Integrated Kerberos-Blockchain Authentication Framework for Securing Vehicular Ad-Hoc Network

September, 2025

Maya Rahayu

Graduate School of
Natural Science and Technology
(Doctor's Course)

OKAYAMA UNIVERSITY

DOCTORAL THESIS

Integrated Kerberos-Blockchain Authentication Framework for Securing Vehicular Ad-Hoc Network

Author:	Maya Rahayu
Supervisor:	Yasuyuki NOGAMI
Co-supervisors:	Kazuhiro UEHARA
	Yukinobu FUKUSHIMA

A dissertation submitted to
OKAYAMA UNIVERSITY
in fulfillment of the requirements for the degree of
Doctor of Philosophy in Engineering
in the
Graduate School of Natural Science and Technology

September, 2025

To Whom It May Concern

We hereby certify that this is a typical copy of the original doctor thesis of

Maya Rahayu

Signature of
The Supervisor

Seal of

Prof. Yasuyuki Nogami

Graduate School of
Natural Science and Technology

Declaration Authorship

This dissertation and the work presented here for doctoral studies were conducted under the supervision of Professor Yasuyuki Nogami. I, Maya Rahayu, declare that this thesis titled, "**Integrated Kerberos-Blockchain Authentication Framework for Securing Vehicular Ad-Hoc Network**" and the work presented in it are my own. I confirm that:

- The work presented in this thesis is the result of original research carried out by myself, in collaboration with others, while enrolled in the Graduate School of Natural Science and Technology at Okayama University as a candidate for the degree of Doctor of Philosophy in Engineering.
- This work has not been submitted for a degree or any other qualification at this University or any other institution.
- The published work of others cited in this thesis is clearly attributed. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help to pursue this work.
- The experiments and results presented in this thesis and in the articles where I am the first author were conducted by myself.

Signed: _____

Date: September 17, 2025

Abstract

Vehicular Ad-Hoc Network (VANET) is a key component of Intelligent Transportation Systems (ITS), supporting real-time communication among vehicles and infrastructure. However, the open and dynamic nature of VANET makes them highly vulnerable to various cyber threats, necessitating robust authentication mechanisms to ensure secure communication. One of the key challenges in VANET authentication lies in meeting the strict time requirements, as authentication delay must remain below 100 milliseconds. In addition, scalability across diverse deployment scenarios—such as suburban and urban environments with varying traffic densities—poses another critical concern. Furthermore, the cost and complexity of deploying traditional authentication infrastructures, especially when relying on additional servers to support handover and high vehicle density, further limit the feasibility of secure VANET deployments at scale.

This dissertation presents a novel authentication framework for VANET by integrating the Kerberos authentication mechanism with blockchain technology. The proposed solution is motivated by the need for secure, efficient, and scalable authentication in highly dynamic vehicular environments, where conventional approaches often suffer from excessive delays and infrastructure dependency. The system leverages Kerberos' authentication protocol and enhances it by storing authentication messages on a blockchain to reduce the need for repetitive validation during handover processes. This approach aims to achieve fast and secure authentication while minimizing signaling overhead and maintaining system simplicity.

The research begins with the implementation of an authentication framework for VANET that integrates Kerberos with blockchain technology to secure vehicle-to-infrastructure (V2I) communications. By storing Kerberos authentication messages in a blockchain ledger accessible to Trusted Authentication Servers (TAS) and Roadside Units (RSUs), the system ensures data integrity while minimizing authentication delays during handovers. This method eliminates the need for repeated connections to the TAS, thereby reducing signaling overhead. The system was evaluated in a simulated environment using OMNeT++ and a map of Tsushima area, Okayama, Japan, with 100 vehicles, 4 RSUs, and 1 TAS, confirming its effectiveness in achieving fast and secure

authentication.

Building upon this foundation, the second study introduces a performance and feasibility evaluation of the blockchain component in the previously proposed Kerberos–Blockchain authentication system for VANET. While the system had demonstrated low signaling overhead and authentication delay, the blockchain layer itself had not been thoroughly assessed. Through simulations involving 100 vehicles, 4 RSUs, and 1 TAS using OMNeT++ and the Ethereum blockchain, I analyzed gas used and memory size. The results confirmed that the blockchain integration is practically feasible and can support secure authentication in VANET environments.

In the third study, I evaluated the scalability and applicability of the proposed Kerberos–Blockchain authentication system across multiple network scenarios. Authentication messages were stored on a blockchain ledger accessible to both TASs and RSUs. Simulations were conducted in three environments: a suburban area with 100 vehicles and one TAS, an urban area with 200 vehicles and one TAS, and the same urban setting with an additional TAS. Results showed that while performance in the suburban scenario met all network delay requirements, the urban case without added infrastructure failed to maintain acceptable authentication times. However, introducing a second TAS effectively restored performance. The blockchain storage remained feasible in all cases, with gas values and memory usage staying within practical limits.

In the fourth study, I proposed modification from the previous system, Kerberos-Blockchain (KBC) approach that has separated Authentication Server (AS) and Ticket Granting Server (TGS), into a Combined Blockchain Server (CBS). This design aimed to address the rising infrastructure cost and authentication delays in dense urban networks. Through comprehensive simulations using OMNeT++ and SUMO for traffic modeling, and Ganache for blockchain implementation, I validated the system's effectiveness in both suburban and urban scenarios. Results demonstrated that CBS consistently maintained authentication delays under 100 ms, while achieving approximately 104% improvement in throughput and 45% reduction in signaling overhead compared to the separated AS–TGS approach. The CBS model thus enhances scalability and cost-efficiency without sacrificing security performance in VANET environments.

Overall, this research contributes a comprehensive authentication solution that adapts classical security frameworks to the demands of modern VANET infrastructures. The proposed framework demonstrates both theoretical soundness and practical applicability. Future directions include optimizing the CBS using lightweight cryptographic approach to address challenges in ultra-dense urban environments. These advancements aim to further strengthen the reliability and resilience of VANET security systems in next-generation ITS.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Yasuyuki Nogami, for his support throughout my doctoral courses at Okayama University. Without his extraordinary understanding and cooperation, I would not have been able to complete my doctoral research. I also appreciate my co-supervisors, Professor Kazuhiro Uehara, and Associate Professor Yukinobu Fukushima, who gave me a lot of effort to improve this thesis. They also gave me knowledge of wireless communication and networks through the classes in my courses.

I sincerely thank Associate Professor Yuta Kodera who always gave me a lot of influence from his great attitude for research. I also appreciate other teachers who have imparted a lot of knowledge to me through the seminars in the Information Security Engineering Laboratory.

I would like to express my heartfelt gratitude to Professor Md. Arshad Ali at Hajee Mohammad Danesh Science and Technology University, Bangladesh, for his wholehearted support and insightful discussions throughout my PhD journey. I thank also Assistant Professor Samsul Huda in Okayama University for the valuable scientific discussions.

Special thanks also go to the student members of the Information Security Engineering Laboratory for fostering a positive work atmosphere and for their generous support. My heartfelt thanks to Mr. Md. Biplob Hossain for his kind support in discussions, collaboration, and co-authoring publications in the field of blockchain.

Thanks to MEXT, Japan for the scholarship that fulfilled my dream to pursue doctoral study in Japan. I sincerely acknowledge all the funds that afforded me to join several conferences and conduct research activities.

I am also grateful to all administrative officers of the Faculty of Engineering who directly or indirectly made an impact on my doctoral course studies. My special thanks to Ms. Yuri Kunisada and Ms. Yumi Sato for their kind support in administrative work.

Last but not least, I cannot thank my husband enough for his sacrifices and support. I would also like to express my heartfelt thanks to my parents, my husband, and my son for allowing me to learn in the doctoral course. They gave me much encouragement, which motivated me to carry out research.

Publications

Peer-Reviewed Journal Paper:

1. **Maya Rahayu**, Md. Biplob Hossain, Samsul Huda, Yuta Kodera, Md. Arshad Ali, and Yasuyuki Nogami, "The Design and Implementation of Kerberos-Blockchain Vehicular Ad-Hoc Networks Authentication Across Diverse Network Scenarios," *sensors*, vol. 24, no. 23, Art. no. 7428, pp. 1–29, 2024.
   doi: 10.3390/s24237428. (Acceptance rate 56%).

Peer-Reviewed International Conference Papers (First author):

2. **Maya Rahayu**, Md. Biplob Hossain, Md. Arshad Ali, Samsul Huda, Yuta Kodera, and Yasuyuki Nogami, "An Integrated Secured Vehicular Ad-Hoc Network Leveraging Kerberos Authentication and Blockchain Technology," *Eleventh International Symposium on Computing and Networking Workshops (CANDARW)*, Matsue, Japan, pp. 260–266, Nov. 2023.
   doi: 10.1109/CANDARW60564.2023.00050. (Acceptance rate 37.2%).

3. **Maya Rahayu**, Md. Biplob Hossain, Samsul Huda, Md. Arshad Ali, Yuta Kodera, and Yasuyuki Nogami, "An In-depth Analysis of Kerberos and Blockchain Integration on VANETs' Security and Performance," *2024 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)* Taichung, Taiwan, pp. 391–392, Jul. 2024.
   doi: 10.1109/ICCE-Taiwan62264.2024.10674056. (Acceptance rate 69.4%).

Peer-Reviewed Journal Papers (Co-author):

4. Md. Biplob Hossain, **Maya Rahayu**, Md. Arshad Ali, Samsul Huda, Yuta Kodera, and Yasuyuki Nogami, "A Blockchain-based Approach with zk-SNARKs for Secure Email Applications," *International Journal of*

Networking and Computing, vol. 14, no. 2, pp. 225–247, 2024. (Acceptance rate 14.06% from CANDAR selected papers).

5. Samsul Huda, Yasuyuki Nogami, **Maya Rahayu**, Takuma Akada, Md. Biplob Hossain, Muhammad Bisri Musthafa, Yang Jie, and Le Hoang Anh, "IoT-Enabled Plant Monitoring System with Power Optimization and Secure Authentication," *Computers, Materials & Continua*, vol. 81, no. 2, pp. 3165–3187, 2024.
   doi: 10.32604/cmc.2024.058144.

Peer-Reviewed International Conference Papers (Co-author):

6. Samsul Huda, Yasuyuki Nogami, Takuma Akada, Maya Rahayu, **Md. Biplob Hossain**, Muhammad Bisri Musthafa, Le Hoang Anh, and Yang Jie, "A Proposal of IoT Application for Plant Monitoring System with AWS Cloud Service," *2023 International Conference on Smart Applications, Communications and Networking (SmartNets),* Istanbul, Turkiye, pp. 1–5, Jul. 2023.
   doi: 10.1109/SmartNets58706.2023.10215620.

7. Md. Biplob Hossain, **Maya Rahayu**, Md. Arshad Ali, Samsul Huda, Yuta Kodera, and Yasuyuki Nogami, "A Smart Contract Based Blockchain Approach Integrated with Elliptic Curve Cryptography for Secure Email Application," *Eleventh International Symposium on Computing and Networking Workshops (CANDARW)*, Matsue, Japan, pp. 195–201, Nov. 2023.
   doi: 10.1109/CANDARW60564.2023.00040. (Acceptance rate 37.2%)

8. Samsul Huda, Yasuyuki Nogami, Md. Biplob Hossain, Yang Jie, Le Hoang Anh, Muhammad Bisri Musthafa, **Maya Rahayu**, and Takuma Akada, "A Secure Authentication for Plant Monitoring System Sensor Data Access," *2024 IEEE International Conference on Consumer Electronics (ICCE),* Las Vegas, NV, USA, pp. 1–2. Jan. 2024.
   doi: 10.1109/ICCE59016.2024.10444465.

9. Md. Biplob Hossain, **Maya Rahayu**, Samsul Huda, Md. Arshad Ali, Yuta Kodera, and Yasuyuki Nogami, "A Blockchain-Based Approach for Secure Email Encryption with Variable ECC Key Lengths Selection," *The 8th International Conference on Mobile Internet Security (MobiSec)*, Sapporo, Japan, pp. 1–14, Dec. 2024. (Acceptance rate 30.1%).

10. Samsul Huda, Md. Biplob Hossain, **Maya Rahayu**, Andri Santoso, and Yasuyuki Nogami, "Design of Blockchain-Based Secure Device Authentication for IoT Plant Monitoring Systems," *2025 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)* Kaohsiung, Taiwan, Jul. 2025. [Accepted for publication]

Peer-Reviewed Book Chapter (Co-author):

11. Springer Communications in Computer and Information Science (CCIS) book series. [Accepted for inclusion]

    The paper will be published as part of the following volume:

    - Series Title: Communications in Computer and Information Science
    - Book Title: Mobile Internet Security
    - Book Subtitle: 8th International Conference, MobiSec 2024, Sapporo, Japan, December 17–19, 2024, Revised Selected Papers
    - Paper Title: A Blockchain-Based Approach for Secure Email Encryption with Variable ECC Key Lengths Selection

Table of Contents

List of Figures

List of Tables

Notations and Abbreviations

Entities

VANETs	Vehicular Ad-Hoc Networks
OBU	On-Board Unit
V	Vehicle
TAS	Trusted Authentication Server
KBC	Kerberos-Blockchain Server
AS	Authentication Server
TGS	Ticket Granting Server
CBS	Combined Server
RSU	Roadside Unit

Keys

K_{Vs}	Vehicle's key
K_{RSU}	RSU's key
K_{TGSs}	TGS's key
K_{TGSse}	TGS session key
K_{Sse}	Service session key
K_{Ss}	Service's key

Message Names

RA	Request authentication
TGT	Ticket Granting Ticket

ST	Service Ticket
AU_{VtTGS}	Vehicle auth V to TGS
AU_{VtS}	Vehicle auth V to Service
AU_{Sm}	Service Auth message
FA	Service Authentication Message
AT_{AStV}	Attributes AS to V
AT_{VtTGS}	Attributes V to TGS
AT_{TGStV}	Attributes TGS to V

Message Contents

ID_v	Vehicle's ID
ID_S	Service name ID
ID_{TGS}	TGS name ID
IP_v	User IP Address
Req_{TGT}	Request lifetime for TGT
Req_{LT}	Request lifetime for ticket
TS_{AStV}	Timestamp attributes AS to V
TS_{TGT}	Timestamp TGT
TS_{VA}	Timestamp Vehicle's Authenticator
TS_{TGStV}	Timestamp attribute TGS to V
TS_{ST}	Timestamp ST
TS_{VtS}	Timestamp vehicle to service
TS_{StV}	Timestamp service to authentication message

Other Parameters

LT_{TGT} Lifetime for TGT

LT_{ST} Lifetime for Service Ticket

Chapter 1

Introduction

This chapter provides problem outline and motivation underlying this research, the key contributions and novelty of the studies presented in this dissertation, and describes the overall structure of the dissertation.

1.1 Introduction

Vehicular Ad-Hoc Networks (VANETs) represent a foundational pillar in the realization of Intelligent Transportation Systems (ITS), offering dynamic, real-time communication among vehicles and infrastructure. By enabling efficient route optimization, traffic congestion avoidance, and rapid incident response, VANETs play a crucial role in supporting the Digital Transformation (DX) of the transportation sector. As modern urban centers and smart city initiatives increasingly rely on connected vehicle ecosystems, the importance of VANETs continues to grow in ensuring road safety, traffic efficiency, and intelligent mobility services.

Despite their transformative potential, VANETs are inherently exposed to significant cybersecurity risks. The open, decentralized, and rapidly changing topology of vehicular networks makes them vulnerable to attacks such as spoofing, Sybil, message tampering, and replay attacks. In particular, the process of authenticating entities that join or handover within the network is a major point of vulnerability. Effective authentication must be performed with minimal delay to maintain the real-time nature of VANET services. The strict requirement of under 100 milliseconds for authentication latency poses a technical challenge, especially under high vehicle density in urban settings. Furthermore, reliance on traditional centralized infrastructure often results in scalability limitations and deployment inefficiencies.

To address these challenges, this dissertation proposes an authentication framework that integrates the Kerberos protocol with blockchain technology. The core motivation is to enable secure, scalable, and low-latency authentication in VANET environments. The framework leverages the time-tested

ticket-based architecture of Kerberos while enhancing it with blockchain's decentralized and tamper-resistant features. Authentication messages are stored in a blockchain ledger accessible to trusted entities, eliminating the need for repeated validation and reducing signaling overhead during vehicle handovers. The modular design allows the solution to adapt across a range of deployment scenarios with varying traffic conditions.

The first study introduces the basic implementation of a Kerberos-based authentication scheme tailored for VANETs. The architecture adopts a two-tier system composed of an Authentication Server (AS) and a Ticket Granting Server (TGS), inspired by classical Kerberos. The system was tested using OMNeT++ in a simulated environment with 100 vehicles, 4 Roadside Units (RSUs), and 1 Trusted Authentication Server (TAS), utilizing a real city map of Tsushima, Japan. Results confirmed that the scheme successfully maintained authentication delays below the required 100 ms threshold, while also minimizing signaling overhead by avoiding redundant connections during handover.

Building upon the first stage, the second study focuses on evaluating the blockchain component of the proposed Kerberos–Blockchain authentication system. Although prior work demonstrated low authentication latency and overhead, it lacked a detailed performance analysis of the blockchain layer. This study utilized Ethereum and Truffle to simulate and assess a blockchain feasibility through gas consumption, and memory usage. The findings verified that the blockchain mechanism is computationally practical and can be reliably integrated into VANETs without excessive overhead.

The third publication expands the evaluation to examine the system's scalability and adaptability in various deployment scenarios. Authentication messages were stored on a blockchain ledger accessible to multiple TAS and RSU nodes. Simulations were carried out in three settings: a suburban environment with 100 vehicles and one TAS, an urban scenario with 200 vehicles and one TAS, and the same urban configuration with an additional TAS. While the suburban setup met all performance benchmarks, the urban setting with only one TAS struggled to keep authentication delay under 100 ms. However, adding a second TAS restored performance to acceptable levels. The system demonstrated consistent feasibility in terms of gas values and memory usage,

even as vehicle volume increased.

To overcome the infrastructure scalability limitations observed in the third study, the fourth research phase proposes a unified authentication architecture. This model consolidates the AS and TGS functions in the previous approach, Kerberos Blockchain-VANET (KBC), into a single Combined Blockchain Server (CBS). The CBS aims to reduce system complexity, latency, and infrastructure costs, particularly in dense urban environments. Simulations using OMNeT++ and SUMO were conducted across both suburban and urban scenarios. The CBS consistently maintained authentication delays below 100 ms, and compared to the KBC setup, it achieved roughly 104% higher throughput and 45% lower signaling overhead. This architecture offers a more scalable and cost-effective authentication framework for future ITS deployments.

Cumulatively, this dissertation presents a comprehensive, modular approach to VANET authentication that evolves from traditional protocol adaptation to blockchain integration and server architecture optimization. The results confirm the proposed system's effectiveness in balancing security, performance, and scalability across a wide range of real-world traffic scenarios. Future enhancements include optimizing CBS using lightweight cryptographic approach and implementing location-based trust mechanisms to address threats such as position spoofing in ultra-dense urban deployments. These contributions mark a significant step toward deploying secure and intelligent vehicular networks at scale.

