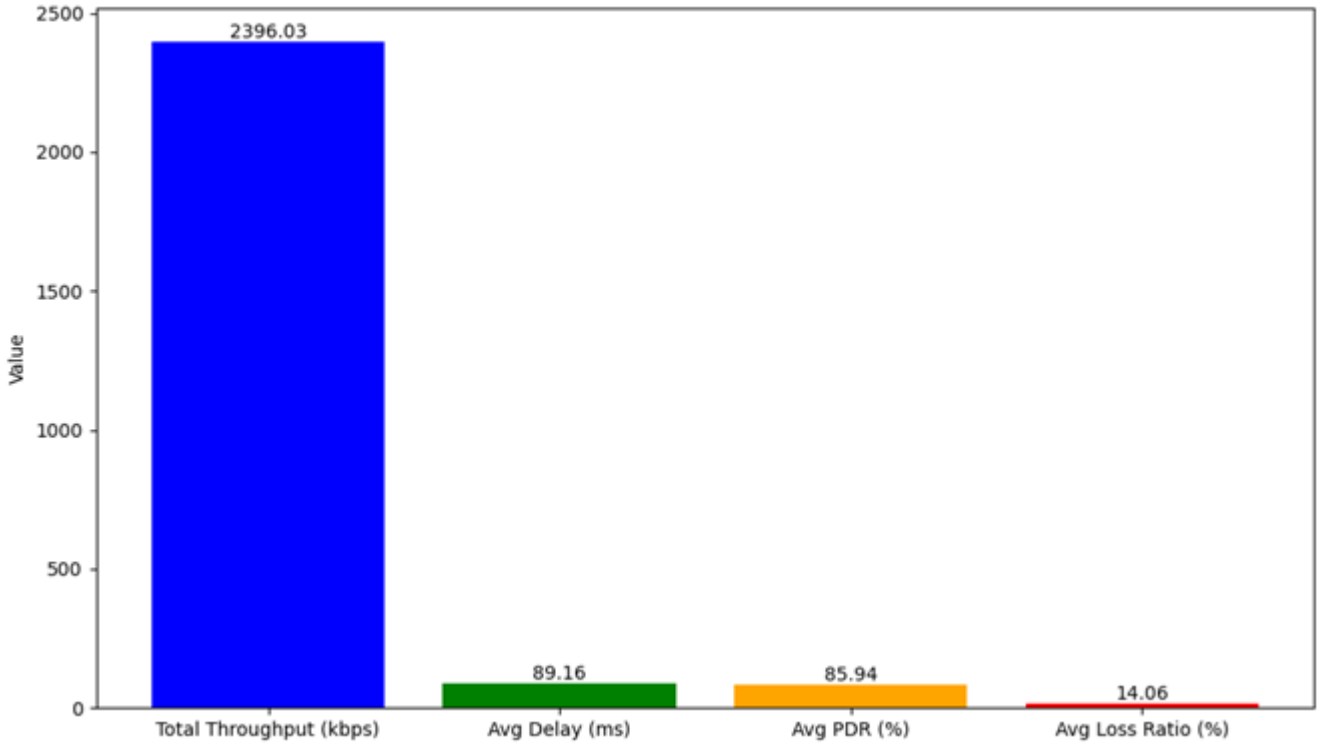

Fig14. Aggregate FlowMonitor metrics for the scenario with 30 vehicles.

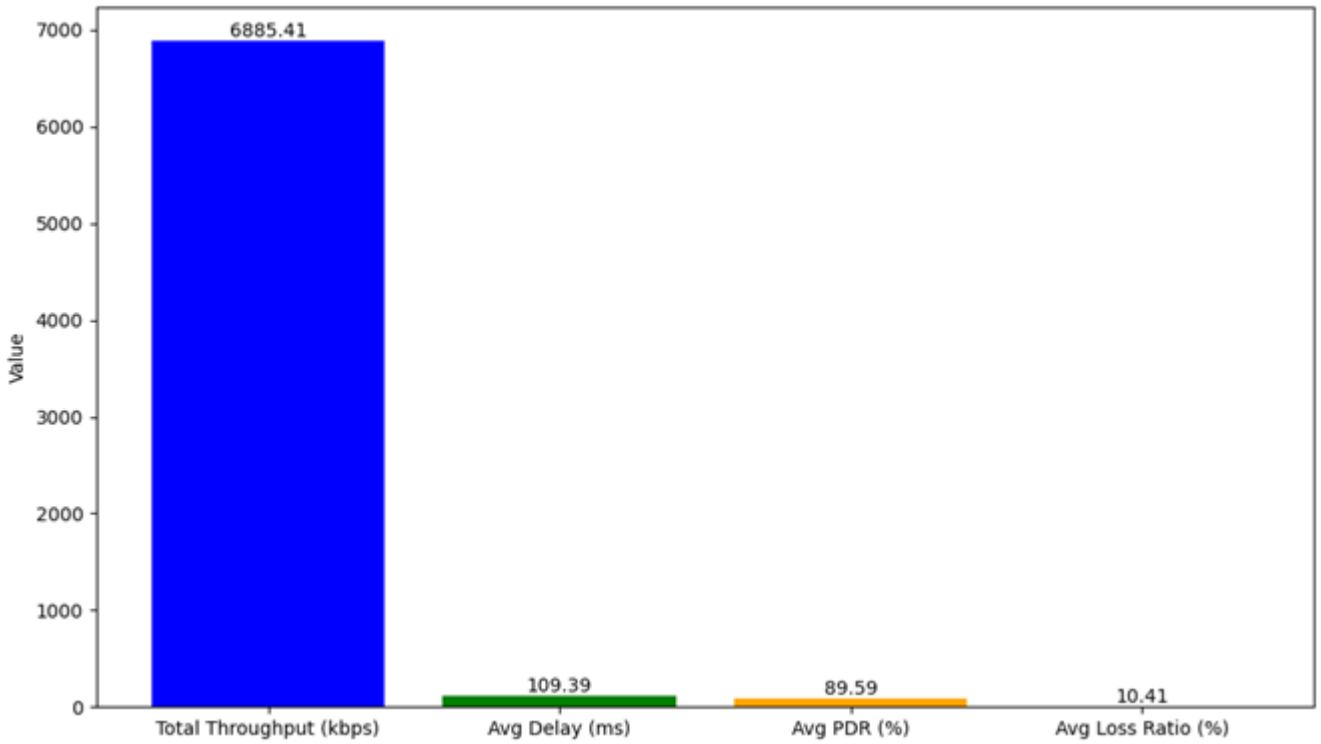

Fig 15. Aggregate FlowMonitor metrics for the scenario with 50 vehicles.