1.2 Problem outline and motivation

Vehicular Ad-hoc Networks (VANETs) are a cornerstone of intelligent transportation systems, enabling vehicles to communicate important safety and traffic information in real-time. By supporting vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) messages (e.g., speed, position, hazard alerts), VANETs improve traffic efficiency, optimize routes, and improve emergency response and accident avoidance [1]. These capabilities make transportation systems safer and more efficient.

However, the open wireless medium and highly dynamic topology of VANETs

also introduce serious security challenges. The network's exposure to adversaries (e.g. due to high vehicle speeds and frequent handovers) makes VANETs vulnerable to attacks such as message manipulation or spoofing [2]. In particular, ensuring the authenticity and integrity of messages is critical, since delayed or corrupted information can directly affect driving decisions. Consequently, robust authentication of network entities is essential for VANET security and integrity [3].

Achieving secure and efficient authentication for vehicles as they move across network domains is an essential thing in VANET. Vehicles frequently roam between roadside units (RSUs) or network zones, requiring re-authentication at each handoff. Even existing Kerberos-based methods (which avoid sending passwords directly) can incur latency or overhead during each handover [4]. Moreover, minimizing authentication latency to meet the strict timing requirements of safety-critical VANET applications. Authentication must be completed in under 100 ms to prevent outdated information from affecting vehicle behavior. Conventional systems may not guarantee such performance under dynamic network conditions. Therefore, designing low-latency, high-speed authentication mechanisms remains a critical task in VANET security [5, 6].

In our first study, we proposed a novel system that integrates Kerberos authentication with blockchain to store authentication messages in a distributed ledger accessible by RSUs. This framework eliminates the need for continuous communication with a centralized Trusted Authentication Server (TAS) during re-authentication, thereby reducing handover delay.

The proposed system was validated through simulations using OMNeT++ and evaluated in a suburban VANET scenario. The results demonstrated a significant reduction in handover delay and signaling overhead, confirming the feasibility of using blockchain to support distributed authentication. However, this study did not evaluate the performance of the blockchain environment itself, particularly its scalability and cost-effectiveness in large-scale deployments.

To address this limitation, our second study focused on evaluating the performance of the blockchain infrastructure supporting the authentication system. I evaluated the feasibility of deploying the Ethereum blockchain for VANETs by analyzing gas consumption and transaction costs in different block

sizes and vehicle densities.

The evaluation, conducted using Truffle and Ganache, demonstrated that gas usage and block size growth remained within acceptable limits under increasing network loads. These findings validated the practicality of blockchain-based authentication for VANETs.

While the previous study provided valuable insights through on-chain analysis, it was limited in scope as it did not examine authentication performance across diverse network environments. A significant challenge that remains is ensuring the scalability of the authentication system in various VANET scenarios. Urban and suburban settings differ substantially in terms of node density, mobility patterns, and network complexity, requiring adaptable solutions. To be viable for real-world deployment, a robust authentication framework must maintain low latency and efficient resource usage even in high-density traffic and dynamically changing topologies [7, 8].

To overcome those limitations, our third study implemented the full Kerberos-Blockchain authentication system and evaluated its performance in multiple VANET environments: suburban, urban with one TAS, and urban with two TASs. This study examined both network performance metrics and blockchain behavior.

The experimental results showed that the system met the authentication and handover delay requirements in all scenarios and maintained efficient gas and memory usage. It confirmed the system's scalability and adaptability to different vehicular densities.

However, in large-scale VANET deployments, expanding the authentication infrastructure to maintain low authentication delay often results in increased system complexity and cost. As the number of Trusted Authentication Servers (TASs) grows, so does the need for careful coordination, leading to synchronization challenges and operational overhead [9, 10]. This escalation raises concerns regarding the practicality of real-world implementation in high-density urban areas. Thus, balancing authentication performance with infrastructure cost and manageability remains a persistent and critical challenge in the design of scalable VANET authentication architectures.

To resolve this infrastructure challenge, our fourth study proposed an integrated server architecture that combines the Authentication Server (AS) and

Ticket Granting Server (TGS) into a single Combined Kerberos Server (CBS). This design aims to eliminate the need for additional TASs while preserving authentication speed and blockchain-based security.

Simulations using OMNeT++ and SUMO demonstrated that the integrated CBS architecture reduces authentication delay, improves throughput by 104%, and decreases signaling overhead by 45% compared to the previous architecture. These results confirmed that the optimized server design maintains performance while reducing infrastructure complexity and cost, making it more suitable for real-world VANET deployment.

1.3 Major contributions

The following are the main contributions of this dissertation:

- **Design of a Blockchain-based Kerberos Authentication Scheme in VANETs (first study):**

  This study is detailed in Chapter 4 and was published in the 2023 CANDARW conference [4]. The contributions of this work are as follows:

  - Proposed a novel integration of Kerberos authentication with blockchain technology to improve the security and efficiency of VANET authentication.
  - Designed a method for storing Kerberos authentication messages in a distributed blockchain ledger, enabling fast and secure vehicle handovers.
  - Demonstrated significant reduction in signaling overhead and authentication delay in handover scenarios.
  - Verified the feasibility and performance of the proposed scheme through OMNeT++ simulation in a suburban VANET environment.

- **Performance Evaluation of Blockchain Component in VANET Authentication (second study):**

This study is presented in Chapter 5, with the main results published in the 2024 ICCE-Taiwan conference [6]. The major contributions include:

  - Investigated the performance of the blockchain infrastructure used in the proposed authentication system.
  - Evaluated gas consumption and transaction feasibility using Ethereum, Truffle, and Ganache under various block sizes and vehicle numbers.
  - Demonstrated that the proposed system meets the practical requirements for transaction cost and block gas limits.

- **Comprehensive System Implementation and Evaluation Across Diverse VANET Scenarios (third study):**

  Chapter 6 elaborates this study, with its results published in *Sensors* journal [8]. The key contributions are:

  - Implemented a full authentication system using Kerberos and blockchain in Ethereum across three different VANET scenarios (suburban, urban with 1 TAS, and urban with 2 TAS).
  - Evaluated authentication delay, handover delay, end-to-end delay, and blockchain metrics (gas usage and memory size).
  - Validated that the system performs reliably under increasing network loads and dynamic conditions.

- **Design of an Integrated Authentication Server Architecture (fourth study):**

  This study is discussed in Chapter 7, and its findings are published in *IEEE Access* [9]. The contributions of this study include:

  - Proposed an optimization of Kerberos-Blockchain authentication architecture by combining AS and TGS into a single Combined Kerberos Server (CBS).
  - Eliminated the need for additional Trusted Authentication Servers (TAS), reducing infrastructure costs.
  - Achieved lower authentication delay, increased throughput, and reduced signaling overhead in both suburban and urban simulations.

1.4 Thesis outline

This subsection provides an overview of the structure of this dissertation. Each chapter is organized to reflect the progression of the research, beginning with foundational concepts, followed by system design, implementation, evaluation, and concluding discussions.

Some of the results presented in this dissertation have been published in international journals and conference proceedings. Specifically, parts of Chapter 4 were published in the 2023 CANDARW conference [4], and the findings in Chapter 5 were presented at the 2024 IEEE ICCE-Taiwan conference [6]. Chapter 6 consolidates and extends evaluations introduced in the *Sensors* journal [8], while Chapter 7 includes results published in *IEEE Access* [9].

The remaining structure of this dissertation is as follows:

Chapter 1: Introduces the research problem, motivation, and challenges related to authentication in VANETs. It outlines the contributions and novelty of this work and provides an overview of the dissertation structure.

Chapter 2: Presents relevant works in literature for this thesis, includes previous authentication methods in VANETs, utilizing the cryptography methods, blockchain, etc.

Chapter 3: Presents a comprehensive review of the underlying technologies and concepts relevant to this study, including VANET architecture, Kerberos authentication, blockchain mechanisms, and the simulation tools used such as OMNeT++, SUMO, Truffle, and Ganache.

Chapter 4: Proposes a Kerberos-Blockchain integrated authentication framework for VANETs. It describes the system architecture, handover optimization strategies, and evaluates its performance through simulations.

Chapter 5: Analyzes the blockchain environment and its impact on authentication performance. This includes gas consumption, block size constraints, and a security evaluation under on-chain and off-chain conditions.

Chapter 6: Describes the implementation and evaluation of the proposed authentication framework across different VANET scenarios, such as suburban, urban (1 TAS), and urban (2 TAS). It compares both network and blockchain performance metrics.

Chapter 7: Introduces the design of an integrated authentication server

architecture that combines the AS and TGS into a single entity (CBS). The chapter evaluates the design against traditional models in terms of delay, throughput, and signaling overhead.

Chapter 8: Concludes the dissertation by highlighting key contributions, limitations, and suggesting potential directions for future work.

Chapter 2

Related Works in Literature

This chapter introduces relevant works from theom the literature for this thesis. Several works have discussed VANETs, VANETs authentication using the cryptography methods and blockchain methods.

2.1 Introduction

Vehicular Ad-Hoc Networks (VANETs) are a core component of Intelligent Transportation Systems (ITS), enabling real-time communication between vehicles and infrastructure to support traffic efficiency, safety, and emergency response. However, the open and dynamic nature of VANETs poses significant security challenges, such as message falsification and identity spoofing. Fast and reliable authentication is therefore critical, especially since most VANET applications require latency below 100 milliseconds.

Many authentication methods have been proposed, including encryption-based schemes and centralized authorities. Kerberos offers strong mutual authentication, while blockchain ensures decentralized and tamper-proof data handling. Yet, these methods are often applied separately, and many rely on multi-server setups that increase overhead. Additionally, the storage of authentication messages for handover scenarios remains underexplored.

This chapter reviews current authentication approaches in VANETs, highlighting their strengths and limitations. It also positions the proposed integrated Kerberos–Blockchain scheme as a novel contribution that aims to reduce authentication delay and improve message security in a unified, streamlined architecture.

2.2 Literature

The Intelligent Transportation System (ITS) is recognized as a key component in the future transportation model, contributing significantly to the de-

velopment of advanced transportation networks in the era of Digital Transformation (DX) [11]. One of ITS's core enablers is the Vehicular Ad-Hoc Network (VANET), which facilitates communication between vehicles and roadside infrastructure to deliver critical information such as speed, location, trajectory, and urgent warnings about hazardous conditions [12]. These capabilities play a vital role in improving traffic efficiency, route optimization, congestion reduction, and emergency response coordination [13].

However, due to VANETs' open and dynamic nature, with high node mobility and frequently changing network topology, security becomes a central concern [14]. Vehicles can easily disconnect or reconnect, creating vulnerabilities that attackers may exploit to inject false messages or retrieve confidential credentials [15]. In this context, message and identity authentication is essential to prevent the distribution of misleading or malicious data, which could severely impact infrastructure and endanger lives [16]. Quick and reliable authentication mechanisms are crucial, as most VANET safety applications require latency no higher than 100 milliseconds to maintain accurate, real-time decision-making [17].

Traditional encryption and key management techniques, though effective in fixed networks, are insufficient in VANET environments due to their dynamic topology. Moreover, insider threats from authenticated participants pose additional challenges in maintaining message confidentiality and authenticity during dissemination [18].

The Kerberos protocol functions as a robust authentication mechanism specifically crafted to enhance security within networks that are susceptible to breaches, utilizing session keys for transmitting data during the signaling phase as opposed to the principal key [19]. However, the centralized structure of the Kerberos authentication mechanism brings forth added complicacy, which could lead to extended authentication delay and add complications to the transition process due to the participation of numerous entities.

In the field of VANETs, many current solutions rely on a central trusted authority, which is not a scalable solution and becomes the network's single point of failure. To address these issues, researchers introduce a decentralized blockchain-based authentication solution for VANETs that integrates blockchain with VANETs. This ensures the distributed structure and pre-

serves an immutable ledger of data, strengthening the system's integrity for VANETs. The Inter Planetary File System (IPFS), Ciphertextbased Attribute Encryption (CP-ABE), and the Ethereum blockchain are the foundations for the distributed VANET system suggested in [20].

Several pieces of research have been conducted to create secure VANET authentication, as shown in Table 2.1. A Two-Factor Lightweight Privacy-preserving (2FLIP) scheme has been introduced in an authentication method for Vehicular Ad-Hoc Networks (VANETs), utilizing a decentralized certificate authority and a biological-password-based, two-factor authentication to significantly reduce computation and communication overhead while ensuring strong privacy preservation and resilience against denial-of-service attacks [21]. In [22], the authors design a protocol to ensure Secure and Efficient Message Authentication (SEMA) between vehicles and Roadside Units (RSUs), focusing on preventing vehicles from being falsely accused and ensuring robustness against various security attacks. It reaches that goal through a combination of pseudonym-based and group-based methods. SEMA achieves mutual authentication between vehicles and RSUs, ensuring that the communication is not only secure but also respects the privacy of the participants. In [23], it introduces a secure and efficient authentication protocol specifically designed for Vehicle-to-Vehicle (V2V) communication within Vehicular Ad-Hoc Networks (VANETs) and Internet of Vehicle (IoV) technologies, aiming to enhance traffic system management and road safety. A novel feature of the protocol is the vehicle password change phase, which incorporates the use of a honey list technique to thwart offline password-guessing attacks, enhancing the overall security of the system. However, those three articles [21–23] still utilize the centralized network, which can create a single point of failure, and the exchanged data can be altered or tampered with, leading to potential trust issues.

To overcome those issues, several researchers have applied blockchain technology during the VANET authentication phase. In [24], the authors build a new blockchain-based authentication infrastructure for wireless networks that utilizes AES, the Temporal Key Integrity Protocol (TKIP), and the Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) to secure the user's login information, using blockchain to verify user credentials. It utilized the hyperledger fabric blockchain in its

proposed method. In [25–27], the authors combine the ECC and blockchain certificate management to make a secure and decentralized authentication in VANETs. However, among all the proposed authentication methods that utilize blockchain, every paper utilizes and collects the data from either the network or the blockchain environment only, and none of them utilize data from both the network and the blockchain side.

Table 2.1: Comparative study of VANET authentication methods.

Reference	**Security Methods**	**Blockchain**	**Simulation Environment**	**Evaluation Area**	**Evaluated Parameters**
[21]	CA decentralization and biological-password-based	No	ONE (Opportunistic Networking Env.), ProVerif	Beijing (Urban)	Auth. overhead and certificate update cost
[22]	Pseudonym-based and group authentication	No	ns3, SUMO, MIRACL	Guilin (Urban)	RSU communication cost, v2v authentication
[23]	Honeylist, SHA-256, XOR	No	ns3	Not indicated	Delay, throughput, comp. cost, energy use
[24]	AES, TKIP, CCMP, Blockchain	Yes	Hyperledger, JMeter	Not indicated	Authentication time
[25]	ECC, Blockchain	Yes	Hyperledger Fabric	Not indicated	BDRA stability, theoretical cost
[26]	ECC, Blockchain	Yes	ns3, BAN logic	Not indicated	Delay, throughput, computation cost
[27]	ECC, Blockchain	Yes	ns2	Not indicated	Auth. delay, gas cost, latency
[Proposed Method]	Kerberos, AES, Blockchain	Yes	OMNeT++, SUMO, Truffle, Ganache	Urban + Suburban	Auth. delay, handover, gas cost, memory size

In our proposal, I integrate Kerberos authentication and blockchain to conduct the innovational authentication system for VANETs. This approach stores Kerberos authentication messages in the blockchain's distributed ledger that can be accessed in the Trusted Authentication Server (TAS) and all RSUs. This authentication message storing aims to simplify handover delay processes, shorten authentication delay, and securely keep the authentication message. To further improve system performance, I implement Kerberos using AES-128 encryption instead of the original Data Encryption Standard 77 (DES77) [28] aiming to reduce authentication time. Then, I assess the feasibility of blockchain technology for VANET authentication scenarios using Ethereum and simulate the process with OMNeT++.

To evaluate its effectiveness, I have designed three different scenarios.

The first scenario involves simulating the system in a suburban environment within the Tsushima Campus area at Okayama University, Japan. This simulation involves 100 vehicles and one Trusted Authority Server (TAS). The second scenario involves an urban environment in the Okayama Station area, featuring a higher vehicle density with 200 vehicles and one TAS. Then, the third scenario is a variation of the second but includes an additional TAS, totaling two TASs. In our evaluation, I focus on network performance metrics such as authentication delay, handover delay, and end-to-end delay. Additionally, I assess blockchain performance by measuring factors such as gas usage and the memory size of the blocks.

Overall, prior literature reveals that while blockchain can enhance privacy and trust, and Kerberos can provide strong mutual authentication, the combination of the two has not yet been explored in a unified-server setting. Most existing Kerberos-blockchain schemes maintain multiple servers and do not fully address signaling cost in dense networks. The proposed approach differs by merging the AS and TGS functionalities into one server, thereby reducing protocol rounds. This directly fills the gap of minimizing authentication overhead identified in earlier studies.

2.3 Summary

Vehicular Ad-Hoc Networks (VANETs) are a critical component of future transportation systems, enabling efficient and real-time communication between vehicles and infrastructure to support safer and more intelligent mobility. These networks offer significant benefits, such as improved traffic flow, route optimization, and enhanced emergency responses. However, the dynamic and open nature of VANETs presents substantial security challenges. High mobility, frequent disconnections, and rapidly changing topologies make VANETs vulnerable to various attacks, especially those aiming to manipulate or steal critical information. Therefore, ensuring message authenticity and the legitimacy of communicating entities is vital for maintaining the safety and trustworthiness of VANETs. Authentication mechanisms must also meet strict latency requirements, typically under 100 milliseconds, to support real-time safety applications.

To address these security concerns, various authentication frameworks have been proposed, including centralized approaches like the Kerberos protocol and decentralized methods based on blockchain technology. While Kerberos offers strong session-based authentication, its centralized structure can lead to delays and single points of failure. On the other hand, blockchain-based solutions provide a decentralized, tamper-resistant environment that enhances trust and integrity. Despite advancements, many existing methods still rely solely on either network-level or blockchain-level data, missing the opportunity to integrate both for comprehensive protection. This gap suggests the need for a hybrid solution that leverages the strengths of both systems to ensure secure, efficient, and scalable authentication in VANETs.

Chapter 3

Fundamental Theory of Blockchain-Based Authentication in VANET

This chapter provides the fundamental concepts and background required to understand the authentication framework proposed in this dissertation. It introduces the architecture and characteristics of VANETs, explores various authentication mechanisms, and reviews the Kerberos protocol and blockchain technology. Additionally, it presents the set of tools used for system implementation and evaluation.