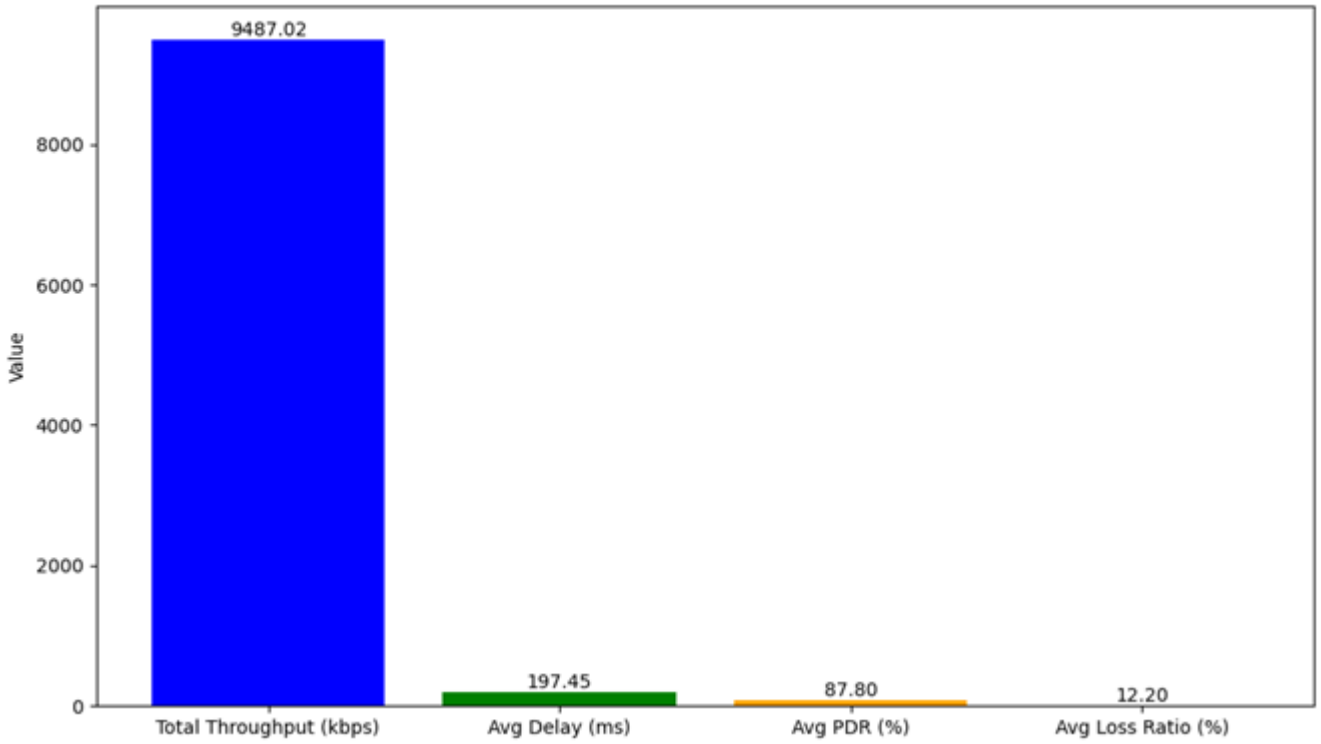

Fig16.Aggregate FlowMonitor metrics for the scenario with 70 vehicles.

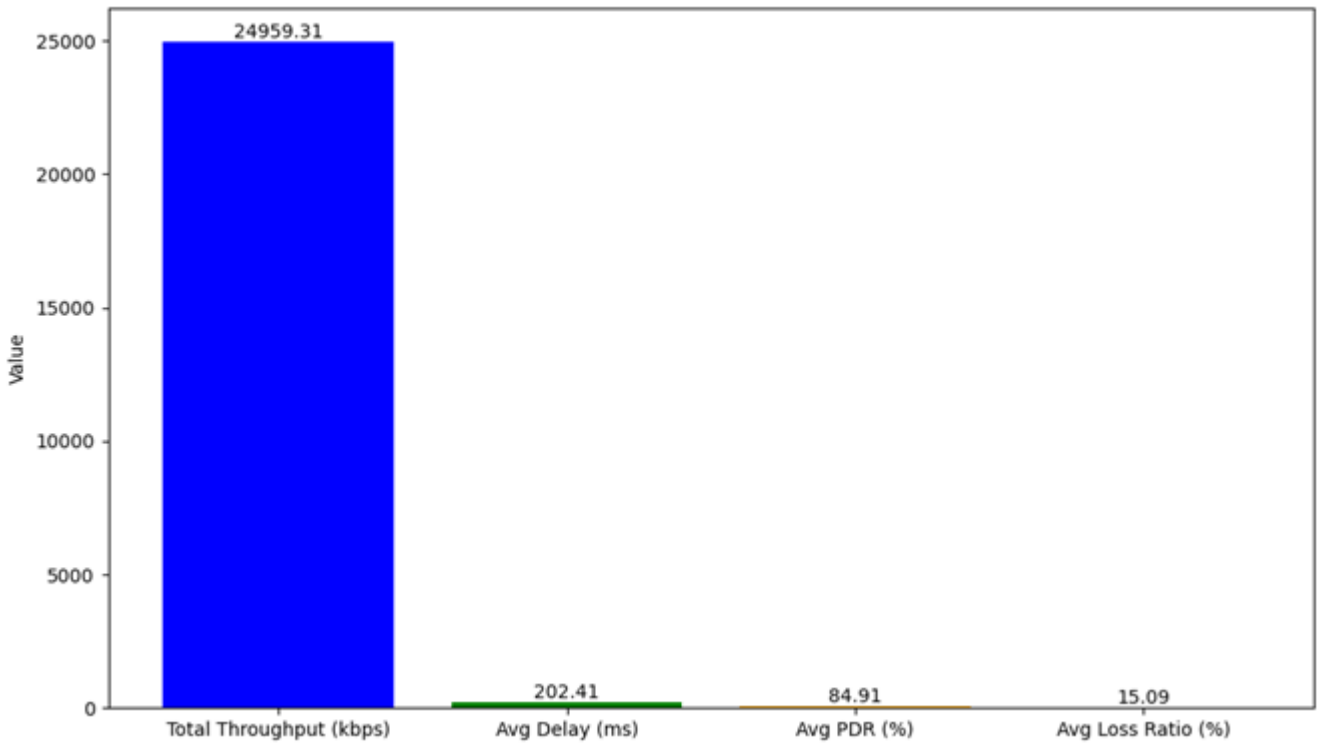

Fig 17. Aggregate FlowMonitor metrics for the scenario with 100 vehicles.

6.3.3 Comprehensive Analysis and Interpretation of Results

Effect of Number of Vehicles on the Distribution of Metrics: *Throughput: As vehicles increase from 30 to 100, the maximum throughput rises from 600 to 1400 kbps, indicating better performance for some flows at higher densities. However, most flows remain below 200 kbps, with increasing dispersion reflecting non-uniform network performance. End-to-End Delay: Maximum delay increases from 2000 ms (30 vehicles) to 4000 ms (100 vehicles), showing the impact of congestion. Most flows have delays below 500 ms, but dispersion grows with more vehicles, highlighting varying communication conditions. Packet Delivery Rate (PDR): PDR for most flows remains between 60% and 100%, consistent with averages (84.91% to 89.59%). Low PDR flows peak in the 70-vehicle scenario but decrease in the 100-vehicle scenario, indicating improved delivery at higher density. Packet Loss Ratio: Packet loss for most flows is between 0% and 40%, consistent with averages (10.41% to 15.09%). High-loss flows peak in the 70-vehicle scenario but decrease in the 100-vehicle scenario, aligning with PDR improvements.*

Correlation Between Metrics: *Throughput and Delay: Flows with high throughput (e.g., above 1000 kbps) often have higher delays (up to 4000 ms in the 100-vehicle scenario), suggesting a trade-off between throughput and latency. Delivery Rate and Packet Loss: There is an inverse relationship between PDR and Loss Ratio. Flows with high PDR (close to 100%) typically have low Loss Ratios (close to 0%). Dispersion: Throughput and delay show high dispersion across all scenarios, while PDR and Loss Ratio are more evenly distributed, except for poorly performing flows.*

6.4 Highway Simulation Results

In this section, the analysis of network performance metrics is presented in vehicle traffic simulation scenarios with a constant speed of 100 km/h and varying numbers of vehicles: 30, 50, 70, and 100. The evaluation metrics include Throughput, End-to-End Delay, Packet Delivery Ratio (PDR), and Packet Loss Ratio. The data is examined at two levels: Aggregate flow Monitor Metrics and the Distribution of metrics for each flow Aggregate.