3.1 VANET Architecture and Characteristics

Vehicular Ad-Hoc Networks (VANETs) have emerged as a critical enabler for intelligent transportation systems (ITS), offering real-time vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication to enhance road safety, traffic efficiency, and driver comfort. As a specialized form of mobile ad hoc networks (MANETs), VANETs are characterized by high node mobility, dynamic topologies, and stringent latency requirements. To support these functionalities, VANETs incorporate a combination of vehicular components, roadside infrastructure, and standardized communication protocols. This section introduces the fundamental architecture of VANETs and outlines their key operational characteristics that influence protocol design and performance evaluation.

3.1.1 VANET Architecture

A typical VANET consists of vehicles equipped with wireless communication units and fixed roadside infrastructure. Each vehicle carries an *on-board unit* (OBU) that integrates a DSRC (Dedicated Short-Range Communications) radio transceiver, GPS/IMU positioning, and processing hardware for

running vehicular applications. The OBU is responsible for sending and receiving safety and non-safety messages over the 5.9 GHz band. For example, Rahayu *et al.* note that "every vehicle is embedded with an On-Board Unit (OBU), which has the role of effectively supporting the process of transmitting and receiving" network messages [8]. In practice, the OBU often interfaces with in-vehicle sensors (cameras, radars, etc.) and actuators, but its defining feature is the V2X radio and protocol stack. Some architectures also distinguish an in-vehicle *application unit* (AU) that hosts safety or infotainment applications and connects via the OBU's wireless link. In any case, vehicles in a VANET are essentially mobile nodes (with ample power and compute) that dynamically form ad hoc connections with each other and with infrastructure.

Roadside Units (RSUs) are the fixed infrastructure component of a VANET. RSUs are installed along roads, intersections, or highways, and typically have stable power and larger processing/storage capabilities than vehicles. An RSU usually contains a DSRC radio (same 5.9 GHz standard) and may be connected to backhaul networks or the Internet. RSUs act as intermediaries for vehicle-to-infrastructure (V2I) communication: they collect messages broadcast by passing vehicles (e.g., traffic reports or emergency alerts) and can disseminate information (such as map updates or traffic advisories). As Guerna *et al.* observe, RSUs provide "stable and high communication, computing, and cache capabilities" to gather and analyze traffic data [29]. RSUs complement the mobile OBUs by offering broader coverage and infrastructure-grade services. Typical DSRC-based RSUs have radio ranges on the order of hundreds of meters to a few kilometers [8].

VANETs have largely adopted the IEEE 802.11p (WAVE/DSRC) protocol as the communication standard. IEEE 802.11p operates in the 5.850–5.925 GHz band with 10 MHz channels and is optimized for low-latency vehicular broadcasts. It supports data rates up to 27 Mbps under ideal conditions. In Rahayu *et al.*'s model, the communication stack follows 802.11p, with OBUs transmitting at 27 Mbps and a typical communication range of around 100 m [8]. Recent improvements have introduced IEEE 802.11bd (also known as Next-Generation V2X), which offers enhanced throughput and reliability. Xue *et al.* report that IEEE 802.11bd "greatly exceeds the performance" of 802.11p in dynamic traffic scenarios [30]. Nonetheless, 802.11p remains the most referenced

and standardized MAC/PHY layer protocol in current VANET studies.

V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure) communication are the two fundamental modes of VANET communication. V2V denotes direct peer-to-peer communication between nearby vehicles, primarily used for real-time safety message broadcasting. For example, vehicles can exchange periodic beacons about their location, speed, and heading to avoid collisions. These messages are usually broadcast within a range of 100–300 m [8]. V2I communication, in contrast, allows vehicles to interact with RSUs, either directly or via multi-hop relays. Within the coverage area (typically 1–3 km), vehicles can receive traffic alerts or upload sensor data [29]. Many VANET protocols explicitly distinguish between V2V and V2I to optimize routing, scheduling, and priority settings.

3.1.2 VANET Characteristics

Vehicular networks exhibit several unique characteristics that differentiate them from conventional MANETs. The first defining trait is *high mobility*. Vehicles often move at high speeds—ranging from 30–60 km/h in cities and up to 120 km/h on highways. As a result, network topologies change rapidly, and communication links are frequently established and broken. Rahayu *et al.* emphasize that VANETs are characterized by "high node mobility," with vehicles frequently joining and leaving communication ranges [8]. Kalaivani also notes that vehicles "moving at high speed make the topology dynamic" [31].

Second, VANETs feature a *dynamic network topology*. Due to continuous vehicle motion, the network graph evolves rapidly. Connections between vehicles may exist for just a few seconds, creating challenges for route maintenance and stability. This requires VANET routing protocols to be opportunistic and resilient. Both Rahayu *et al.* and Kalaivani describe this property as a key challenge in VANET protocol design [8, 31].

Third, VANETs operate under *real-time constraints*. Many applications such as emergency warnings or lane-change assistance—must be executed with minimal latency. Rahayu *et al.* note that VANETs have "real-time transmission constraints," particularly for safety-critical messages that need to be delivered within strict time bounds (often under 100 ms) [8]. Kalaivani adds

that these systems require "hard delay constraints" to be effective in emergencies [31].

Finally, VANET links are highly *volatile*. A communication opportunity may be extremely brief due to speed or poor connectivity. For instance, a vehicle may only be within range of an RSU for a few seconds, termed "dwell time" in Rahayu *et al.*'s work [8]. This implies that VANET protocols must be designed to function even under sporadic, intermittent, and rapidly-changing connectivity patterns [31].

3.2 Authentication Mechanisms in VANETs

Vehicular ad hoc networks (VANETs) require robust authentication to ensure that messages originate from legitimate vehicles and have not been tampered with. In these networks, open wireless channels and high node mobility create a threat environment where adversaries could spoof identities or inject false information. As noted by Yadav and Yadav, "the identity of the sender must be secure," motivating the application of various cryptographic authentication schemes (e.g., digital signatures, ring signatures, blockchain-based methods) to safeguard privacy and trust [32]. In the following, I review three fundamental approaches for VANET authentication: cryptographic protocols, public key infrastructure (PKI), and blockchain-based methods. Each subsection below presents the theoretical foundations of these techniques.

3.2.1 Cryptographic Protocols

Cryptographic protocols use mathematical algorithms to achieve security goals such as confidentiality, integrity, and entity authentication. In VANETs, these protocols typically combine symmetric and asymmetric cryptography. *Symmetric-key* algorithms (e.g., AES) enable efficient encryption and decryption of messages using a shared secret, as well as message authentication codes (MACs) for integrity. *Asymmetric-key* methods (public-key cryptography) use key pairs: a private key held by one entity and a public key known to others. Public-key schemes (e.g., RSA or elliptic-curve cryptography) support digital signatures, where a sender signs a message with its private key and any receiver

can verify it with the corresponding public key. This provides authentication (assurance of the sender's identity) and non-repudiation. In practice, standards such as IEEE 1609.2 specify the use of Elliptic Curve Digital Signature Algorithm (ECDSA) to sign safety messages in VANETs [33]. Key agreement protocols (e.g., Diffie–Hellman) are also used to establish shared symmetric keys between entities over the insecure channel. In summary, a VANET authentication protocol is typically composed of sequences of cryptographic operations (key exchanges, encryption/decryption, signing/verification) that let vehicles prove identity and verify messages [33, 34].

Fundamental cryptographic primitives include:

- **Symmetric encryption:** Algorithms like AES encrypt data with a secret key shared by sender and receiver. This provides confidentiality and, when combined with a MAC, data integrity.
- **Message Authentication Code (MAC):** A short tag (e.g., using HMAC) computed over a message with a symmetric key, allowing recipients to verify the message's integrity and authenticity given the shared key.
- **Public-key cryptography:** Asymmetric schemes (e.g., RSA, ECDSA) use public/private key pairs. Digital signatures computed with a private key can be verified by others with the public key, ensuring the signer's identity.
- **Cryptographic hash functions:** One-way functions (e.g., SHA-256) map data to fixed-size hashes. Hashes are used to create unique message digests (for integrity checks) and to link protocol steps (e.g., chaining commitments).
- **Key agreement protocols:** Methods such as Diffie–Hellman allow two vehicles to establish a shared secret over an insecure channel, which can then be used for symmetric encryption.

These cryptographic protocols form the theoretical basis for message authentication in VANETs [33].

3.2.2 Public Key Infrastructure (PKI)

Public Key Infrastructure (PKI) provides the framework to manage public and private keys and to bind them to vehicle identities via digital certificates. In PKI, a trusted *Certificate Authority* (CA) issues a digital certificate to each vehicle. A digital certificate contains the vehicle's public key and identity information (or a pseudonym), and is digitally signed by the CA. Any node can then authenticate a vehicle by verifying the CA's signature on its certificate and checking that the certificate has not expired or been revoked. In essence, PKI enables entities to trust each other's public keys: the CA vouches for each binding between a key and a vehicle identity [35].

A PKI encompasses the policies, hardware, software, and procedures for certificate life-cycle management: issuance, distribution, and revocation. For instance, when a vehicle joins the network, it registers with the CA (or a Registration Authority) and obtains a certificate. If a private key is compromised, the CA can invalidate the corresponding certificate and broadcast this via a Certificate Revocation List (CRL). In the VANET context, standards like IEEE 1609.2 define lightweight certificate formats and CRL distribution mechanisms suited for high-speed vehicles. World Bank documentation summarizes PKI as "the system—including policies, institutions, and technologies—that manages the distribution, authentication, and revocation of digital certificates" [35]. Thus, PKI provides a hierarchical trust model: vehicles accept messages as authentic if and only if the attached certificate chain leads to a trusted root CA.

3.2.3 Blockchain-based Authentication

Blockchain introduces a decentralized model for authentication by using a distributed ledger. A *blockchain* is a series of blocks, each containing a set of transactions and a cryptographic hash of the previous block, forming an immutable chain [33]. Transactions in this context could represent the registration of vehicle identities, the issuance of credentials, or updates to trust parameters. Blockchain systems rely on consensus mechanisms (e.g., Proof-of-Work or Proof-of-Authority) to agree on each new block, thereby establishing trust in an open network without a single central authority [33,34].

The combination of cryptographic hashes and decentralized consensus makes the blockchain tamper-evident: once a block is confirmed, its data cannot be altered without detection [33].

Key properties of blockchain relevant to authentication include:

- **Decentralization:** The ledger is maintained by all participating nodes rather than a single CA, eliminating a single point of trust or failure [33].
- **Immutability:** Each block contains the hash of the previous block; altering any transaction would change these hashes and break the chain, so recorded data (e.g., certificates) cannot be forged or modified undetected [33].
- **Consensus-driven trust:** Protocols like Proof-of-Work or Practical Byzantine Fault Tolerance require a majority of nodes to validate each new block, allowing the network to agree on a common state even if some nodes are malicious [33].
- **Transparency:** All network participants can view the ledger, so any addition (such as a new vehicle registration) is visible and verifiable by all, enhancing auditability.

In a blockchain-based VANET authentication scheme, one might store vehicles' public keys or credential hashes on the blockchain. Since blockchain entries cannot be retroactively changed without consensus, an adversary cannot surreptitiously insert a fake credential or revoke another vehicle's identity. In theory, this decentralized ledger replaces or complements the CA: vehicles trust the blockchain's consensus instead of a single certificate authority. Overall, blockchain's cryptographic and decentralized structure provides a fundamental framework for distributed authentication in vehicular networks [33].

3.3 Kerberos Authentication Protocol

Kerberos is a symmetric key-based authentication protocol originally designed at MIT to enable secure user identification over untrusted networks [36].

It is widely used in distributed systems such as UNIX environments and Active Directory due to its capability for mutual authentication and session key distribution without transmitting passwords.

3.3.1 Kerberos in Traditional and Emerging Contexts

The original Kerberos design involves a centralized Key Distribution Center (KDC), which consists of two roles: the Authentication Server (AS) and the Ticket Granting Server (TGS) [37]. Clients authenticate to the AS using a shared secret (e.g., password-derived key) to receive a Ticket-Granting Ticket (TGT). This TGT is later used to request service tickets from the TGS, allowing access to multiple services without re-entering credentials [38].

In IoT and VANET environments, however, the assumptions of stable infrastructure and powerful devices often do not hold. To address these limitations, Kerberos has been adapted in several ways. For example, KESIC introduces intermediary servers to handle tickets on behalf of constrained IoT nodes [39], while DKSM decentralizes ticket issuance using blockchain to improve availability and scalability [40]. For VANETs, integration with blockchain has been proposed to replicate Kerberos credentials across distributed RSUs, supporting faster authentication during vehicle mobility [8].

3.3.2 Protocol Structure and Exchange Roles

Kerberos operates in three main phases:

- **AS Exchange:** The client authenticates with the AS to obtain a TGT and session key $K_{c,tgs}$.
- **TGS Exchange:** Using the TGT, the client requests a service ticket from the TGS, receiving $K_{c,s}$ and a service ticket.
- **Client–Server Exchange:** The client presents the ticket to the target service and proves identity using an authenticator encrypted under $K_{c,s}$ [38].

3.3.3 Security Features and Ticket-Based Protection

Kerberos includes several key security properties:

- **Replay protection:** Timestamps and nonces prevent reuse of tickets or authenticators [37].
- **Session keys:** Fresh symmetric keys are issued per session to ensure forward secrecy.
- **Confidentiality and integrity:** All tickets and authenticators are encrypted using symmetric cryptography, ensuring they are tamper-proof and readable only by the intended party [38].
- **Limited ticket lifetimes:** Tickets expire after a short time, reducing vulnerability in case of compromise.

These properties make Kerberos a lightweight yet secure solution, which explains its adaptation in resource-constrained or delay-sensitive environments such as IoT and vehicular networks [8, 39, 40].

3.4 Adopted Open Tools

This section introduces the open-source tools used to implement and evaluate the proposed VANET authentication framework. OMNeT++ was employed as the primary network simulator, with INET and VEINS providing protocol and vehicular communication models. SUMO was integrated to simulate realistic traffic mobility. For blockchain development, Truffle and Ganache were used to deploy and test smart contracts in a local Ethereum-compatible environment. These tools collectively support comprehensive validation of the system's design and performance.

3.4.1 OMNeT++

"OMNeT++ is an object-oriented modular discrete event network simulation framework"[41]. It is implemented in C++ and provides a generic architecture for building custom network simulators. Users create models as hier-

archically nested modules (using the NED language) that exchange messages, enabling flexible composition of complex systems. OMNeT++ supports features like parallel execution and graphical animation (via its Qtenv interface), making it widely used for research in communication networks and distributed systems[41].

3.4.1.1 INET

"The INET Framework is an open-source OMNeT++ model suite for wired, wireless and mobile networks"[42]. It implements a wide range of network protocols (e.g., Ethernet, IP, TCP, UDP, 802.11, routing protocols) and device models (wired interfaces, WLAN radios, sensors, etc.), as well as mobility and application models. By providing off-the-shelf implementations of the Internet stack and other protocols, INET enables realistic simulation of traditional and ad-hoc networks in OMNeT++[42].

3.4.1.2 VEINS

"Veins is an open source framework for running vehicular network simulations"[43]. It couples the OMNeT++ network simulator with the SUMO road-traffic simulator to perform co-simulation of communication and mobility. Veins provides models for vehicular communication (e.g., IEEE 802.11p/ITS-G5) and realistic traffic patterns (from SUMO), allowing researchers to evaluate inter-vehicular communication (VANET) protocols and cooperative driving applications in an integrated environment[43].

3.4.2 SUMO

"SUMO (Simulation of Urban Mobility) is an open source, highly portable, microscopic and continuous traffic simulation package designed to handle large networks"[44]. Developed by the German Aerospace Center (DLR), SUMO models individual vehicle movements on detailed road networks with continuous space and time. It supports multi-modal scenarios (cars, trucks, buses, pedestrians) and includes tools for generating road layouts and traffic demand.

SUMO is widely used in transportation research for large-scale traffic analysis and for coupling with network simulators like OMNeT++ in vehicular network studies[44].

3.4.3 Truffle

Truffle is a development framework for Ethereum and other EVM-based blockchains that provides tools for compiling, testing, and deploying smart contracts [45]. It is described as "a world class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier" [45]. Truffle integrates contract compilation and linking, automated migration scripts, a built-in JavaScript console, and support for writing and running tests, thereby streamlining the smart-contract development lifecycle.

3.4.4 Ganache

"Ganache is a personal blockchain for rapid Ethereum and Filecoin distributed application development"[46]. It runs a local, in-memory Ethereum network that developers can use throughout the development cycle. Ganache is available as both a desktop application (with a rich GUI) and a CLI tool, and it provides features like instant mining, account management, and the ability to manipulate blockchain time. By simulating a deterministic local blockchain, Ganache allows safe testing and debugging of smart contracts before deploying them to public networks[46].

3.5 Summary

This chapter provided the theoretical foundation necessary to understand and implement the proposed authentication scheme for vehicular networks. It began by outlining the architecture and operational characteristics of VANETs, emphasizing their high mobility, dynamic topology, and real-time communication constraints [8,31]. It then explored the fundamental principles of three major authentication mechanisms: cryptographic protocols, public key infrastruc-

ture (PKI), and blockchain-based approaches [33,35]. The Kerberos authentication protocol was reviewed both in its original form and in its adaptations for constrained and dynamic environments such as IoT and VANETs [8,36,39,40]. This included an overview of its key components (AS, TGS), message flows, and built-in security features. Finally, the chapter introduced the simulation and development tools used throughout the dissertation, including OMNeT++, INET, VEINS, SUMO for VANET modeling, and Truffle and Ganache for blockchain prototyping [41–46]. Together, these components establish a comprehensive foundation for the design and evaluation of secure authentication in VANET environments.

Chapter 4

A Design of the Kerberos-Blockchain Authentication Framework

In this chapter, I present the proposed design of the Kerberos-Blockchain-based authentication framework, as outlined in Publication 1 [4]. I begin by introducing the overall goals and motivations behind integrating Kerberos and blockchain for VANET authentication. The chapter then explains the architectural structure and the technical components of the framework, including how blockchain enhances Kerberos and how the framework addresses handover latency and scalability issues. I also describe the evaluation setup and present performance results followed by an analytical discussion.

4.1 Introduction

VANET acts as the foundational network enabling communication between vehicles and infrastructure. This communication delivers critical information such as speed, location, direction, and emergency alerts, contributing to improved traffic efficiency, emergency response, and evacuation management. However, due to its dynamic and decentralized nature, VANET is highly vulnerable to security threats, necessitating fast and reliable authentication mechanisms. Kerberos offers strong security by using session keys, but its centralized structure can cause delays and complications during handover processes. On the other hand, blockchain provides a tamper-resistant, decentralized solution and has been widely explored for authentication in vehicular environments. Nonetheless, existing methods still face challenges related to performance efficiency and security, particularly in key exchange processes and managing network latency.

This research proposes a novel authentication protocol that combines Kerberos authentication with blockchain technology. The novelty lies in utilizing blockchain's decentralized characteristic to store authentication messages. By leveraging blockchain, it can reduce the handover time through its decentral-

ized characteristic. If the vehicle finishes the initial authentication stage, it does not need to connect the Kerberos server in the handover phase. Due to the authentication messages are already stored in a distributed and secure manner in the blockchain system. Once authenticated, it only has to allow a request RSU to call the authentication messages in the blockchain.