6.4.1 FlowMonitor Metrics Analysis

Figures 18 to 21 show the distribution of performance metrics for each flow in scenarios with 30, 50, 70, and 100 vehicles. The number of flows (Flow IDs) in these scenarios is 175, 500, 1250, and 1750, respectively.

***Throughput Distribution**: 30-vehicle Scenario: In Figure 18, the throughput of most flows is below 200 kbps, with a maximum value of about 1000 kbps. The dispersion of throughput is relatively low, but some flows have higher throughputs (up to 1000 kbps).50-vehicle Scenario: Figure 19 The throughput of most flows is below 200 kbps, but the maximum value reaches about 1200 kbps. The dispersion is slightly higher than in the 30-vehicle scenario, with many flows having high throughput rates (up to 1200 kbps).70-vehicle scenario: Figure 20 shows that the throughput rates of most flows are still below 200 kbps, but the maximum value reaches around 1200 kbps. The dispersion is similar to the 50-vehicle scenario, but the number of flows has increased (1250), indicating a wider distribution. 100-vehicle scenario: In Figure 21, the throughput rates of most flows are below 200 kbps, with a maximum value of around 1400 kbps. The dispersion is higher than in the previous scenarios, and the number of flows with high throughput rates (over 1000 kbps) has increased, but still, a small number of flows have such values.*

***End-to-End Delay Distribution:** 30-vehicle scenario: In Figure 18, Most flow delays are below 500 ms, but some have high delays (up to 2500 ms). The delay dispersion is significant, indicating the difference in the communication conditions of the flows.50-vehicle scenario: In Figure 19, the delay of most flows is below 500 ms, but the maximum delay reaches about 3000 ms. The delay dispersion is higher than that of the 30-vehicle scenario, possibly due to higher congestion.70-vehicle scenario: In Figure 20, the delay of most flows is below 500 ms, but the maximum delay reaches about 3000 ms. The delay dispersion is similar to that of the 50-vehicle scenario, but a wider distribution is observed with larger flows.100-vehicle scenario: In Figure 21, the latency of most flows is below 500 ms, but the maximum latency reaches about 4000 ms. The latency dispersion is larger than in the previous scenarios, indicating the impact of network congestion at high density.*

Packet Delivery Ratio (PDR) distribution:

30-vehicle scenario: In Figure 18, the packet delivery ratio for most flows is between 60% and 100%, consistent with the overall average (80.46%). Some flows have low delivery ratios

(below 40%), indicating packet delivery problems in these flows.50-vehicle scenario: In Figure 19, the packet delivery ratio for most flows is between 60% and 100%, consistent with the overall average (77.19%). The number of flows with low PDR (below 40%) is slightly higher than in the 30-vehicle scenario. 70-vehicle scenario: In Figure 20, the packet delivery rate for most flows is between 60% and 100%, consistent with the average (76.62%). More flows have low delivery rates (below 40%) than in the previous scenarios. 100-vehicle scenario: In Figure 21, the packet delivery rate for most flows is between 60% and 100%, consistent with the overall average (81.77%). The number of flows with low PDR (below 40%) is lower than in the 70-vehicle scenario, indicating a relative improvement in packet delivery.

Packet Loss Ratio Distribution: *30-vehicle scenario: In Figure 18, the packet loss rate for most flows is between 0% and 40%, consistent with the overall average (19.54%). Some flows have high loss rates (up to 100%), which connection drops can cause. 50-vehicle scenario: In Figure 19, the packet loss rate for most flows is between 0% and 40%, consistent with the overall average (22.81%). The number of flows with high loss (up to 100%) is slightly higher than in the 30-vehicle scenario. 70-vehicle scenario: In Figure 20, the packet loss rate for most flows is between 0% and 40%, consistent with the overall average (23.38%). More flows have high loss rates (up to 100%) than in the previous scenarios. 100-vehicle scenario: In Figure 21, the packet loss rate for most flows is between 0% and 40%, consistent with the overall average (18.23%). The number of flows with high loss (up to 100%) is lower than in the 70-vehicle scenario, indicating a relative reduction in the packet loss rate.*