The contributions of this chapter are listed below:

- It introduces a novel authentication protocol that integrates Kerberos authentication with blockchain technology for VANETs.
- It leverages the decentralized nature of blockchain to securely store authentication messages, eliminating the need for centralized retrieval.
- It evaluates the authentication delay and signaling overhead to compute the performance of the system.

4.2 Proposed Method

This chapter explains the proposed authentication system, structured as a series of system stages in VANET. It includes the proposed system model, initialization, registration, and authentication phases.

4.2.1 Proposed System Model

This section describes our proposed system, which consists of its entities and stages. Our method combines Kerberos authentication with the blockchain system to store the authentication messages of the vehicle. By utilizing blockchain, If the vehicle finishes the initial authentication stage, then in the handover phase the vehicle does not need to connect Kerberos server. This is because the authentication messages are already stored in a distributed and secure manner in the blockchain system. It will make the time will be more efficient. Once authenticated, it only has to request RSU to call the authentication messages.

The illustration of the system is shown in Figure 4.1. It includes vehicles, RSUs, blockchain network, and Kerberos server as TAS. The RSUs and TAS

will join the blockchain network that uses Ethereum framework to enable the efficiency and scalability of other blockchain frameworks. The TAS consists of Authentication Server (AS) and Ticket Granting Server (TGS) subsystems.

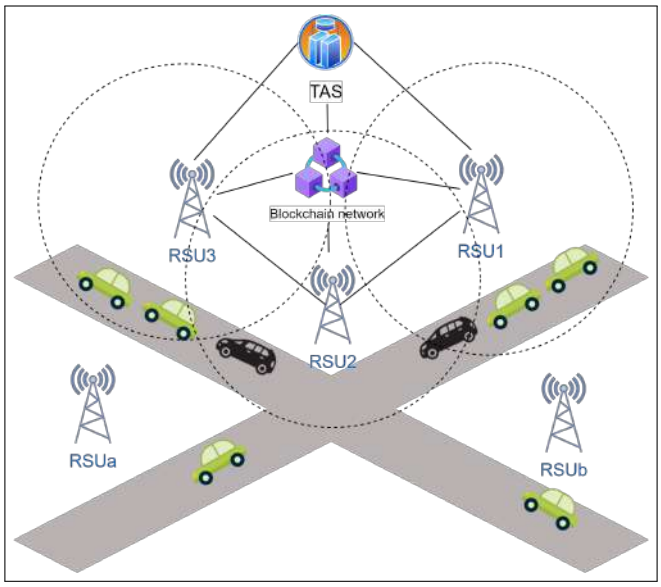


Figure 4.1: Illustration of the proposed system.

The stages in this proposed scheme include system initialization, registration phase, generation and sending of entities' key, authentication phase, and handover phase. Blockchain will store the authentication message in the initial authentication phase. During the handover process, the RSU will check the authentication messages, and the vehicle does not need to connect to the TAS. The proposed method in this paper will called KBC (Kerberos Blockchain). I use the Ethereum framework to store the authentication message in the blockchain. It provides a decentralized and secure platform for managing trust and reputation. In our implementation, we run Ganache CLI in 'fork' mode to clone the Ethereum Mainnet state at a chosen block height, creating a single locally controlled node that inherits real contract code and balances without spending real Ethereum. It means it only contains one physical nodes that only one Ganache process is active, acting as a sole P2P node form consensus, mining, and ledger updates. Although only one physical node is active, we simulate multiple logical participants by assigning its accounts (RSUs and TAS). In this case we will utilize four RSUs and one TAS. These accounts share the same blockchain state but allow us to simulate different entities without

running separate servers. This approach delivers full Ethereum fidelity and instant mining while maintaining a permissioned, reproducible environment.

4.2.2 System Initialization and Registration Phase

Offline registration involves the submission of credentials from the vehicles, including their vehicle identification number, password, source, destination, service type, and roles (read, write, and modify). RSUs also are registered with their location, MAC address, IP address, and RSU ID. Before beginning the authentication phase, the Kerberos server will generate and send the keys for every entity. The registration phase is shown in Figure 4.2.

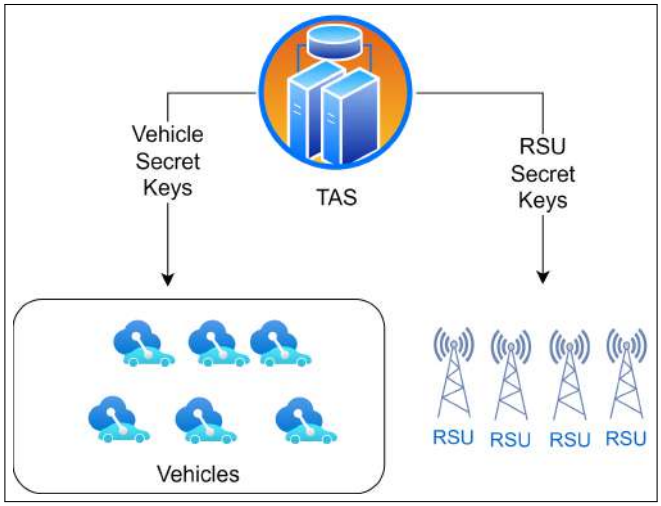


Figure 4.2: Registration phase.

4.2.3 Authentication Phase

The authentication phase protocol consists of several stages. I classify it into several big stages including the vehicle and AS communication, the vehicle, and TGS communication, the stage for storing authentication messages to the blockchain, and the vehicle and RSU communication stages. The signaling process of this authentication phase is shown in Figure 4.3.

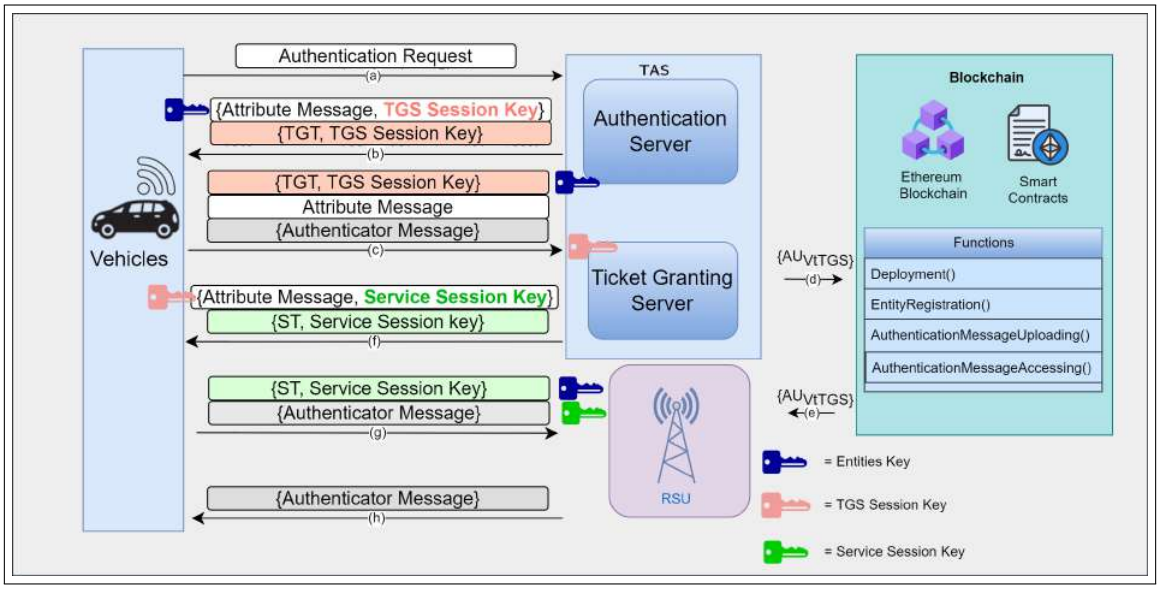


Figure 4.3: Proposed authentication scheme.

4.2.3.1 Vehicle and AS Communication Stage

This stage includes messages sent from the vehicle to AS and vice versa, showed in steps (a) and (b) in Figure 4.3. Firstly, the vehicle sends the RA to the AS. This RA includes the ID_v, the desired service name the vehicle wants to access (in this case, the service name is RSU service), IP_v, and the requested lifetime for TGT (Req_{TGT}). Req_{TGT} can finite the lifetime to make the system more secure by its limited time. That information will be sent to the authentication server in TAS. Equation (7.1) shows messages the vehicle sends to AS, and (6.2) and (6.3) show the reply messages. One group of messages that are not encrypted is shown in the parenthesis symbol. Another group of messages that are encrypted is shown in a bracket symbol. The encrypted key name is shown before the bracket.

$$RA = (ID_v||ID_s||IP_v||Req_{LT}). \tag{4.1}$$

$$\begin{aligned} AT_{AStV} =& K_{vs}[ID_{TGS}||TS_{AStV}|| \\ & LT_{TGT}||K_{TGSse}]. \end{aligned} \tag{4.2}$$

$$TGT = K_{TGSs}[(ID_v||ID_{TGS}||TS_{TGT}|| \\ IP_v||LT_{TGT}||K_{TGSse})]. \tag{4.3}$$

The AS has a list of registered users and their keys. It will check whether the ID_v and the messages are in that list. If correct, a copy of that K_{Vs} will be taken. After that, the AS will generate an AT_{AStV}, and the TGT will be sent to the user. The AT_{AStV} contains ID_{TGS}, timestamp TS_{AStV} and lifetime. TGT will contain ID_v, ID_{TGS}, TS_{TGT}, IP_v and LT_{TGT}. Both of those messages will be encrypted by the K_{TGSse} which is a randomly generated symmetric key that the user will use to decrypt several messages from the TAS and RSU as a service server only for that time. Attribute message (AT_{VtTGS}) will be encrypted by K_{Vs}, and the TGT is encrypted with the K_{TGSs}. Those two messages are then sent from the AS to the vehicle.

4.2.3.2 Vehicle and TGS Communication Stage

The vehicle needs to decrypt AT_{AStV} by the K_{Vs}, and then it will get K_{TGSse}. Afterward, the vehicle will generate two messages. The first message consists of ID_S the vehicle wants to access and LT_{TGT}. The second message is the AU_{VtTGS}, which contains KS_{se} and ID_v that encrypted by K_{RSU} and the ID_v, TS and $Nonce$ that encrypted by the KS_{se}. This stage is shown in stage (c) in the Figure 4.3.

Besides those two generated messages, a vehicle will forward TGT that it got from AS. The vehicle then will send those three messages shown in (6.4), (7.2), and (7.3) to the TGS. TGS will check its own list about the ID_S in plaintext. If the IDS exists in the TGS server list, TGS will copy the K_{Ss}. In the TGT there is a K_{TGSse}, then TGS can use this K_{TGSse} to decrypt the AU_{VtTGS}.

$$TGT = K_{TGSs}[(ID_v||ID_{TGS}||TS_{TGT}|| \\ IP_v||LT_{TGT}||K_{TGSse})]. \tag{4.4}$$

$$AT_{VtTGS} = (ID_s||Req_{LT}). \tag{4.5}$$

$$AU_{VtTGS} = K_{RSU}[KS_{se}||ID_v]||KS_{se}[ID_v||TS||Nonce] \tag{4.6}$$

4.2.3.3 TGS and Blockchain Communication Stage

TGS will start validating the data inside those messages by asking the blockchain whether the AU_{VtTGS} ever exists. TGS will make sure that ID_v in the TGT and AU_{VtTGS} are the same compared with the timestamp. Kerberos algorithm is configured to give a two-minute time limit. It will also compare between IP address inside TGT to the TGT of the user. It will also check whether the TGT is expired or not. If everything is okay, TGS maintains a cache from the user's recently received AU_{VtTGS}. To avoid a replay attack, TGS will check to ensure the AU_{VtTGS}, which was just received, is never saved in the cache. If the authentication is not in the cache yet, then TGS will add it. Then, TGS will store that AU_{VtTGS} message on the blockchain.

To ensure that the communications are safely saved and accessible by authorized users, TGS will store them in the blockchain in stages. The creation of a transaction is the initial step. The network nodes then confirmed the transaction. The nodes confirm the sender's digital signature and determine whether the sender has sufficient resources to deliver the message to determine whether the transaction is genuine.

The transaction is added to a block after it has been verified. After that, the block is broadcast to the network for validation. The nodes verify the digital signature of who created the block, and the authenticity of the transactions in the block is also examined. The block is added to the blockchain when it has been confirmed. A consensus process adds blocks to the blockchain, ensuring that every node in the network accepts the block's validity. This stage is shown in step (d) and (e) in Figure 4.3.

$$AU_{VtTGS} = K_{RSU}[KS_{se}||ID_v]||KS_{se}[ID_v||TS||Nonce] \tag{4.7}$$

$$ST = K_{Ss}[(ST||ID_v||ID_S||TS_{TGStV}||IP_v||LT_{ST}||K_{Sse})]. \quad (4.8)$$

After successfully storing the AU_{VtTGS}, TGS will then generate its messages in (4.7) and (4.8) to send to the vehicle. The first message is AU_{VtTGS}. The second message contains ID_v, service ID, timestamp, user IP address, and the lifetime for the service ticket. TGS then generates a random symmetric K_{TGSse} that will be encrypt both messages. Those two messages will then be sent from the TGS server to the vehicle. This stage is shown in step (f) in Figure 4.3.

4.2.3.4 Vehicle and RSU Communication Stage

After receiving those messages, it will move to stages (g) and (h). Vehicles will decrypt the AT_{TGStV} with KT_{GSse}, and then it will get the K_{Sse}. The vehicle will generate new AU_{VtS} which contains ID_v and TS_{VtS} and encrypt that message with the K_{Sse}. The vehicle forwards the ST and the authentication message to the server for this. In this case, the RSU is the service that vehicles want to use. The sent messages from the vehicle to the RSU are shown as follows:

$$ST = K_{Ss}[(ST||ID_v||ID_S||TS_{TGStV}||IP_v||LT_{ST}||K_{Sse})]. \quad (4.9)$$

$$AU_{VtS} = K_{Sse}[ID_v||TS_{VtS}]. \quad (4.10)$$

$$AU_{Sm} = K_{Sse}[ID_v||TS_{StV}]. \quad (4.11)$$

The RSU will do a similar process with the TGS. RSU will decrypt the ST with its K_{Ss}, and get access from the K_{Sse}. The RSU will use it to decrypt AU_{VtS}. If it is finished, it will make its own AU_{Sm} that contains ID_{RSU} and the TS_{StV} like in (7.6).

Afterward, that AU_{Sm} will be sent to the user, and the vehicle will decrypt with the K_{Sse}. Now the vehicle will verify whether the service name inside the authentication is the name it wants. To avoid a reply attack, the vehicle

will also check the TS_{StV} to ensure the authentication is new. The vehicle also maintains its cache. As the user has already mutually authenticated the service, it will cache a copy of the encrypted service ticket for the future.

4.2.4 Handover Phase

The handover signaling diagram of the proposed method is shown in Figure 4.4. When a prior RSU identifies that a vehicle or group of vehicles has moved out of its coverage area, a handover process is triggered.

- Step 1: The vehicle send the VID-targetRSU that verifies the target RSU has authenticity. The source RSU will determine the authenticity of the destination RSU by checking the neighbor table. In the initial authentication phase, the vehicle already received service ticket for RSU with the single session key K_{Sse} that valid for all RSU in the network. This service ticket and this K_{Sse} then will be uploaded in the network.

- Step 2: Before sending the request, that vehicle use same K_{Sse} that vehicle get from initial authentication phase for encrypt Authentication-MessageUploading() to blockchain. The vehicle advances by transmitting a request message to the destination RSU. This RSU will send the request for the **AuthenticationMessageUploading()** transaction to the blockchain ledger.

- Step 3: The blockchain ledger will check the RSU with the smart contract agreements, especially for the entity name. After that, it will give the AU_sm to the destination RSU. Because all the RSU share identic K_{Sse}, so smart contract just need to check the authentication message. The destination RSU then can decrypt directly service ticket without connect to the TAS and get same K_{Sse}.

- Step 4: That RSU will equalize the authentication message sent by the vehicle and the authentication message in the blockchain ledger. Following successful validation, the destination RSU sends a message confirming the completion of the handover to the vehicle, thereby finalizing the transfer. Subsequently, the vehicle updates its ledger and disseminates

the information. Session key K_{Sse} is released only one time for service and valid for all RSU.

The handover signaling diagram of the proposed method is shown in Figure 4.4.

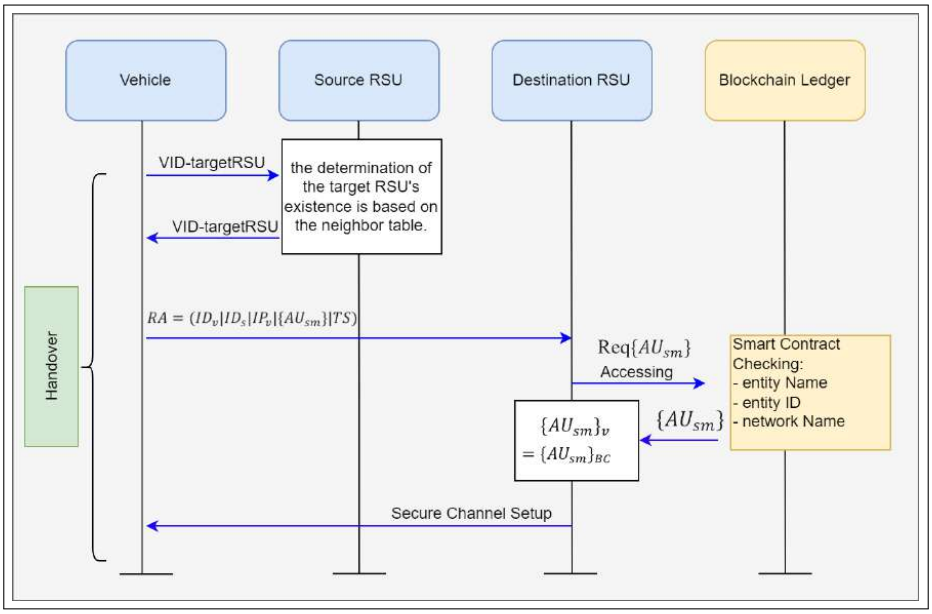


Figure 4.4: Handover signaling diagram.

4.3 Evaluation and Result Analysis

I analyze several parameters in this evaluation, including authentication delay and the signaling overhead.

4.3.1 Implementation Environments

To build the designed system, some networking tools are used to model the communication process and protocol among the vehicular network entities. The main tool used is OMNeT++, which models and simulates any discrete system approach. The vehicular network in OMNeT++ was built by the NED language. It worked with other frameworks, including VEINS and INET. Afterward, the vehicle nodes are also built, and the maps, junctions, and

routes are set up using the SUMO. The parameters are shown in Table. 4.1. I configured the stage's parameters, such as location, speed, and routes. The map that I used is a map near Okayama University, Tsushima, Japan area map. The map's scale is 1:20,000. In the first stage, I created all the entities that function as a Kerberos server, including 100 vehicles, 4 RSUs, and 1 TAS.

Table 4.1: Simulation parameters.

Parameter	**Value / Unit**
Communication model	IEEE 802.11p
Simulation area	5000 m × 2500 m
Communication range of RSU	1000 m
Communication range of vehicle	100 m
Data rate	27 Mbps
Safety messages	Every 30s
Mobility	Duaroute Mobility
Blockchain platform	Ganache CLI in fork mode (forked from Ethereum Mainnet)
Physical P2P nodes	1 (single Ganache process handling all consensus and mining)
Logical participants	4 RSUs, 1 TAS

The blockchain is also built to store the authentication messages. The blockchain setup runs all blockchain experiments on Ganache CLI with Truffle framework in fork mode, cloning Ethereum Mainnet state at a chosen block height. It provides a single, local Ethereum node that inherits real contracts, balances, and storage from Mainnet, while offering full control over mining and block times. Although only one physical node is active, we leverage its multiple unlocked accounts that assigning each to a distinct role including RSU and TAS to simulate a multi-party network without additional servers. This configuration lets us test our smart-contract logic against authentic Mainnet state in a fully deterministic, permissioned environment, with logical separation of roles but without the overhead of a true multi-host P2P deployment.

4.3.2 Authentication Delay

The authentication delay refers to the sum of the messages' transmission delay and message processing time. It is calculated by equation (6.10) where

T^i_{SRA} is the time for a vehicle to send the authentication request. T^i_{RFA} is time for vehicle i receive service authentication message from RSU. N denotes the total number of vehicles in the simulation. The calculation of the authentication delay is as follows:

$$\text{AuthDelay} = \frac{1}{N}\sum_{i=1}^{N}(T^i_{SRA} \ - \ T^i_{RFA}). \tag{4.12}$$

To evaluate, I set up 20 experiments with several variations from the number of vehicles in the set-up environment. It gradually increases by 5 vehicles in every experiment from 5 to 100. According to the result, the authentication delay gradually rises with the surge in vehicles. It rises from 63.57 ms for the lowest number of vehicles to 94.73 ms for 100 vehicles. This is caused by the number of queues of the service inside the TAS increased, as I only have one TAS. However, this still can meet the requirements of the maximum authentication delay. The authentication delay should be less than 100 ms to have an efficient mechanism in VANET before broadcasting a new safety message [17].

4.3.3 Signaling Overhead

The signaling overhead parameter counts the signaling messages as the cost per unit of time vehicle involved in handover. This parameter is calculated by the multiplication of the RSUs distance, unit transmission overhead, and message size in vehicular communication. The computation procedures of signaling overhead in DMM are shown in the following equation [47].

$$\begin{aligned} C_{DMM} =& K[a(L_{Rs} + L_{RA})hop_{(v-RSU)} + \\ & 2b(L_{PBA} + L_{PBU})hop_{(CMD-RSU)} \end{aligned} \tag{4.13}$$

The signaling overhead of SEBGMM for K Vs' handover is shown as follows [48].

$$C_{SEGBMM} = K[a(L_{Rs} + L_{RA})hop_{(v-RSU)}] \tag{4.14}$$

The signaling overhead for this proposed system (KBC) is shown as follows.

$$C_{KBC} = hop_{RSU-RSU}[a * Trans_u * L_{msg}] \tag{4.15}$$

Let the $hop_{RSU-RSU}$ denote the average distance from one RSU to another RSU, a denotes the weighting factor for a link, $Trans_u$ denotes the unit transmission, and L_{msg} denotes the size of the total message that goes and forth in the signaling process. The size of the messages chosen in this paper is shown in Table. 4.2, and the comparison result with previous work is shown in Figure 4.5. All the schemes used blockchain-based security in VANETs. The first scheme (DMM) [47] has the heaviest signaling overhead due to its architecture that does not consider session key negotiation and the initial authentication phase of nodes. Our proposed scheme and SEBGMM [48] have similarities in distinguishing the initial authentication and handover phases. However, the proposed scheme has the smallest signaling overhead due to its simpler architecture that does not utilize a control mobility database (CMD) and only utilizes the TAS and RSU.

Table 4.2: The size of parameter.

Parameters	Value
Session Key	16 byte
Vehicle ID	8 byte
TS	4 byte
Ticket for initializing authentication	8 byte
HMAC	8 byte
input bit length AES	16 byte
Lifetime	3 byte
IP Address	16 byte
Service Name	3 byte

4.4 Summary

Herein, I proposed a Kerberos authentication algorithm that combined blockchain technology to authenticate vehicles in the V2I environment. The

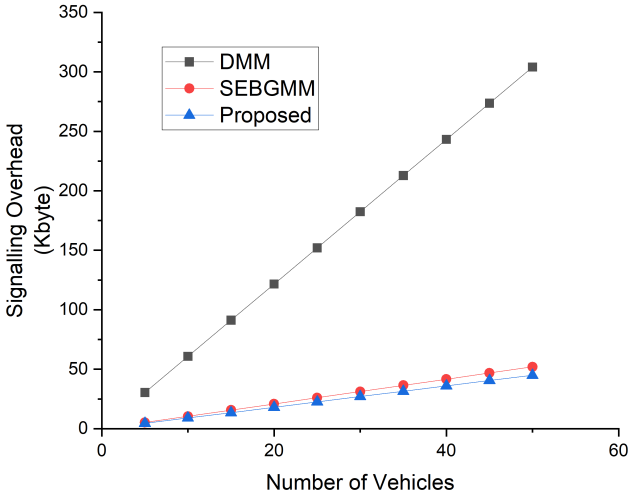


Figure 4.5: Signaling overhead.

proposed authentication protocol uses blockchain to store authentication messages to reduce the process in the handover phase in VANETs after the initial authentication phase. The system meets the minimum authentication delay requirement that should be below the specified threshold of 100 ms. The authentication delay increases as the number of vehicles. The signaling overhead is lightweight due to its simple architecture and the handover authentication that no longer needs to reconnect to the TAS after initial authentication. The future work in this research is creating a controller plane to detect the malicious nodes, overcoming pass-the-hash attacks, and implementing privacy-preserving authentication to develop anonymous authentication.

Chapter 5

An In-Depth Analysis of Kerberos and Blockchain Integration on VANETs' Security and Performance

In this chapter, I provide a comprehensive analysis of the integration of Kerberos and blockchain within VANET environments. While our previous work evaluated the off-chain performance metrics such as authentication delay and signaling overhead, this study focuses on assessing the blockchain (on-chain) behavior, including gas consumption, block size, and security robustness. The primary contributions in this chapter have been published in the ICCE-TW conference [6].

5.1 Introduction

Previously, I proposed an authentication system for VANETs that combines blockchain technology and the Kerberos protocol[4]. In this chapter, I concerned about the on-chain environment that investigates the blockchain's performance. Specifically, I consider a scenario with 100 vehicles, 4 RSUs, and 1 TAS. To conduct our evaluation, I employ the Ethereum blockchain platform and the OMNeT++ simulation framework. The Truffle framework and Ganache software were used to measure the gas consumption of blockchain operations, focusing on varying block sizes. The results confirmed the blockchain's effectiveness in supporting VANET authentication scenarios, meeting transaction cost requirements.

The contributions of this chapter are listed below:

- It integrates Blockchain and Kerberos for VANET authentication. This research builds upon a previously proposed system that innovatively integrates Kerberos authentication with blockchain technology for securing VANET communications.

- It evaluate blockchain performance in the on-chain environment by assess the gas used and the block size.
- It confirm the blockchain feasibility for the implementation of the VANET.

5.2 Review of the System

This section revisits the architecture and workflow of the Kerberos-Blockchain authentication framework discussed in Chapter 4.

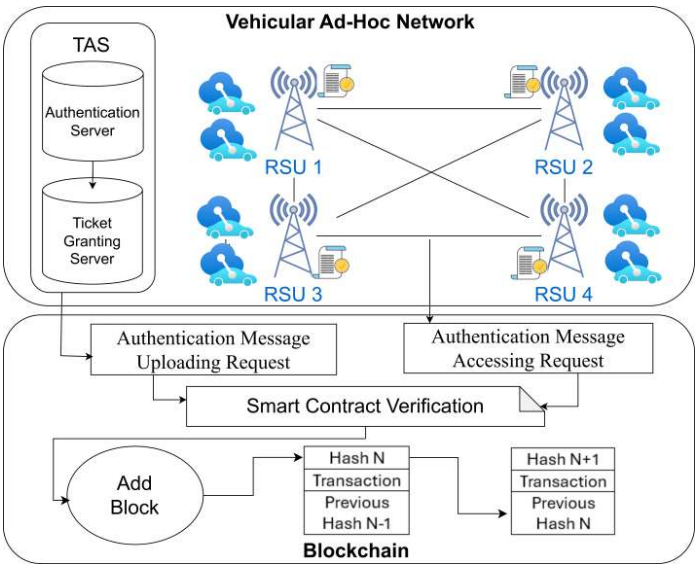


Figure 5.1: Kerberos-Blockchain VANETs system.

This paper discusses the off-chain environment that is concerned with the blockchain part. The proposed system of this work is summarized in Figure 5.1. The setup of the system consists of TAS, RSUs, and vehicles. TAS is a Kerberos server that is responsible for verifying and validating the authenticity of the vehicles. TAS will create an authentication message (AU_{VtTGS}) which contains KS_{se} and ID_v that encrypted by K_{RSU} and the ID_v, TS and $Nonce$ that encrypted by the KS_{se}.

$$AU_{VtTGS} = K_{RSU}[KS_{se}||ID_v]||KS_{se}[ID_v||TS||Nonce] \tag{5.1}$$

Utilizing blockchain to store the authentication messages by TAS and access

by RSU, can diminish handover time as the vehicles only need to connect with RSU, instead of TAS.

In blockchain, we implement two key Solidity functions including **AuthenticationMessageUploading()** and **AuthenticationMessageAccessing()** to manage handover credentials. When the TAS invokes **AuthenticationMessageUploading()**, it submits the Kerberos AP_REQ payload (AU_{VtTGS}) as an argument; the smart contract enforces the access control rules, and once consensus is reached, the transaction is committed to a new block. Later, each RSU invokes **AuthenticationMessageAccessing()** to retrieve AU_{VtTGS}; after satisfying the contract's validation logic, the payload is returned for decryption and verification.

For our experiments, this entire blockchain layer runs on Ganache CLI in fork mode, which clones Ethereum Mainnet state at a specified block height. This provides a single, locally controlled node that inherits real contract code and account balances without spending real ETH. Although only one physical node (one Ganache process) handles consensus, mining, and ledger updates, we simulate multiple logical participants—four RSUs and one TAS—by assigning its unlocked accounts to distinct roles. These accounts share the same blockchain state but allow us to model different entities without separate servers. This setup delivers full Ethereum fidelity with instant mining in a permissioned, reproducible environment.

5.3 Evaluation

The evaluation setup involved a simulated vehicle network in the suburban Tsushima area of Okayama, Japan. In a typical suburban environment with a line-of-sight scenario, the communication range of an RSU can reach around $1km$[16]. The simulation area covered $5000m \times 2500m$. It included 100 simulated vehicles, 4 RSUs, and 1 TAS. The system combined two approaches. The Ethereum blockchain handled on-chain tasks like executing smart contracts and recording transactions. Meanwhile, the OMNeT++ simulation framework managed off-chain tasks such as simulating vehicle movements, communication, and data processing. The evaluation focused on two main aspects. First, it analyzed the gas consumption of on-chain operations,

specifically how block sizes affected gas usage. Second, it performed a security analysis to assess the strength of the authentication system against potential threats.

5.3.1 Blockchain's Performance

This section evaluates the blockchain's efficiency in recording authentication-related transactions, focusing on two primary metrics: gas consumption and block size.

Table 5.1 presents the gas value, representing the unit quantity of work completed in each blockchain transaction. The gas value is calculated for four smart contract operations, among which **Deployment** and **AuthenticationMessageUploading()** depend on the number of authentication messages. Since each vehicle contributes one authentication message, the gas consumption for these operations increases proportionally with the number of vehicles, as shown in Figure 5.2. The corresponding gas used is 1161682 GWEI for one vehicle and 2211216 GWEI for 100 vehicles. GWEI is a subunit of the Ethereum blockchain's gas value, where 1 GWEI = 10^{-9} Ether. The transaction cost in Ether is obtained by converting the gas value using Equation 5.2, and subsequently, the cost in USD is calculated by multiplying the Ether value by the exchange rate at the time of data collection, as expressed in Equation 5.3. The exchange rate on 10 February 2024 was $P_{\text{Ether}} = 2487.23$ USD per Ether. The analysis confirms that the maximum gas usage remains below the block gas limit of 6721975 GWEI, ensuring the proposed strategy is feasible and practical.

$$\text{Tx Cost (Ether)} = \frac{\text{Gas Used (GWEI)}}{10^9} \tag{5.2}$$

$$\text{Tx Cost (USD)} = \text{Tx Cost (Ether)} \times P_{Ether} \tag{5.3}$$

The block's size and transaction data are dynamic and depend on its gas limit [49]. The experimental result shows that the block's size increases by the number of vehicles from 10,386 bytes for 1 vehicle and 20,094 bytes for 100 vehicles.

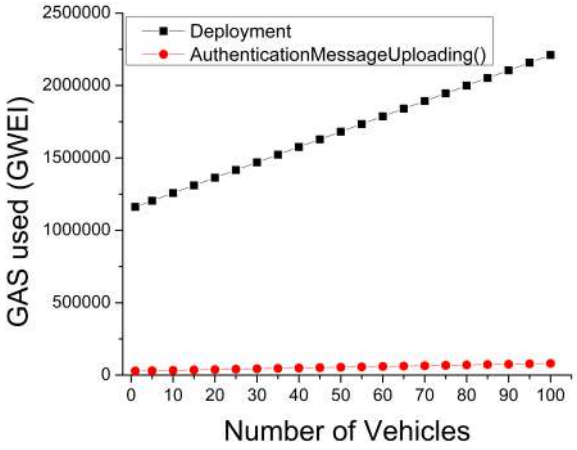


Figure 5.2: Number of Vehicles vs GAS values.

Table 5.1: The gas used for the smart contract operations.

Operation	**Gas used (GWEI)**	**Tx cost (Ether)**	**Tx cost (USD)**
Deployment	1145042	0.001145	2.8478
EntityRegistration()	112223	0.000112	0.2785
AuthenticationMessageUploading()	29733	0.000029	0.0721
AuthenticationMessageAccessing()	27208	0.000027	0.0671

* *1 GWEI* $= 10^{-9}$ *Ether*, *1 Ether* = 2487.23 USD on 10 February 2024.

5.3.2 Security Analysis

Blockchain and smart contract integrating with VANETs to ensure the several issues. Firstly, The Ethereum blockchain uses cryptographic hash functions to ensure that authentication messages are immutable when records are added to the blockchain such as collision resistance and irreversibility. It contribute to the prevention of authentication message tampering or deletion. Secondly, smart contracts enable transaction verification by creating agreements for each entity to fulfill, ensuring only authorized entities can access the authentication message. It reduces the risk of data exposure to unauthorized parties and contributing to system robustness. Thirdly, the verification process for adding new blocks involves all network entities, ensuring collective responsibility for transaction execution and trust among participants. This distributed validation, along with Proof of Work (PoW), mechanism contributes to network transparency.

5.4 Summary

In this second study, I conducted the evaluation of blockchain's performance and security analysis in our proposed authentication system for VANETs, employing both blockchain and Kerberos. Using Ethereum blockchain and OMNeT++ simulation, the evaluation demonstrated the practicality and met implementation requirements regarding transaction costs.

Chapter 6

The Design and Implementation of Kerberos-Blockchain Vehicular Ad-Hoc Networks Authentication Across Diverse Network Scenarios

In this chapter, I present the implementation and evaluation of the Kerberos-Blockchain-based authentication system across various VANET scenarios, as outlined in Publication 3. I begin by introducing the design overview, entities, and the multiple stages involved in the proposed framework. Following that, the chapter describes specific implementation scenarios, including initial registration, authentication, and handover phases. I then evaluate the system's performance from network and blockchain perspectives and provide a comparative analysis with other schemes.

6.1 Introduction

In our proposal, I integrate Kerberos authentication and blockchain to conduct the innovational authentication system for VANETs. This approach stores Kerberos authentication messages in the blockchain's distributed ledger that can be accessed in the Trusted Authentication Server (TAS) and Road Side Units (RSUs). This authentication message storing aims to simplify handover delay processes, shorten authentication delay, and securely keep the authentication message. To further improve system performance, I implement Kerberos using AES-128 encryption instead of the original Data Encryption Standard 77 (DES77) [28] aiming to reduce authentication time. Then, I assess the feasibility of blockchain technology for VANET authentication scenarios using Ethereum and simulate the process with OMNeT++.

To evaluate its effectiveness, I have designed three different scenarios. The first scenario involves simulating the system in a suburban environment

within the Tsushima Campus area at Okayama University, Japan. This simulation involves 100 vehicles and one TAS. The second scenario involves an urban environment in the Okayama Station area, featuring a higher vehicle density with 200 vehicles and one TAS. Then, the third scenario is a variation of the second but includes an additional TAS, totaling two TASs. In our evaluation, I focus on network performance metrics such as authentication delay, handover delay, and end-to-end delay. Additionally, I assess blockchain performance by measuring factors such as gas usage and the memory size of the blocks.

The contributions of this chapter are listed below:

- It proposes a system that enhances the effectiveness of the authentication protocol by storing the authentication message from Kerberos authentication in the blockchain system.
- It reduces the authentication delay by utilizing blockchain to access the authentication message, which can make the re-authentication process simpler than the initial authentication process.
- It evaluates the authentication delay, handover delay, and end-to-end delay to compute the performance of the system and investigate the effects of TAS on these delays.
- It evaluates the gas value and the memory size of the block to store the authentication message, which ensures the practicability of the system if implemented in the blockchain environment.
- It evaluates the proposed method in diverse networks, including the suburban area and the urban area.

6.2 The Proposed Method and Scenarios

This section elucidates the overview of the authentication system, organized as a sequence of system phases in VANETs. The components encompass the suggested system framework, commencement, enrollment, and authentication steps.

6.2.1 Overview of the System

This subsection delineates the outlines of the proposed system, comprising its entity functions, an overview of the steps, and an overview of the authentication and blockchain part. The methodology integrates Kerberos authentication with the blockchain framework for the preservation of vehicles' authentication messages [4]. By employing blockchain technology, a possible decrease in handover time can be achieved as a result of its decentralized characteristics. Following the vehicle's completion of the initial authentication phase, there is no need to establish a connection with the Kerberos server when transitioning to the handover phase. The functions of entities, an overview of the phases, and an explanation of the authentication part and blockchain part are presented in the following sub-sections.