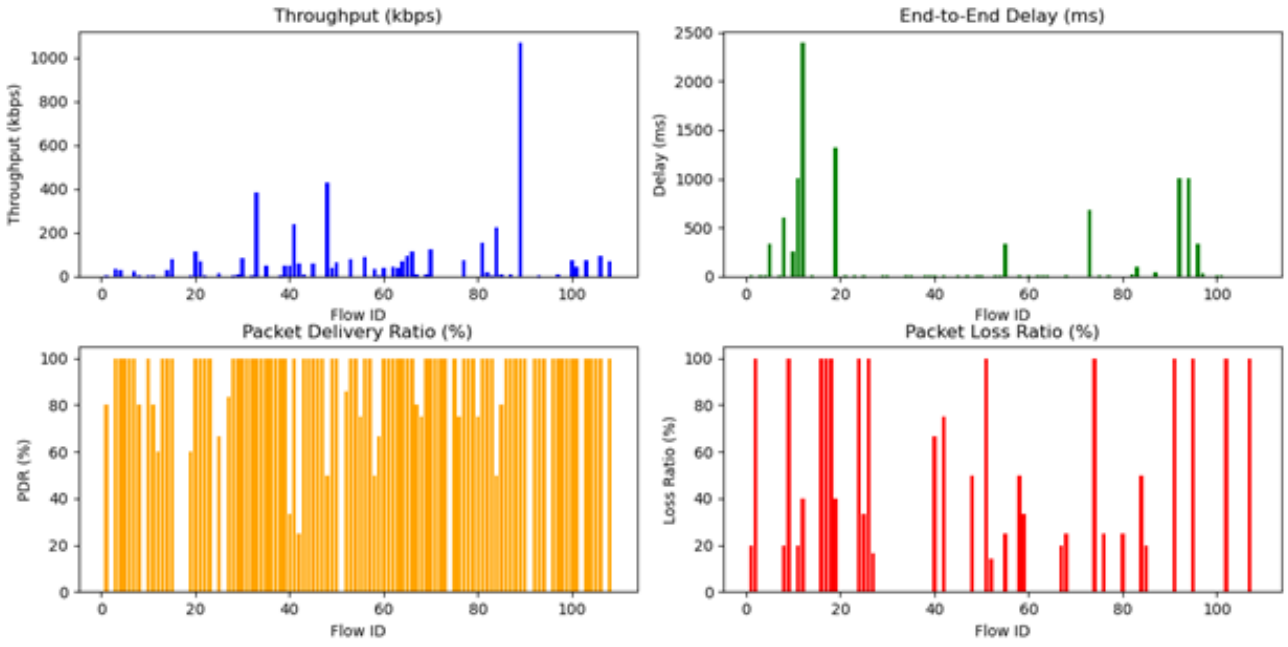

Fig18. Performance metrics under a scenario with 30 vehicles

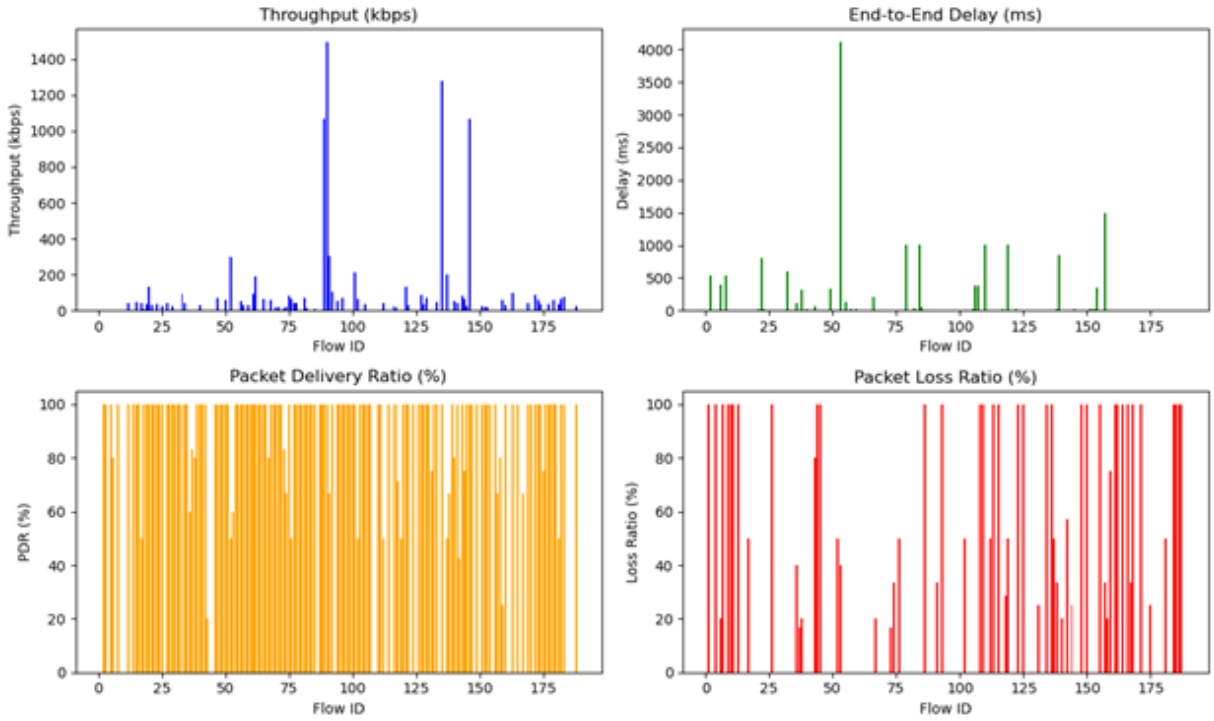

Fig19. Performance metrics under a scenario with 50 vehicles

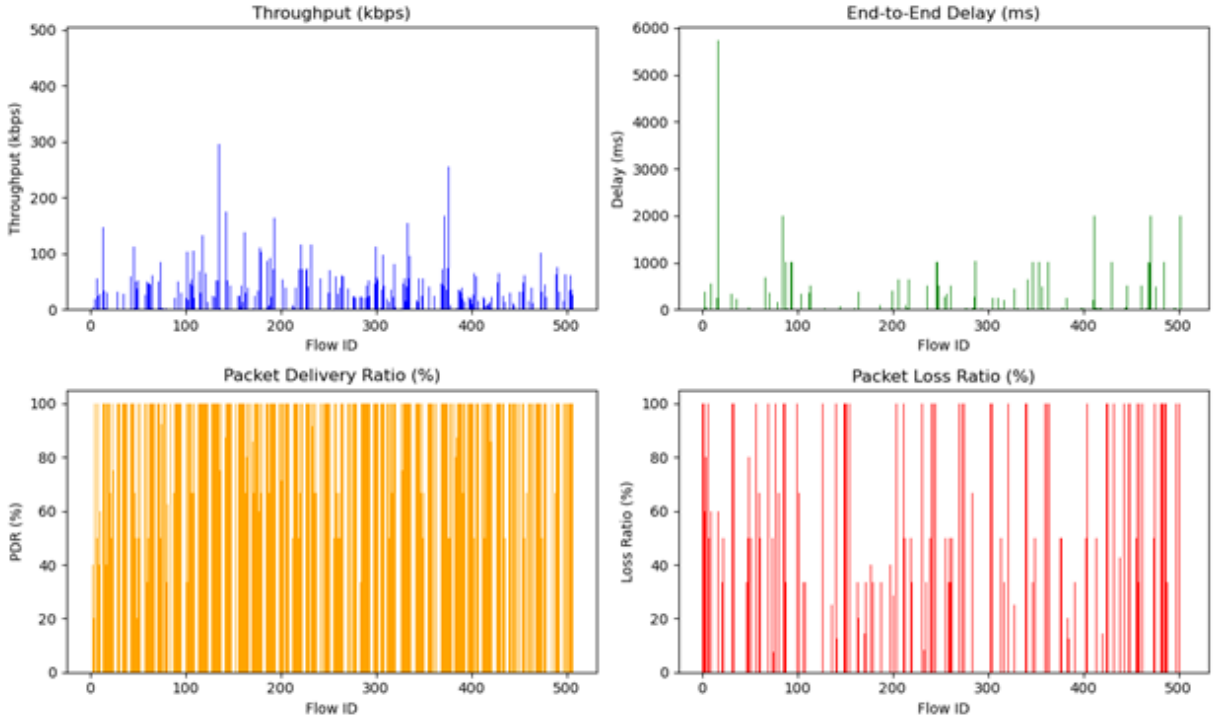

Fig20. Performance metrics under a scenario with 70 vehicles

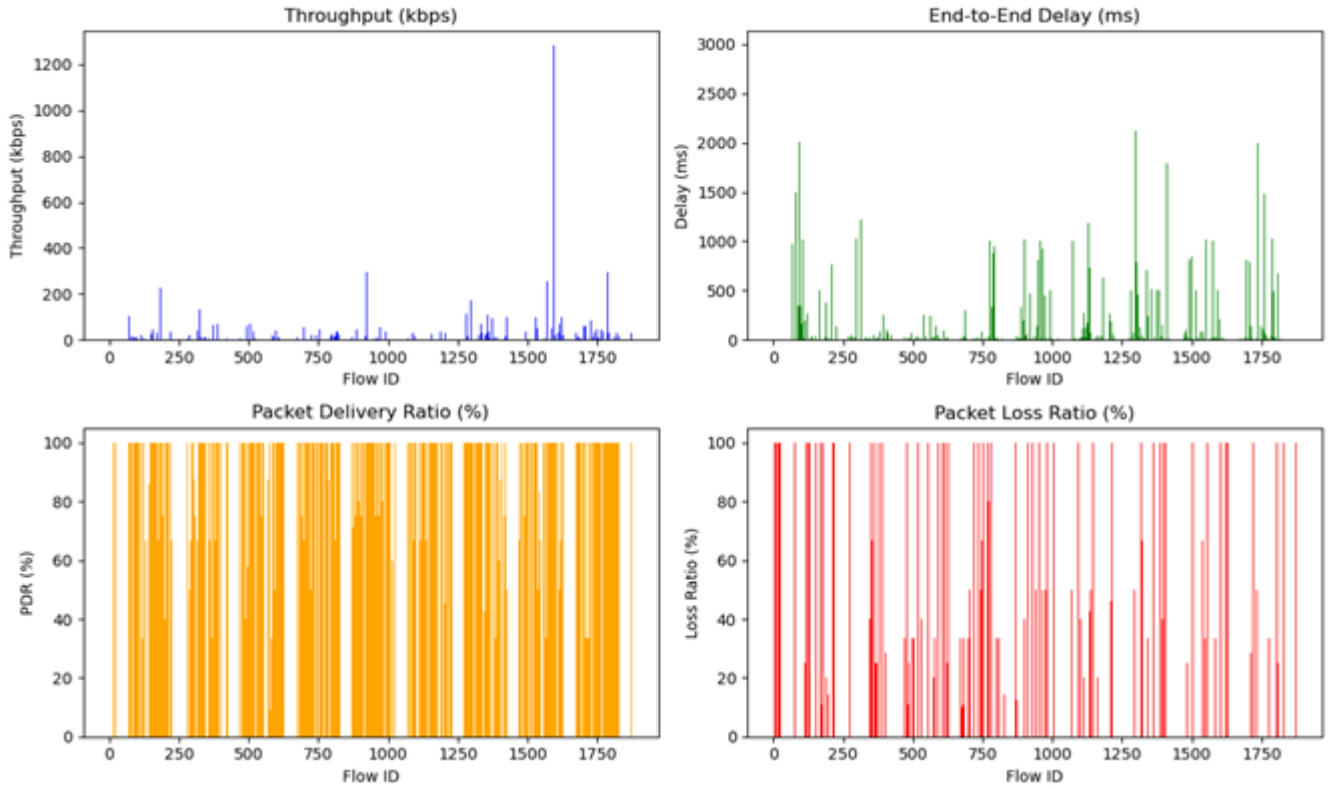

Fig21. Performance metrics under a scenario with 100 vehicles

6.4.2 Aggregate FlowMonitor Metrics Analysis

Figures 22 to 25 show the aggregate flow metrics for four scenarios with vehicle counts of 30, 50, 70, and 100

***Observations**: Throughput: The throughput increases from 12308.30 kbps to 29697.11 kbps as the number of vehicles increases from 30 to 50 (about 2.4 times). This increase indicates better network capacity utilization with the rise in the number of flows. With the number of vehicles increasing to 70 and 100, the throughput decreases to 9442.52 kbps and 4654.47 kbps, respectively. This sharp decrease can be attributed to network congestion and more competition for access to the communication channel. End-to-end Delay (Avg Delay): The delay