6.2.1.1 Entities and Function

Our system comprises three fundamental entities: the Trusted Authentication Server (TAS), the Road Side Units (RSUs), and the vehicle. The TAS functions as a Kerberos server responsible for registering all entities, authenticating vehicles, and disseminating authentication messages to the blockchain. Within the TAS, there exist subordinate components, namely the Authentication Server (AS) and the Ticket Granting Server (TGS). RSUs serve as intermediaries between the vehicles and the TAS. They facilitate the integration of vehicles into the network, assist in the transfer of data among network entities, and disseminate information pertinent to traffic conditions. The vehicles assume the role of network nodes and data sources. Equipped with an On-Board Unit (OBU), they are capable of transmitting road-related information to RSUs and other proximate vehicles.

6.2.1.2 Overview of the Phases

The phases in the overall system include the system initialization and registration phase, initial authentication phase, and handover phase. The registration phase has a function to register all of the entities, including the RSU, TAS, and vehicles. The initial authentication phase and handover phase are

presented in Figure 6.1. In the initial authentication process, vehicles send the authentication request to the TAS, especially to the AS. If verified, the AS sends the ticket-granting ticket to the vehicles as a credential to obtain the Service Ticket from the TGS. Service ticket issued for RSU includes legitimate session key for all RSU because all RSU shared same session key. Vehicle will store this session key for next handover without re-contacting the TAS. With that service ticket, the vehicle is authenticated and can connect to the network through RSU1. When exiting the communication range of RSU1, the vehicle initiates a handover request to RSU2 by transmitting its authentication message. RSU2 has to check the credentials to match the data from the blockchain. RSU2 sends the authentication message request to the blockchain, and if it matches the vehicle will be re-authenticated. In this handover process, the vehicle does not need to perform the initial authentication phase anymore, which can reduce the authentication delay.

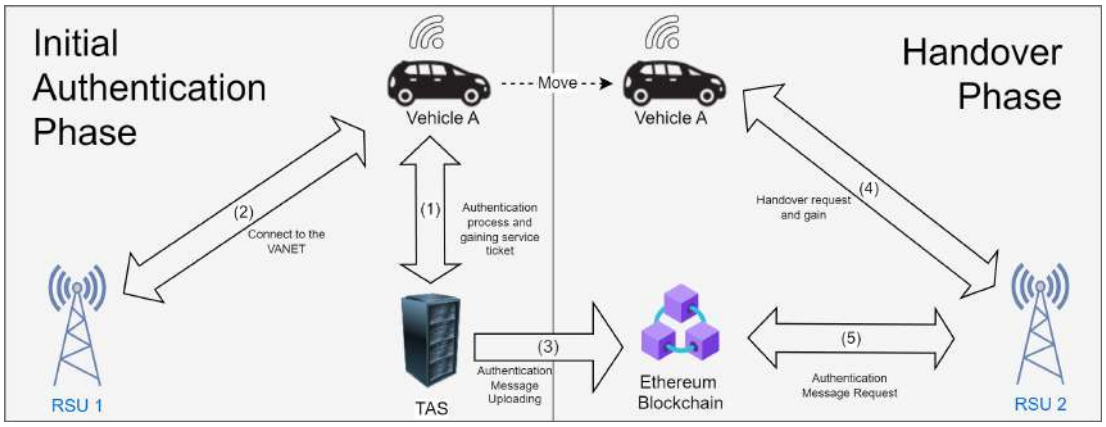


Figure 6.1: Resume of initial authentication phase and handover process.

6.2.1.3 Main Parts of the Kerberos-blockchain VANET System

This research includes two main parts: the authentication part and the blockchain part, as shown in Figure 6.2. The authentication part uses Kerberos authentication. The Kerberos server has two sub-servers that have different functions. The Authentication Server (AS) has the function of checking the vehicle's credentials and generating a TGT as the vehicle's credential to be able to obtain service tickets from the Ticket Granting Server (TGS). The TGS

will also generate an authentication message as the credential in the handover process that needs re-authentication for every vehicle that enters the new RSU communication range. That authentication message will be uploaded to the blockchain.

The blockchain part includes smart contract creation to upload and access that authentication messages. The agreements that I used to write down the solidity codes are entity name, entity ID, and network name. The entities include vehicles, RSU, and TAS. Within these solidity codes, I created four smart contract functions, including the deployment function, the entity registration, the authentication message uploading and, the authentication message uploading function. The deployment function is a function at the beginning to initialize the system. Entity registration is a function to register every entity in the system. Authentication message uploading is a function for uploading the authentication messages, and authentication message accessing is a function to access the authentication messages. In the authentication message uploading and accessing function, I need to know the entity name, which I create as a unique parameter. Based on this parameter, I set the entities that are able to upload the authentication messages as only the TAS. Then, the entity that can access the authentication messages is the RSU. After entering the input parameter, it will be verified by all the other parties in the network. After reaching a consensus, the transaction will be added as a new block in the blockchain.

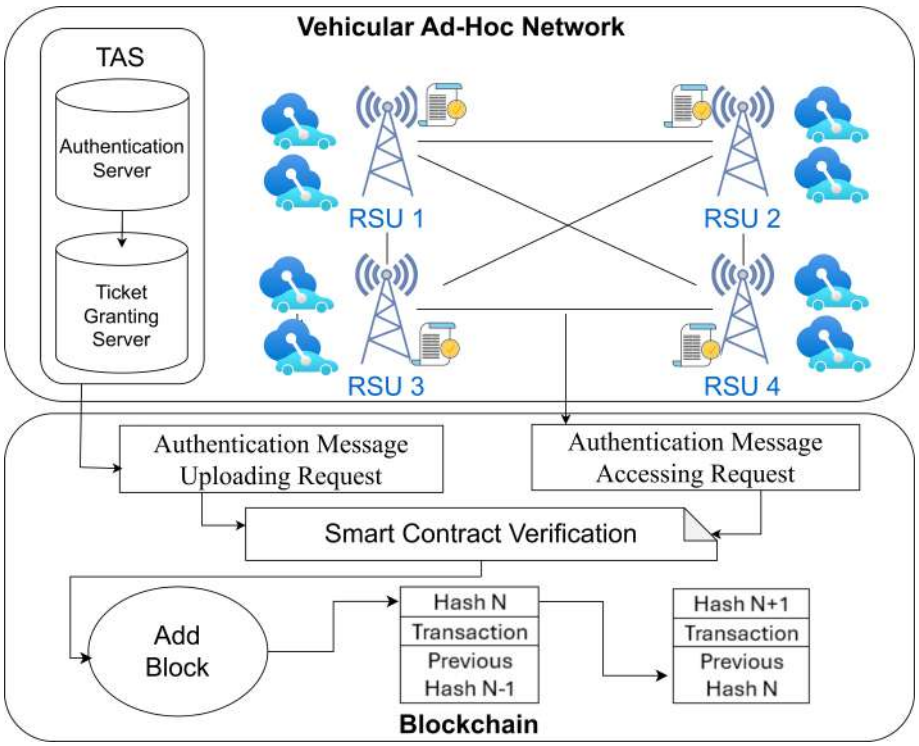


Figure 6.2: Main parts of the Kerberos-blockchain VANETs system.

To implement the system, I use Ethereum blockchain. Blockchain is particularly suitable for VANETs (Vehicular Ad-Hoc networks) for several reasons, primarily due to its robust and well-proven characteristics. There are several existing Distributed Ledger Technologies (DLTs) besides blockchain, such as Directed Acyclic Graph (DAG), Holochain, and Hashgraph [50]. While each of these DLTs offers innovative features, they come with disadvantages that make them less suitable in VANETs compared to blockchain. DAG is relatively newer and has shown vulnerabilities, such as potential attacks exploiting low transaction fees and double-spending. These are significant risks in a critical environment like VANETs. Holochain, Radix, and Corda are newer technologies and do not have the same level of standardization, while many blockchain platforms, especially Ethereum, have established standards and protocols that can be leveraged for VANET applications. While Corda supports smart contracts, it is more business-oriented, focusing on enterprise use cases rather than the real-time, high mobility of VANETs. The support for smart contracts in blockchain, especially in Ethereum, allows the creation of automated services such as vehicle coordination and cooperative agreements, fitting well with the

VANET's requirements. The ethereum use the standard version f3553dd [51]. There are several platforms of blockchain that have already been established and have matured standardization, such as Ethereum and Hyperledger [52]. This system utilizes the Ethereum blockchain rather than Hyperledger. Although it supports smart contracts, Hyperledger emphasizes chain code, which has limited flexibility compared to Ethereum's smart contracts, which have robust and flexible capabilities. It enables more complex vehicle interactions and autonomous services, essential in VANET systems.

For our experiments, we run Ganache CLI in fork mode, cloning Ethereum Mainnet state at a specified block height. This yields a single, locally controlled node that inherits real contract code and account balances without spending actual ETH. Although only one physical node handles consensus, mining, and ledger updates, we simulate multiple logical participants—four RSUs and one TAS—by assigning Ganache's unlocked accounts to distinct roles. These accounts share the same blockchain state but allow us to model different entities without separate servers. This setup delivers full Ethereum fidelity with instant mining in a permissioned, reproducible environment.

6.2.2 Testing Scenarios

Figure 6.3 shows three case scenarios for the experimentation of the system. The first scenario, as shown in Figure 6.3a, focuses on a suburban area. In this scenario, I utilized 100 vehicles, four Roadside Units (RSUs), and one Trusted Authentication Server (TAS). The limited number of RSUs is justified by the relatively low density of the suburban environment, which does not necessitate a large number of vehicles. The experiments were conducted using the Tsushima Campus area maps in Okayama, Japan, as shown in Figure 6.4a. The maps are generated by using Openstreet Maps, licensed under the Open Database License (ODbL) and inserted into SUMO tools to simulate the VANET. SUMO is distributed under the Eclipse Public License (EPL), which is a permissive open-source license. In the first scenario, the TAS is placed in the middle of the network, surrounded by RSU1, RSU2, RSU3, and RSU4.

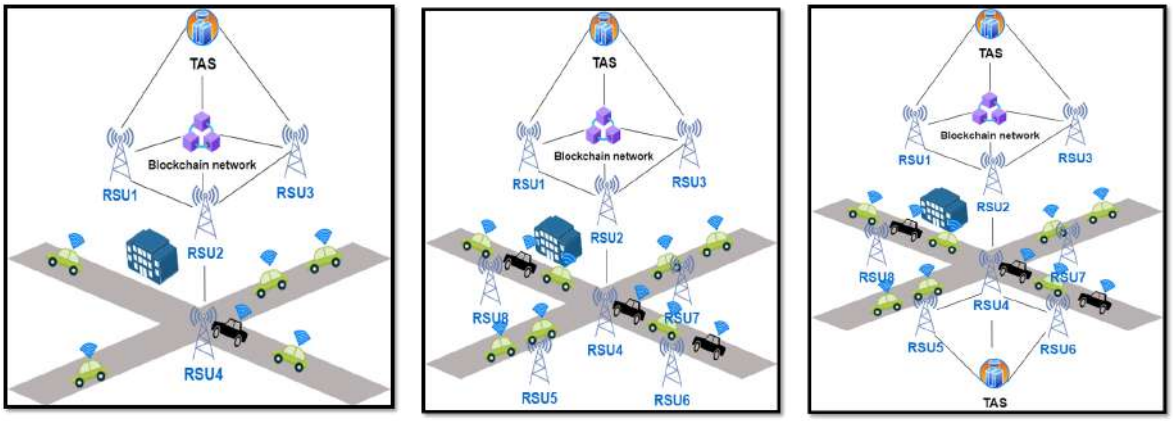


Figure 6.3: Experiment case scenarios: (**a**) suburban, (**b**) urban with 1 TAS, and (**c**) urban with 2 TASs.

The second scenario is an urban scenario with 1 TAS, as shown in Figure 6.3b, characterized by higher vehicle density. In this case, I increased the number of vehicles from 100 to 200. Given the greater availability of infrastructure in the city, I also doubled the number of Roadside Units (RSUs) from 4 to 8. This scenario was conducted in the densely populated area around Okayama Station in Okayama City, Japan, as highlighted in Figure 6.4b. The objective is to determine whether or not the performance of the authentication system remains within acceptable limits when the number of vehicles increases. Specifically, I aimed to assess if the authentication delay in the VANET environment stays below the critical threshold of 100 milliseconds, as stipulated in [17]. I placed the TAS in the centre of the maps, surrounded by the 8 RSUs. The small yellow dots in the maps indicate the vehicles.

The third scenario, depicted in Figure 6.3c, was also implemented using the urban area maps depicted in Figure 6.4c. In this case, I employed more than one Trusted Authentication Server (TAS) to ensure the system's performance in an environment requiring additional infrastructure. Specifically, I utilized 8 RSUs, 200 vehicles, and 2 TAS units. The introduction of multiple TAS units aimed to distribute the authentication management tasks effectively, accommodating the increased number of vehicles. In the third scenario, the TASs are positioned on the right and left sides of the map, unlike the central placement in scenarios 1 and 2, to achieve balanced load distribution.

This configuration is intended to prevent load imbalances that could negatively impact TAS performance.

The number of vehicles in a VANET system varies based on factors like area size, use case, and simulation goals. Denser urban areas have higher vehicle counts, while less populated areas have lower values. In one study [53], the authors chose 50–200 vehicles to evaluate routing protocols in a city scenario with high-density traffic. In a separate analysis [54], authors selected 0–120 vehicles to study urban traffic congestion and system effectiveness. In another study [55], authors chose 10–100 vehicles to simulate low and moderate traffic scenarios in real-world vehicular environments. This range helps analyze how increasing vehicle density impacts key performance metrics such as packet delivery ratio and delays. The selection of vehicle numbers for simulations depends on balancing real-world traffic patterns with available computational resources.

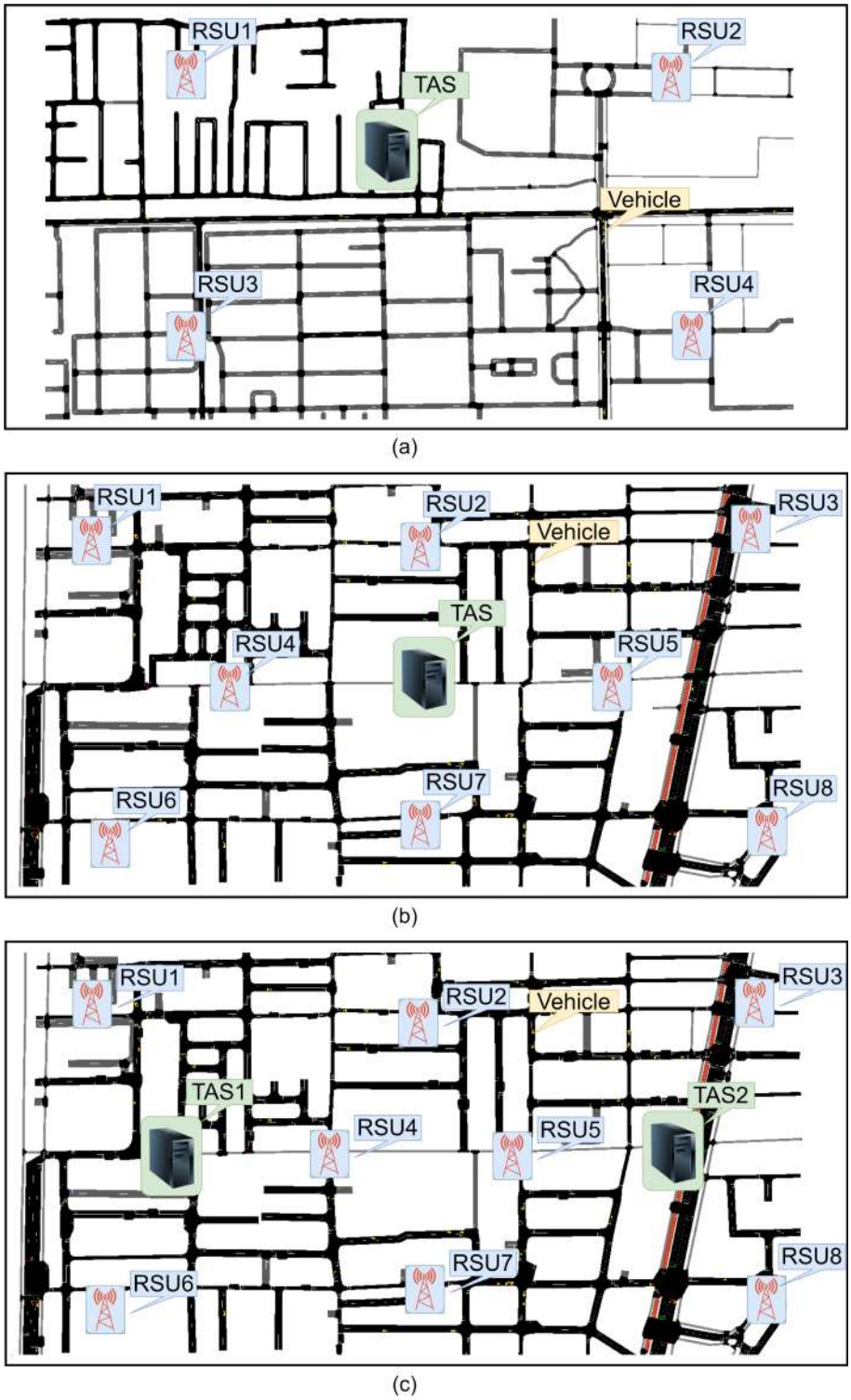


Figure 6.4: Maps for the scenario of (**a**) suburban and (**b**) urban with 1 TAS and (**c**) urban with 2 TASs.

6.2.3 System Initialization and Registration Phase

Offline enrollment comprises the provision of credentials from the vehicles, encompassing their unique vehicle identification number, user identification number, password, origin, destination, type of service, and permissions (read, write, and modify). Roadside Units (RSUs) are also enrolled with their specific location, MAC address, IP address, and RSU ID. Prior to commencing the verification process, the Kerberos server will produce and transmit the confidential keys for each entity.

6.2.4 Initial Authentication Stage

The protocol for the authentication phase comprises numerous distinct stages. These stages are delineated as the communication between the vehicle and the Authentication Server (AS), the communication between the vehicle and the TGS, the phase dedicated to storing authentication messages on the blockchain, and the communication stages between the vehicle and the RSU. The illustration of the signaling process during this authentication phase can be observed in Figure 6.5.