in the 50-vehicle scenario is the highest (147.01 ms), likely due to higher traffic and increased waiting time for sending packets. As the number of vehicles increases to 70 and 100, the delay decreases to 87.12 ms and 90.96 ms, respectively. This decrease may be due to the decrease in packet delivery rate under congestion conditions. Packet Delivery Rate (Avg PDR): The packet delivery rate is highest in the 100-vehicle scenario (81.77%), while the lowest value is observed in the 70-vehicle scenario (76.62%). The variation in PDR between scenarios is relatively small (between 76.62% and 81.77%), indicating relative stability in packet delivery despite changes in the number of vehicles. Packet Loss Ratio (Avg Loss Ratio): The packet loss rate is lowest in the 100-vehicle scenario (18.23%), and highest in the 70-vehicle scenario (23.38%). Similar to PDR, the variation in packet loss rate between scenarios is limited (between 18.23% and 23.38%), indicating an inverse relationship with PDR.*

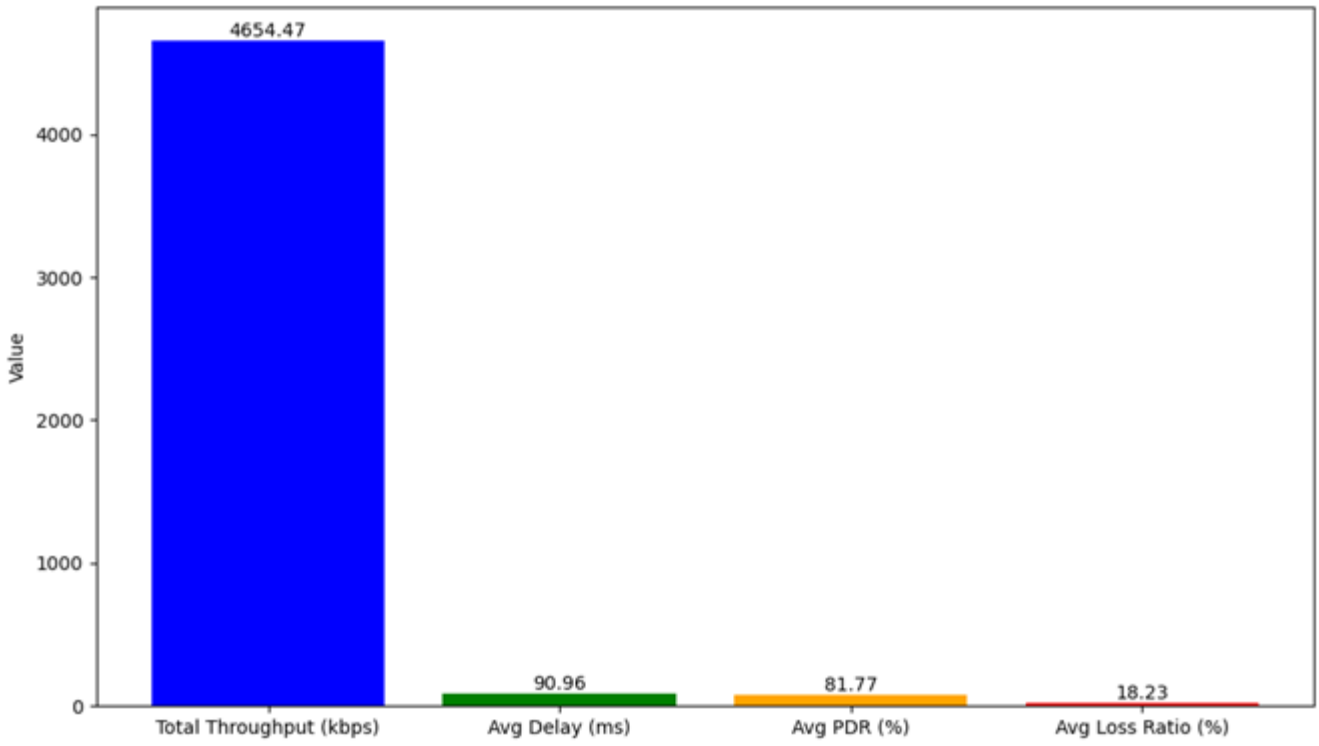

Fig22. Aggregate FlowMonitor metrics for the scenario with 30 vehicles.

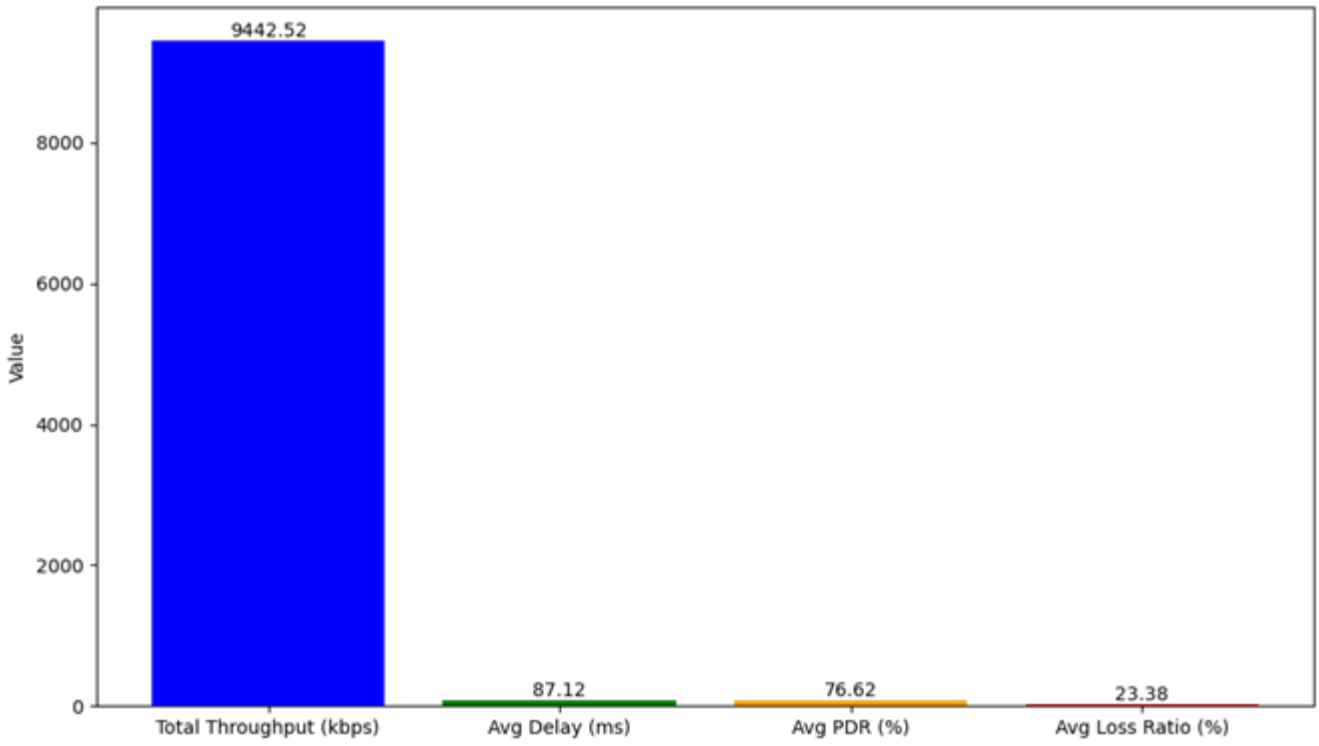

Fig23. Aggregate FlowMonitor metrics for the scenario with 50 vehicles.

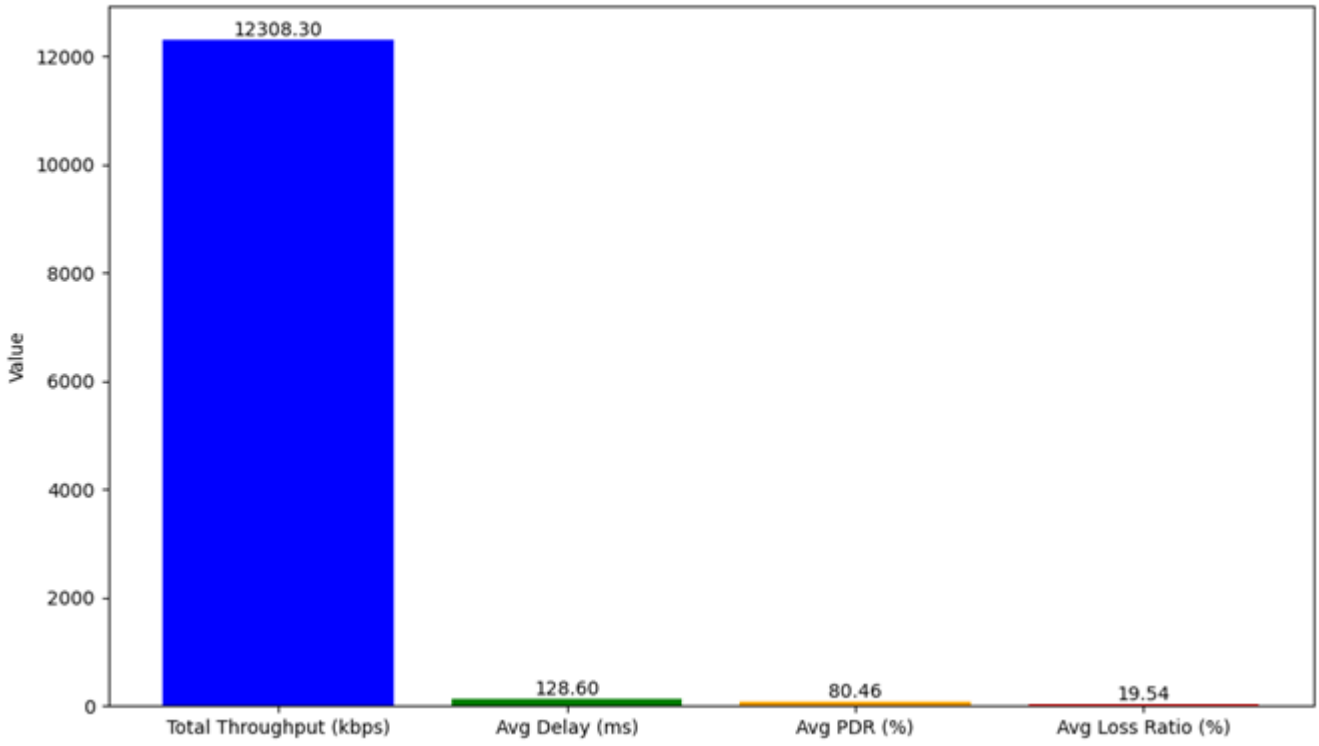

Fig24. Aggregate FlowMonitor metrics for the scenario with 70 vehicles.

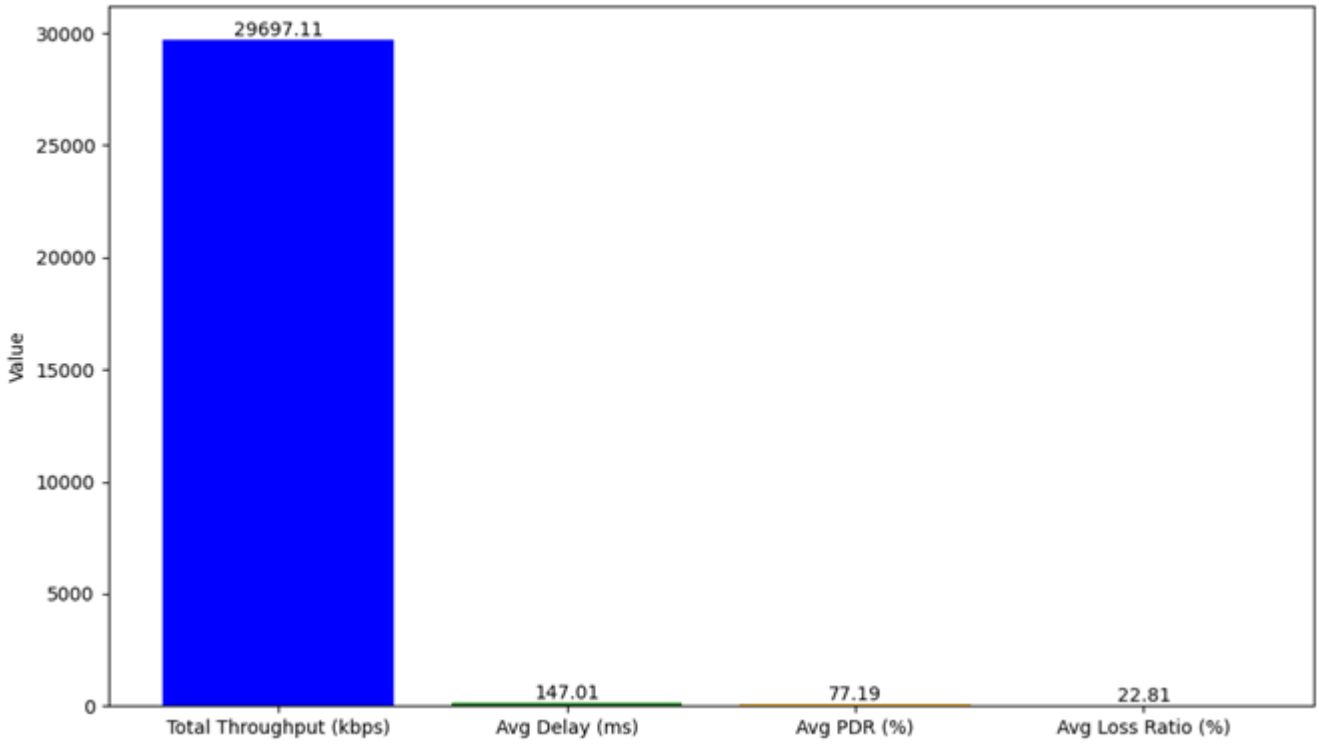

Fig25. Aggregate FlowMonitor metrics for the scenario with 100 vehicles.