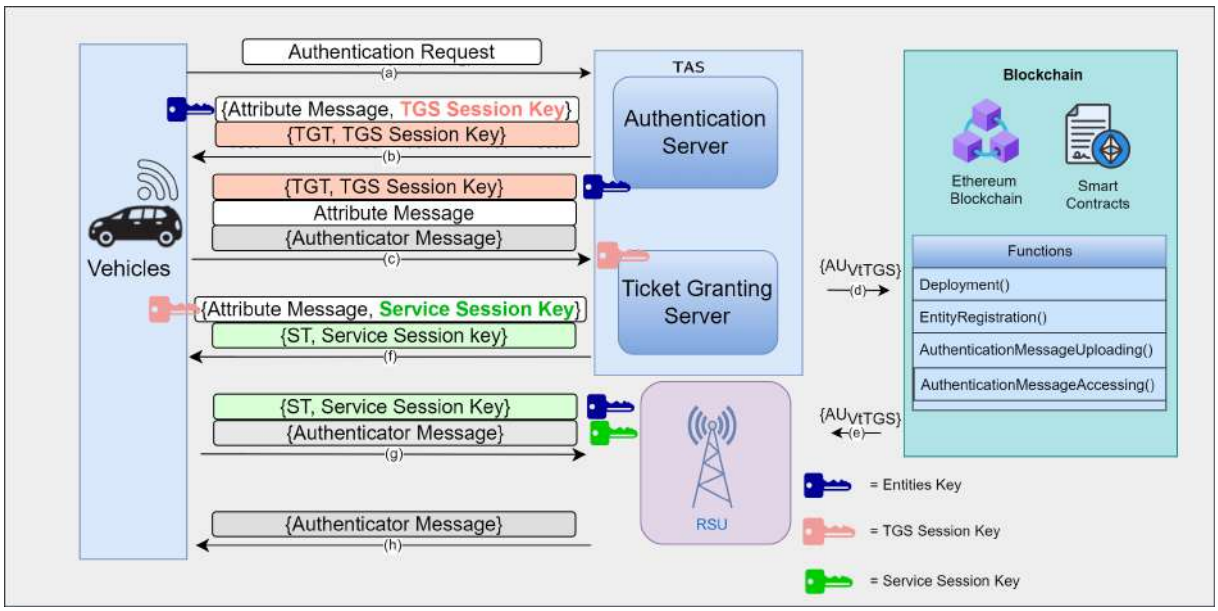


Figure 6.5: Initial authentication phase.

6.2.4.1 Vehicle and AS Communication Stage

This phase encompasses communications transmitted from the vehicle to the AS, and vice versa, mirroring steps (a) and (b) depicted in Figure 6.5. Initially, the vehicle transmits the Registration Acknowledgment (RA) to the AS, containing the ID_v, the specific service name that the vehicle intends to utilize (in this instance, the service name pertains to RSU service), IP_v, and the stipulated duration for the Ticket Granting Ticket (TGT) (Req_{TGT}). The Req_{TGT} serves to restrict the duration, thereby enhancing system security through a finite time constraint. These data will be forwarded to the AS in the TAS.

The messages that are sent by the vehicles to the AS are shown in Equation (7.1). Equations (6.2) and (6.3) show the corresponding response messages. The parenthesis symbol denotes one group of unencrypted messages, while the bracket symbol illustrates the encrypted messages.

$$RA = (ID_v||ID_s||IP_v||Req_{LT}). \tag{6.1}$$

$$AT_{AStV} = K_{vs}[ID_{TGS}||TS_{AStV}|| \quad LT_{TGT}||K_{TGSse}]. \tag{6.2}$$

$$TGT = K_{TGSs}[(ID_v||ID_{TGS}||TS_{TGT}|| \quad IP_v||LT_{TGT}||K_{TGSse})]. \tag{6.3}$$

The Authentication Server (AS) possesses a registry of authorized users alongside their corresponding confidential keys. Verification involves confirming the presence of ID_v and the messages in the aforementioned registry. Upon successful validation, a duplicate of K_{Vs} is extracted. Subsequently, the AS initiates the creation of an AT_{AStV}, following which the Ticket Granting Ticket (TGT) is dispatched to the user. AT_{AStV} incorporates ID_{TGS}, timestamp TS_{AStV}, and a designated duration of validity. The TGT includes ID_v, ID_{TGS}, TS_{TGT}, IP_v, and LT_{TGT}. Encryption of both messages is carried out using K_{TGSse}, a symmetric key generated randomly, intended for the user's decryption of various communications from the TAS and RSU as a service server, limited to that specific instance. The Attribute message (AT_{VtTGS}) is encrypted with K_{Vs}, while the TGT undergoes encryption with

K_{TGSs}. Subsequently, these two messages are dispatched from the AS to the vehicle.

6.2.4.2 Vehicle and TGS Communication Stage

The decryption of AT_{AStV} by the key K_{Vs} is necessary for the vehicle, followed by the acquisition of K_{TGSse}. Subsequently, two messages will be generated by the vehicle, with the initial message comprising ID_S and LT_{TGT}. The second message, the AU_{VtTGS} which contains KS_{se} and ID_v that encrypted by K_{RSU} and the ID_v, TS and $Nonce$ that encrypted by the KS_{se}. Those key uses the AES encryption that utilizes the 128-bit key length. The 128-bit AES is lightweight and suitable to be implemented in the VANET environment rather than AES 192-bit and AES 256-bit. The VANET has several resource constraints, including low power consumption and memory efficiency. Vehicles in a VANET environment may rely on embedded systems with limited power. AES-128 strike a balance between security and power consumption. The AES-128 also requires less memory and processing than AES with longer key lengths, which is advantageous in devices with limited memory, such as those in vehicular communication systems. Moreover, our system also utilizes blockchain, which has a limited block size. This particular phase is illustrated in step (c) within Figure 6.5. After generating the AU_{VtTGS}, the TGS will sent it to the blockchain network to store it in step (d). Then, the AU_{VtTGS} is sent to the RSU in step (e). The details of AU_{VtTGS} storage in the blockchain is explained in Section 7.2.3.5.

In addition to the aforementioned generated messages, a vehicle is set to transmit the TGT acquired from the AS. Subsequently, the vehicle will dispatch the trio of messages illustrated in Equations (6.4)–(7.3) to the TGS. Upon receipt, the TGS will verify the presence of ID_S in plaintext within its own registry. The ID_S should be found in the TGS server's records; the TGS will duplicate the K_{Ss}. Encoded within the TGT lies a K_{TGSse}, which enables the TGS to decipher the AU_{VtTGS}. Those messages will be sent to the vehicles in step (f).

$$TGT = K_{TGSs}[(ID_v||ID_{TGS}||TS_{TGT}|| \quad IP_v||LT_{TGT}||K_{TGSse})]. \tag{6.4}$$

$$AT_{VtTGS} = (ID_s||Req_{LT}). \tag{6.5}$$

$$AU_{VtTGS} = K_{RSU}[KS_{se}||ID_v]||KS_{se}[ID_v||TS||Nonce] \tag{6.6}$$

6.2.4.3 Vehicle and RSU Communication Stage

Upon reception of attribute messages and the service ticket, the system will progress to steps (g) and (h) of Figure 6.5. The vehicles will utilize KT_{GSse} to decipher the AT_{TGStV}, subsequently obtaining K_{Sse}. Because every RSU have identic keytab for service, RSU2 can directly able to decrypt service ticket and take same K_{Sse}, so vehicle can do the authentication only with authentication messages, without involving TAS anymore. Subsequently, a fresh AU_{VtS} will be created by the vehicle, encompassing ID_v and TS_{VtS}, which will then be encrypted using K_{Sse}. The vehicle will then transmit both the ST and the authentication message to the server. In this scenario, the RSU represents the desired service for the vehicles. The messages sent from the vehicle to the RSU are detailed below:

$$ST = K_{Ss}[(ST||ID_v||ID_S||TS_{TGStV}|| \quad IP_v||LT_{ST}||K_{Sse})]. \tag{6.7}$$

$$AU_{VtS} = K_{Sse}[ID_v||TS_{VtS}]. \tag{6.8}$$

$$AU_{Sm} = K_{Sse}[ID_v||TS_{StV}]. \tag{6.9}$$

The RSU will perform a similar procedure involving the TGS. Subsequently, the RSU will decipher the ST utilizing its K_{Ss}, thereby acquiring authorization from the K_{Sse}. This authorization will be utilized to decrypt AU_{VtS}. Upon completion, the RSU will generate its own AU_{Sm} comprising ID_{RSU} and the TS_{StV}, as demonstrated by Equation (7.6).

Subsequently, the AU_{Sm} will be delivered to the user and subsequently decrypted by the vehicle utilizing K_{Sse}. The vehicle will then proceed to verify whether or not the service name included in the authentication corresponds to its designated service recipient. To mitigate potential replay assaults, the vehi-

cle will also validate the timeliness of the authentication by inspecting TS_{StV}. Furthermore, the vehicle will maintain its cache mechanism. After the mutual authentication procedure between the user and the service, the vehicle will retain a protected version of the service ticket in its cache for future reference.

6.2.4.4 TAS and RSU Interaction

TAS and RSU has a nondirect interaction in the initial authentication phase and the handover process. In the initial authentication phase, as shown in Figure 6.5, it indicates that TAS and RSU do not directly communicate; however, their functions are interrelated. The TAS comprises two components: the Authentication Server (AS) and the Ticket Granting Server (TGS). The AS is primarily responsible for managing the initial authentication of vehicles requesting RSU services within the secure network. Its functions include verifying vehicle identities, issuing Ticket Granting Tickets (TGTs), and encrypting user credentials for security. The TGS then issues service tickets, which vehicles use to access RSU services. The service server validates these tickets to authenticate the client and authorize service access.

In the handover process, the TGS generates the AU_{VtTGS}, serving as the credential required during the procedure. This credential is uploaded to the blockchain, which is managed via a smart contract to control the uploading and access of the AU_{VtTGS}. The smart contract's Solidity code includes parameters such as entity name, entity ID, and network name. The TAS is the designated entity authorized to upload the AU_{VtTGS}. This credential is accessed by the RSU when a vehicle submits a handover request. The RSU retrieves the AU_{VtTGS} by entering its entity name, which is verified through a consensus mechanism. Upon validation, the transaction is added as a new block, granting the RSU access to the AU_{VtTGS}.

6.2.5 Authentication Message Uploading in the Blockchain Phase

In the TGS communication stage, after obtaining the authentication message (AU_{VtTGS}), the TGS will send it to the blockchain. When authenticating

a vehicle, this AU_{VtTGS} proves the authenticity of the car. The solidity code is used to obtain this AU_{VtTGS}, which was produced previously by the off-chain environment (VANET). After that, to see it on the Ganache platform, it must pass via the smart contract. In this case, **AuthenticationMessage-Uploading()** is used to take the AU_{VtTGS} as input and then pass it as an argument of the function. After that, the entity "TAS" sends an **AuthenticationMessageUploading()** transaction request to store the AU_{VtTGS} in the blockchain. It will be saved in the blockchain block once the transaction is successful. After that, by establishing the events inside the function, I can see the information extracted from the blockchain on the Ganache platform.

Here, I consider the entity "TAS" as a sender and "RSU" as a receiver. I also use "EntityID" equal to "1" and "2" for the "TAS" and "RSU", respectively. Both entities exist in the same network as "TsushimaVanet". To obtain the AU_{VtTGS} from the blockchain, the entity "RSU" now delivers an **AuthenticationMessageAccessing()** transaction request. It will be able to acquire the AU_{VtTGS} from the blockchain after completing the transaction.

6.2.6 Handover Phase

The schematic representation illustrating the handover signaling process of the suggested approach can be observed in Figure 6.6. Upon recognition by a preceding Road Side Unit (RSU) that a vehicle or a cluster of vehicles has exited its designated coverage zone, an initiation of the handover procedure takes place.

- Step 1: The vehicle send the VID-targetRSU that verifies the target RSU has authenticity. The source RSU will determine the authenticity of the destination RSU by checking the neighbor table. In the initial authentication phase, the vehicle already received service ticket for RSU with the single session key K_{Sse} that valid for all RSU in the network. This service ticket and this K_{Sse} then will be uploaded in the network.

- Step 2: Before sending the request, that vehicle use same K_{Sse} that vehicle get from initial authentication phase for encrypt Authentication-MessageUploading() to blockchain. The vehicle advances by transmitting

a request message to the destination RSU. This RSU will send the request for the **AuthenticationMessageUploading()** transaction to the blockchain ledger.

- Step 3: The blockchain ledger will check the RSU with the smart contract agreements, especially for the entity name. After that, it will give the AU_sm to the destination RSU. Because all the RSU share identic K_{Sse}, so smart contract just need to check the authentication message. The destination RSU then can decrypt directly service ticket without connect to the TAS and get same K_{Sse}.
- Step 4: That RSU will equalize the authentication message sent by the vehicle and the authentication message in the blockchain ledger. Following successful validation, the destination RSU sends a message confirming the completion of the handover to the vehicle, thereby finalizing the transfer. Subsequently, the vehicle updates its ledger and disseminates the information. Session key K_{Sse} is released only one time for service and valid for all RSU.

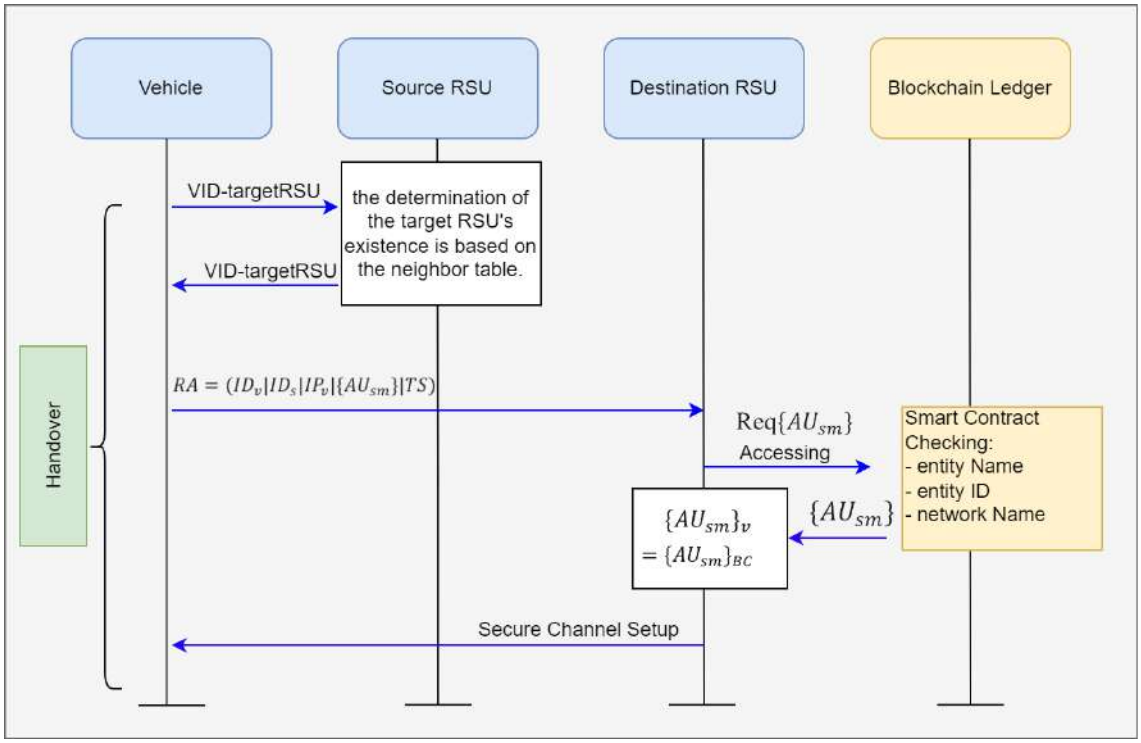


Figure 6.6: Handover signaling procedure.

6.3 Implementation and Discussion

The following subsections describe the simulation conditions and the outcomes of the suggested system, including network performance and blockchain performance results.

6.3.1 Implementation Environments

The evaluation of the proposed system's feasibility is detailed in this section. Table 6.1 presents the deployment context, framework, requisite technical resources, and software components.

Table 6.1: Implementation environment.

Software/Tool	**Configuration/Version**
Operating System	Windows 11 22H2 64-bit
Processor (CPU)	AMD Ryzen 7 5800U @ 1.90 GHz
Memory (RAM)	16 GB
Truffle Framework	Version 5.11.0
Ganache	Version 2.7.1
OMNeT++	Version 6.0.2
SUMO	Version 1.4.0

To develop the proposed system, various tools are employed to model the processes among the entities in the vehicular network. I utilize the off-chain and on-chain environment that is illustrated in Figure 6.7. In this case of an off-chain environment, I use OMNeT++ and SUMO to simulate the VANET authentication protocol, and to generate the authentication messages. OMNeT++ is utilized in our experiment as a tool for modeling and simulating discrete systems [41]. The design of the vehicular network uses NED language, combined with VEINS and INET frameworks. VEINS is an open-source framework designed for vehicular network simulations, utilizing OMNeT++ and SUMO [56]. The INET Framework serves as an open-source model library for OMNeT++, providing various protocols and models for communication networks [42].

The setup of vehicle nodes, as well as the creation of maps, junctions, and routes, are achieved through the use of SUMO [57]. After generating

the authentication messages are sent to the on-chain environment. In the on-chain environment, I run the blockchain and set the solidity code for the smart contract. In our experiment, I use Truffle and Ganache in the on-chain environment. Truffle constitutes a comprehensive development environment designed specifically for the Ethereum blockchain, offering an array of tools that facilitate the construction, evaluation, and deployment of smart contracts [45]. Ganache is the Ethereum development tool to simulate a blockchain environment locally and to conduct tests on the deployed smart contracts [46].