6.4.3 Comprehensive analysis and interpretation of results

Effect of the number of vehicles on the distribution of metrics*: Throughput: As the number of vehicles increases from 30 to 100, the maximum throughput rate increases from 1000 kbps to 1400 kbps. This indicates that there are flows with better performance at higher densities. However, most flows in all scenarios have low throughput rates (below 200 kbps), and the dispersion increases with the number of vehicles. This dispersion indicates non-uniformity in network performance. End-to-End Delay: The maximum delay increases with the number of vehicles from 2500 ms (30 vehicles) to 4000 ms (100 vehicles). This increase indicates the effect of network congestion at higher densities. In all scenarios, most flows have delays below 500 ms, but the dispersion of delays increases with increasing number of vehicles, indicating greater differences in the communication conditions of the flows. Packet Delivery Rate (PDR): The packet delivery rate in all scenarios for most flows is between 60% and 100%, consistent with the overall averages (76.62% to 81.77%). The number of flows with low PDR (below 40%) is higher in the 70-vehicle scenario, but decreases in the 100-vehicle scenario, indicating a relative improvement in packet delivery at high density. Packet Loss Ratio: The packet loss rate in all scenarios for most flows is between 0% and 40%, consistent with the overall averages (18.23% to 23.38%). The number of flows with high loss (up to 100%) is higher in the 70-vehicle scenario but decreases in the 100-vehicle scenario, which is consistent with the improvement in PDR.*

Correlation between metrics: *Throughput and delay: Flows with high throughput (e.g., over 1000 kbps) typically have higher latency (up to 4000 ms in the 100-vehicle scenario). This suggests that better throughput performance may be associated with higher latency. Delivery rate and packet loss: An inverse relationship between PDR and Loss Ratio is observed in all scenarios. Flows with high PDR (close to 100%) typically have low Loss Ratios (close to 0%). Dispersion: The dispersion in throughput and delay is high in all scenarios, but PDR and Loss Ratio are more evenly distributed, except for the poorly performing flows.*

6.5Discussion

Comparing the highway and urban simulation results, the proposed scheme performs best in the urban scenario with 100 vehicles, where it achieves the highest average Packet Delivery Ratio (PDR) of 89.59% and the lowest average Packet Loss Ratio of 10.41%, indicating robust packet delivery and minimal losses despite high vehicle density. The throughput in this scenario is 4654.47 kbps, with a maximum of 1400 kbps, and the average end-to-end delay is the lowest at 109.39 ms, reflecting efficient communication under dense conditions. In the

highway scenario, the best performance is observed with 50 vehicles, where the throughput peaks at 29697.11 kbps, significantly higher than other highway scenarios, and the PDR is 77.19% with a packet loss ratio of 22.81%. However, the highway 50-vehicle scenario has a higher average delay of 147.01 ms than the urban 100-vehicle scenario. In contrast, the highway 100-vehicle scenario shows a lower PDR (81.77%) and higher packet loss (18.23%) with a maximum delay of 4000 ms, indicating poorer performance under high density. For lower vehicle counts (30 and 70), the urban scenario consistently outperforms the highway in PDR and packet loss, with the urban 50-vehicle scenario also showing a high throughput of 23959.31 kbps. The urban 100-vehicle scenario demonstrates the best overall performance with high PDR, low packet loss, and minimal delay, improving efficiency as vehicle numbers increase. In contrast, the highway 50-vehicle scenario excels in throughput but is less efficient in delay and packet delivery.

7 Conclusion

In light of the inherent vulnerabilities and exposure to various attacks in edge-layer communication channels, this paper proposes a lightweight authentication and key exchange scheme that leverages Elliptic Curve Cryptography to enable secure data offloading in IoV environments. The security evaluation of the proposed scheme against active and passive attacks has been conducted using both formal and informal methods, with results demonstrating its strong resilience. We also examined the proposed scheme from various perspectives, including computational cost, communication cost, the number of bits used, and security requirements. The results indicate that the costs associated with the proposed scheme are lower than those of others. Simulations were conducted using the NS3 tool in urban and highway scenarios, with a variable number of vehicles. These simulations analyzed end-to-end delay, packet loss, delivery, and throughput. The results show that the proposed scheme performs well in both scenarios. Notably, the overall performance in the urban scenario with 100 vehicles is optimal, exhibiting a high packet delivery ratio (PDR), low packet loss, and minimal delay. Additionally, the scheme's efficiency improves as the number of vehicles increases. In the highway scenario, 50 vehicles demonstrate superior throughput. In the future, we plan to implement the proposed scheme on a satellite platform, taking into account the unique challenges of space-based environments, such as limited bandwidth, high latency, and dynamic link reliability.

Data availability

The data used to support this proposed scheme are provided within the article.

References

[1] P. Sun, Y. Wan, Z. Wu, Z. Fang, and Q. Li, "A survey on privacy and security issues in IoT-based environments: Technologies, protection measures and future directions," *Comput. Secur.*, vol. 148, p. 104097, 2025.

[2] K. Ala'F, A. Alqammaz, A. M. Khasawneh, L. Abualigah, K. A. Darabkh, and Z. Zinonos, "An environmental remote sensing and prediction model for an IoT smart irrigation system based on an enhanced wind-driven optimization algorithm," *Comput. Electr. Eng.*, vol. 122, p. 109889, 2025.

[3] J. M. Kizza, "Internet of things (iot): growth, challenges, and security," in *Guide to Computer Network Security*, Springer, 2024, pp. 557–573.

[4] N. Sharma and P. Dhiman, "A survey on IoT security: challenges and their solutions using machine learning and blockchain technology," *Cluster Comput.*, vol. 28, no. 5, pp. 1–40, 2025.

[5] Y. Salami, F. Taherkhani, Y. Ebazadeh, M. Nemati, V. Khajehvand, and E. Zeinali, "Blockchain-Based Internet of Vehicles in Green Smart City: Applications and Challenges and Solutions," *Anthropog. Pollut.*, vol. 7, no. 1, pp. 87–96, 2023, doi: 10.22034/AP.2023.1978624.1144.

[6] A. M. S. Saleh, "Blockchain for secure and decentralized artificial intelligence in cybersecurity: A comprehensive review," *Blockchain Res. Appl.*, p. 100193, 2024.

[7] S. B. Far and M. R. Asaar, "A blockchain-based anonymous reporting system with no central authority: Architecture and protocol," *Cyber Secur. Appl.*, vol. 2, p. 100032, 2024.

[8] Y. Salami and S. Hosseini, "BSAMS: Blockchain-Based Secure Authentication Scheme in Meteorological Systems," *Nivar*, vol. 47, no. 120–121, pp. 181–197, 2023, doi: https://doi.org/10.30467/nivar.2023.415722.1260.

[9] Y. Liu *et al.*, "An overview of blockchain smart contract execution mechanism," *J. Ind. Inf. Integr.*, p. 100674, 2024.

[10] Y. Salami, V. Khajehvand, and E. Zeinali, "Efficiency Simultaneous key Exchange-Cryptography Extraction from Public key in Fog-Cloud Federation-based Secure Offloading for Automatic Weather Stations Observing Systems," *Nivar*, vol. 47, no. 120–121, pp. 153–165, 2023, doi: 10.30467/nivar.2023.416270.1261.

[11] Y. Salami, Y. Ebazadeh, and V. Khajehvand, "CE-SKE: cost-effective secure key exchange scheme in Fog Federation," *Iran J. Comput. Sci.*, vol. 4, no. 3, pp. 1–13, 2021.

[12] Y. Salami and V. Khajehvand, "LSKE: Lightweight Secure Key Exchange Scheme in Fog Federation," *Complexity*, vol. 2021, p. 4667586, 2021, doi: https://doi.org/10.1155/2021/4667586.

[13] Y. Salami, V. Khajehvand, and E. Zeinali, "SAIFC: A Secure Authentication Scheme for IOV Based on Fog-Cloud Federation," *Secur. Commun. Networks*, vol. 1, pp. 1–19, 2023, doi: https://doi.org/10.1155/2023/9143563.

[14] Y. Salami, "SO-ITS: a secure offloading scheme for intelligent transportation systems in federated fog-cloud," *Iran J. Comput. Sci.*, 2025, doi: 10.1007/s42044-025-00318-9.

[15] M. Bakirci, "Evaluating the impact of unmanned aerial vehicles (UAVs) on air quality management in smart cities: A comprehensive analysis of transportation-related pollution," *Comput. Electr. Eng.*, vol. 119, p. 109556, 2024.

[16] S. M. Bhat and A. Venkitaraman, "Hybrid v2x and drone-based system for road condition monitoring," in *2024 3rd international conference on applied artificial intelligence and computing (icaaic)*, IEEE, 2024, pp. 1047–1052.

[17] Z. Shuai *et al.*, "Metaverse-enabled intelligence for open-terrain field vehicle fleets: Leveraging parallel intelligence and edge computing," *IEEE Trans. Intell. Veh.*, 2024.

[18] P. Tang, J. Li, and H. Sun, "A Review of Electric UAV Visual Detection and Navigation Technologies for Emergency Rescue Missions," *Sustainability*, vol. 16, no. 5, p. 2105, 2024.

[19] A. Saboor, E. Vinogradov, Z. Cuil, S. Coene, W. Joseph, and S. Pollin, "Elevating the future of mobility: UAV-enabled intelligent transportation systems," in *2024 7th International Conference on Advanced Communication Technologies and Networking (CommNet)*, IEEE, 2024, pp. 1–7.

[20] V. Chiciudean, H. Florea, R. Beche, F. Oniga, and S. Nedevschi, "Data augmentation for environment perception with unmanned aerial vehicles," *IEEE Trans. Intell. Veh.*, 2024.

[21] Y. Salami, V. Khajehvand, and E. Zeinali, "A new secure offloading approach for internet of vehicles in fog-cloud federation," *Sci. Rep.*, vol. 14, no. 1, p. 5576, 2024, doi: 10.1038/s41598-024-56141-y.

[22] Y. Salami, V. Khajehvand, and E. Zeinali, "SOS-FCI: a secure offloading scheme in fog–cloud-based IoT," *J. Supercomput.*, pp. 1–31, 2023.

[23] Y. Salami, V. Khajehvand, and E. Zeinali, "LSMAK-IOV: Lightweight Secure Mutual AKE Scheme in Fog-Based IoV," in *2024 10th International Conference on Artificial Intelligence and Robotics (QICAR)*, IEEE, 2024, pp. 1–5. doi: 10.1109/QICAR61538.2024.10496659.

[24] D. Kwon *et al.*, "Design of Secure Handover Authentication Scheme for Urban Air Mobility Environments," *IEEE Access*, vol. 10, pp. 42529–42541, 2022, doi: 10.1109/ACCESS.2022.3168843.

[25] A. Kumar and H. Om, "Handover authentication scheme for device-to-device outband communication in 5G-WLAN next generation heterogeneous networks," *Arab. J. Sci. Eng.*, vol. 43, no. 12, pp. 7961–7977, 2018.

[26] 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. Top. Comput.*, vol. 9, no. 3, pp. 1386–1396, 2020.
[27] Y. Zhou, X. Long, L. Chen, and Z. Yang, "Conditional privacy-preserving authentication and key agreement scheme for roaming services in VANETs," *J. Inf. Secur. Appl.*, vol. 47, pp. 295–301, 2019.
[28] S. ZakeriKia, R. Hajian, S. H. Erfani, and A. M. Rahmani, "Robust and anonymous handover authentication scheme without key escrow problem in vehicular sensor networks," *Wirel. Networks*, vol. 27, no. 7, pp. 4997–5028, 2021.
[29] N. Sharma and P. Dhiman, "Lightweight privacy preserving scheme for IoT based smart home," *Recent Adv. Electr. Electron. Eng. (Formerly Recent Patents Electr. Electron. Eng.*, vol. 17, no. 8, pp. 763–777, 2024.
[30] N. Sharma and P. Dhiman, "A secure addressing mutual authentication scheme for smart IoT home network," *Multimed. Tools Appl.*, pp. 1–33, 2024.
[31] N. Sharma and P. Dhiman, "Design of secure and unique addressing with mutual authentication scheme in IoT networks," *Peer-to-Peer Netw. Appl.*, vol. 18, no. 1, p. 50, 2025.
[32] N. Sharma and P. Dhiman, "Design of a Multifactor Unidentified Remote End User Authentication Mechanism for IoT Network," *Informatica*, vol. 49, no. 10, 2025.
[33] Y. Salami, "SOBT-UF: Secure Offloading in Blockchain Infrastructure for Intelligent Transportation Systems Using 5G-Enabled UAVs Within a Fog-Edge Computing Federation," in *2024 19th Iranian Conference on Intelligent Systems (ICIS)*, IEEE, 2024, pp. 217–222. doi: 10.1109/ICIS64839.2024.10887460.
[34] "Avispa." [Online]. Available: http://www.avispa-project.org/
[35] D. Von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," *Proc. APPSEM 2005 Work.*, pp. 1–17, 2005.
[36] D. Basin, S. Mödersheim, and L. Viganò, "An on-the-fly model-checker for security protocol analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2808, no. 3, 2003, pp. 253–270. doi: 10.1007/978-3-540-39650-5_15.
[37] M. Turuani, "The CL-Atse Protocol Analyser BT - Term Rewriting and Applications," F. Pfenning, Ed., Springer Berlin Heidelberg, 2006, pp. 277–286.
[38] D. Von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proceedings of APPSEM 2005 workshop*, 2005, pp. 1–17.
[39] Y. Salami, V. Khajehvand, and E. Zeinali, "E3C: a tool for evaluating communication and computation costs in authentication and key exchange protocol," *Iran J. Comput. Sci.*, pp. 1–11, 2024, doi: https://doi.org/10.1007/s42044-024-00176-x.
[40] "NS3." [Online]. Available: https://www.nsnam.org