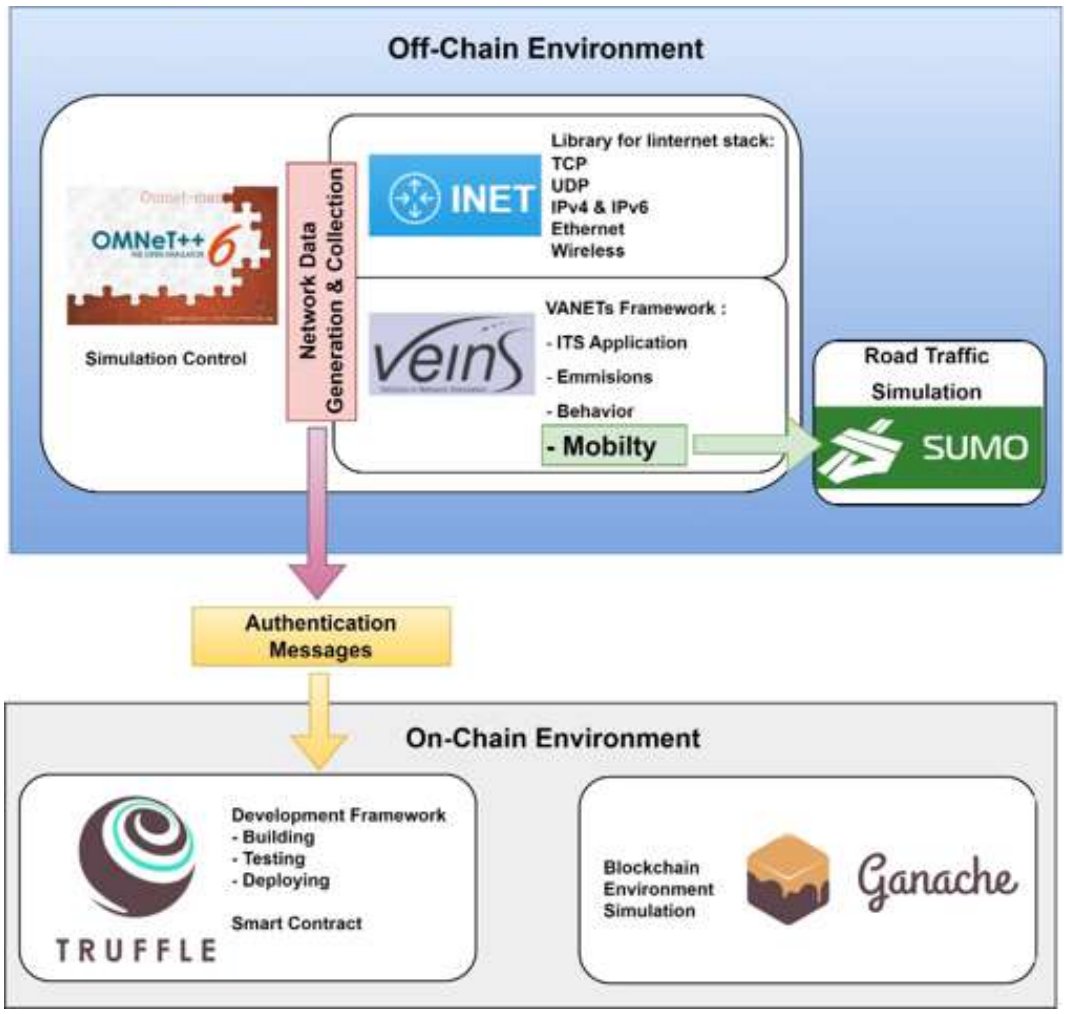


Figure 6.7: Off-chain and on-chain environment of the proposed system.

The relevant parameters that are considered for the experiment are detailed in Table 6.2. The communication model that I used is the IEEE 802.11p

standard [58] that is specifically designed for vehicular environments, supports vehicle-to-vehicle and vehicle-to-infrastructure communication, and is compatible with Dedicated Short-Range Communication (DSRC).The simulation area I used is 5000 m $\times$ 2500 m, which is located in two different areas with a scale of 1:20,000. In scenario 1, which represents a suburban area, I considered the geographical area near Okayama University, Tsushima, Japan. The initial creation of entities in the first stage comprises 100 vehicles, 4 RSUs, and 1 TAS, which functions like a Kerberos server. In scenarios 2 and 3, which represent an urban area, I used an area near Okayama Station that has higher vehicle density. Scenario 2 includes 1 TAS, 200 vehicles, and 8 RSUs. Scenario 3 contains 2 TASs, with the same number of vehicles and RSUs. I chose a communication range of 1000 m for the RSUs to allow broader coverage with fewer RSUs, and this range is often used as a standard in VANETs that aligns with typical ranges used in the IEEE 801.11p standard. The communication range of the vehicle that I used is 100 m, which provides reliable communication without excessive interference. The data rate set in the simulation is 27 Mbps, which aligns with the capabilities of the IEEE 802.11p standard. The safety messages are sent every 30 s to reach a sufficient update rate for traffic and hazard information. The mobility that I used in the simulation is the Duaroute Mobility as a standard mobility that is used in the IEEE 802.11p standard. Duaroute calculates routes for vehicles based on demand and network conditions. It generates routes by considering factors like road networks, traffic lights, and traffic density.

The blockchain is also designed to store the authentication messages. We conduct all experiments using Ganache CLI with the Truffle framework in fork mode, replicating Ethereum Mainnet state at a specified block height. It provides a single, local Ethereum node that inherits real contracts, balances, and storage from Mainnet, while offering full control over mining and block times. Although only one physical node is active, we leverage its multiple unlocked accounts that assigning each to a distinct role. In this case, the logical nodes are including 4 RSUs and 1 TAS in scenario 1, 8 RSUs and 1 TAS in scenario 2, and 8 RSUs and 2 TAS in scenario 3. It will simulate a multi-party network without additional servers. This configuration lets us test our smart-contract logic against authentic Mainnet state in a fully deterministic, permissioned

Table 6.2: Simulation parameters.

Parameter	**Value / Unit**
Communication model	IEEE 802.11p
Simulation area	5000 m × 2500 m
Communication range of RSU	1000 m
Communication range of vehicle	100 m
Data rate	27 Mbps
Safety messages	Every 30s
Mobility	Duaroute Mobility
Blockchain platform	Ganache CLI in fork mode (forked from Ethereum Mainnet)
Physical P2P nodes	1 (single Ganache process handling all consensus and mining)
Logical participants	Scenario 1: 4 RSUs and 1 TAS; Scenario 2: 8 RSUs and 2 TAS; Scenario 3: 8 RSUs and 1 TAS

environment, with logical separation of roles but without the overhead of a true multi-host P2P deployment.

6.3.2 The Network Performance Results

This subsection highlights the simulation results based on the network performances. The delays for the transmission and processing of messages and also the signaling overhead are described here in detail. The scenarios of suburban and urban are considered to evaluate the performances of the network.

6.3.2.1 Delay

To evaluate the behavior of the VANET protocol in terms of delay, Figure 6.8 has been plotted. AODV is used as the routing protocol with respect to suburban, urban with 1 TAS, and urban with 2 TASs scenarios. In VANETs, there are several parameters that are directly related to delays of the system. These includes simulation area, the vehicle's communication range, number of intermediary nodes (hops), number of vehicles and TASs, etc.

If the density of the vehicles is more in a simulation area, it leads to more

data and authentication requests within the network. As a result, the demand for authentication services rises, potentially increasing the overall network delay.

The communication range of a vehicle denotes the farthest distance for direct communication with another vehicle. In a larger communication range, vehicles can communicate over greater distances without requiring intermediate nodes. As a result, this can reduce the number of hops needed to transmit data across the network, which reduces the overall network delay, as more hops can increase the overall delay due to the additional processing and transmission times at each hop.

In a VANET, the relationship between the number of vehicles, TASs, and network delay are crucial. The TAS manages the authentication and security of the system. More TASs can help to distribute the load and handle requests more efficiently, potentially reducing network delays. In the same vehicle's communication range, if the number of TASs is fixed, then the delay of the system only depends on vehicle quantity. An increase in vehicle count correlates with a rise in system delay. I analyze three types of delay that happened in those three scenarios: authentication delay, handover delay, and end-to-end delay.

The authentication delay encompasses both the transmission and processing time of messages, starting from when a vehicle sends the Request Authentication (RA) message until it receives the AU_{Sm} as the credential to enter the network through the RSU [59]. The authentication delay must not exceed 100 ms to satisfy VANET criteria [17]. Equation (6.10) quantifies the delay, with T^i_{SRA} indicating the duration for vehicle authentication requests. Vehicle i is assigned time T^i_{RFA} for receiving the message of service authentication from the RSU. The variable N signifies the total vehicle count in the simulation context [60]. The following is a breakdown of how the authentication delay is calculated:

$$AuthDelay = \frac{1}{N}\sum_{i=1}^{N}(T^i_{SRA} \; - \; T^i_{RFA}). \tag{6.10}$$

Figure 6.8 shows the delays, including the authentication, handover, and end-to-end delay for overall scenarios. In the suburban scenario, it shows that in

a simulation involving 100 vehicles within a 5000 m × 2500 m communication area, the authentication delay becomes 85 ms (<100 ms), indicating that this delay performance still meets the requirement for VANETs [17]. However, when I tried to increase the number of vehicles from 100 to 200, considering the urban area, it effects the authentication delay of the network. In the urban area, I first considered 1 TAS. In this case, the authentication delay exceeds the VANET requirement and it reached 124 ms. This is because an increase in vehicles typically raises the demand of authentication services, potentially increasing the overall network delay, including authentication delay. Adding more TASs can mitigate this issue. For this, I considered 2 TASs in the same urban area without changing the other parameters and observed the effect of TASs in the overall network delay. In this case, the authentication delay reduces to 74 ms, which also meets the requirement for VANETs. This is because an additional TAS can help to distribute the load more effectively. Even when the authentication requests rise due to the increase of vehicles, this additional TAS can handle requests more efficiently throughout the network, reducing the delay associated with data transmission and processing.

Another delay observed in our study is the handover delay. Handover delay is quantified from the vehicle's exit from the previous RSU's coverage to its re-authentication by the new RSU [61]. The data indicate that, in every scenario, the handover delay is consistently lower than the initial authentication delay. However, the handover delay values exhibit a linear relationship with the changes in the authentication delay, increasing or decreasing correspondingly. In this proposed method, the handover delay indicates the re-authentication process associated with transitioning between RSU coverage areas; the vehicle needs re-authentication to make sure it has the correct credentials. From all of our experiments, the handover delay for scenarios 1, 2, and 3 are 55 ms, 69 ms, and 44 ms, respectively. Due to the authentication delay requirements, all the scenarios in our experiments still have values below the maximum limit (100 ms) of authentication delay. This indicates that our system functions well in the case of handover delay in all the cases, including the suburban and the urban areas.

The period required for a packet to traverse from its origin to its target location is termed end-to-end delay [62]. This is established by evaluating the

time the packet leaves the source vehicle and the time it reaches the destination vehicle. Equation (6.11) delineates the method for accurately calculating the end-to-end delay, denoted as EED, with TA representing arrival time and TS representing sending time [63]. Based on the ETSI TS 122 186 criteria regarding the service specifications for upgraded V2X scenarios [64], the maximum permissible end-to-end delay for information interchange between an On-Board Unit (OBU) and a Roadside Unit (RSU) while platooning must be less than 20 ms.

Figure 6.8 illustrates that scenarios 1 and 3 exhibit end-to-end delays of 15 ms and 9 ms, respectively, which meet this requirement. Conversely, scenario 2 records a 27 ms delay, surpassing the acceptable limit. This suggests that the urban scenario with 200 vehicles necessitates more than 1 TAS to meet the ETSI standard's ETSI TS 122 186 [64] end-to-end delay requirement.

$$EED = \sum T_A \; - \; T_S. \tag{6.11}$$

In this study, the vehicle serves as the source node while the TAS acts as the destination node. The delay includes propagation, transmission, queuing, and processing delays. Additionally, the end-to-end delay is consistently less than the authentication delay and handover delay. The variation patterns in the end-to-end delay observed in our proposed system closely follow the authentication delay values due to their intrinsic correlation. The authentication delay is derived from the Kerberos authentication process, which includes the processing time at each node and the time taken to exchange multiple messages among entities. So, the total delay in authentication within our system is notably affected by the end-to-end delay.

In an urban setting with 2 TASs scenario (scenario 3), it shows that all types of delay are reduced compared to the other two scenarios. This reduction is attributed to the increased number of TASs. However, the reduction of the delay is not exactly 50% due to the addition of 2 TASs compared to the scenario with 1 TAS. It is only 40.3%. This discrepancy is influenced by the Ad-Hoc On-Demand Distance Vector (AODV) routing protocol utilized in the VANET system. It is a popular routing protocol for VANETs that only establishes routes when needed [65]. This protocol has the ability to promptly adjust to dynamic alterations within the network, facilitated by its on-demand

route discovery mechanism, enabling swift responses to network modifications. The AODV protocol selects the minimal path based on node count between source and destination. In our authentication scheme, the sender is the vehicle and the receiver is the TAS. When the network has two TASs, one TAS may become more heavily loaded than the other, leading to load imbalance. The reason for this is that the vehicle's hop distance to the TAS depends on its mobility and dynamic network topology. Although AODV can dynamically adjust routes if the current route becomes invalid during its path maintenance stage, the newly adjusted route may still not account for equal load balancing. This load imbalance can result in higher authentication delays. Several factors contribute to the increased authentication delays, such as load imbalance, including server processing time and queuing delay. Each authentication request requires time for the server to process, which involves the encryption and decryption of Kerberos authentication messages, message generation, and other authentication steps. If the number of authentication requests exceeds the processing time capacity, the time needed to handle the request will increase. On the queuing delay side, if one server accepts many requests at the same time, then the request has to be in the queue until the server needs to process it. The larger the queue, the more time is required for each request, thereby increasing the overall authentication delay in the TAS with higher authentication requests.

A summary of the simulation network prerequisites is illustrated in Table 6.3. The "✓" symbol indicates that the delay has fulfilled the network performance requirement, and the "×" symbol indicates that the delay has not fulfilled the network performance requirement. The data presented in the table indicate that scenarios 1 and 3 fulfill all the network requirements for authentication delay and handover delay. This means that our proposal is still appropriate for implementation with 1 TAS for 100 vehicles and 2 TASs for 200 vehicles. However, scenario 2 did not fulfill the network requirements for authenticating when the number of vehicles is 200 and the number of TASs is only 1. Our proposal is appropriate for implementation in the suburban area with 1 TAS and the urban area, which has more density with 2 TASs.

Table 6.3: Network requirements fulfillment summary.

Scenario	Authentication Delay	Handover Delay
1	✓	✓
2	×	✓
3	✓	✓

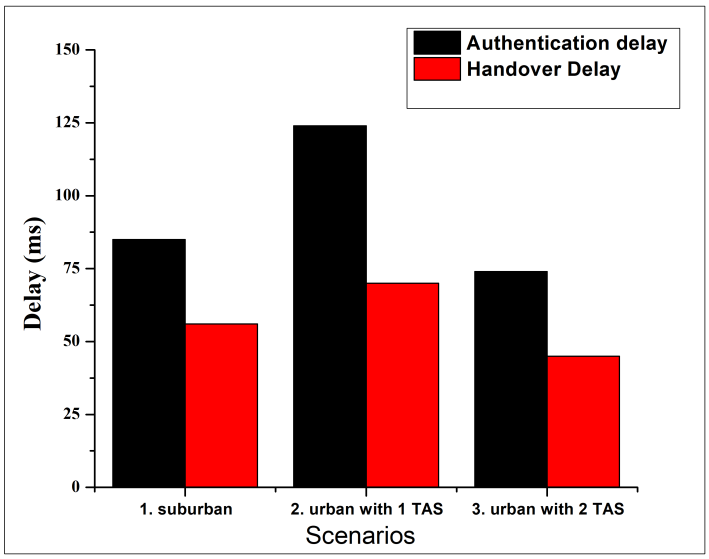


Figure 6.8: Comparison of several delays of different scenarios.

6.3.2.2 Signaling Overhead

The signaling overhead parameter quantifies the signaling messages as the expense per temporal unit during which the vehicle engages in the handover process. This parameter is computed by the product of the distance to the Road Side Units (RSUs), the unit transmission overhead, and the size of the messages utilized in vehicular communication. I evaluate the signaling overhead by counting our system's signaling overhead and comparing it to other proposed schemes. It is compared with the group-based handover control scheme for mobile internet using partially distributed mobility management (GP-DMM) [47] and a secure blockchain-based group mobility management scheme in the VANET (SEBGMM) [48]. The methodologies employed for the calculation of signaling overhead within the framework of GP-DMM are

delineated in the subsequent equation [47].

$$C_{DMM} = K[a(L_{Rs} + L_{RA})hop_{(v-RSU)} + 2b(L_{PBA} + L_{PBU})hop_{(CMD-RSU)} \tag{6.12}$$

The signaling overhead associated with SEBGMM for K Vs' handover is demonstrated as follows [48]:

$$C_{SEGBMM} = K[a(L_{Rs} + L_{RA})hop_{(v-RSU)}] \tag{6.13}$$

The signaling overhead for this proposed system (KBC) is shown as follows:

$$C_{KBC} = hop_{RSU-RSU}[a * Trans_u * L_{msg}] \tag{6.14}$$

where $hop_{RSU-RSU}$ represents the mean spatial separation between two Roadside Units (RSUs), a symbolizes the coefficient of weighting assigned to a particular link, $Trans_u$ signifies the unit of transmission, and L_{msg} indicates the aggregate size of the message that is exchanged during the signaling procedure. The size of the selected messages in the signaling process in this study is delineated in Table 6.4, while the comparative outcomes with prior research are illustrated in Figure 6.9. All the schemes used blockchain-based security in VANETs. The initial methodology (GP-DMM) [47] exhibits the most substantial signaling overhead, attributable to its framework that inadequately addresses session key negotiation and the preliminary authentication phase of nodes. Our proposed framework and SEBGMM [48] exhibit parallels in differentiating between the initial authentication and handover phases. Nonetheless, the proposed framework demonstrates the minimal signaling overhead attributable to its more streamlined architecture, which does not employ a Control Mobility Database (CMD) and solely relies on the TAS and RSU.

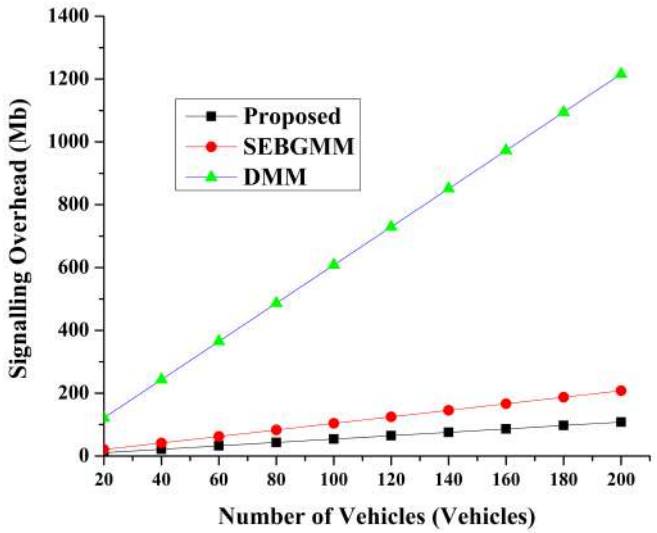


Figure 6.9: Signaling overhead.

Table 6.4: The size of messages in the signaling process.

Parameter	**Value**
Session Key	16 byte
Vehicle ID	8 byte
Timestamp (TS)	4 byte
Ticket for Initial Authentication	8 byte
HMAC	8 byte
AES Input Bit Length	16 byte
Lifetime	3 byte
IP Address	16 byte
Service Name	3 byte

6.3.3 Blockchain Performance Results

The four main functions of the smart contracts that make up the proposed system are as follows: **Deployment**, **EntityRegistration()**, **AuthenticationMessageUploading()**, and **AuthenticationMessageAccessing()**. The Ethereum blockchain platform compiles and deploys these contracts, determining how much gas is needed for each operation. Table 6.5 presents the basic gas consumption for each smart contract operation in our proposed system.

To calculate the basic gas consumption, first, one user registered and then uploaded an authentication message to the block of the blockchain. After that, another user accessed this authentication message from the block. To do so, the gas values for every operation were calculated. The experimental results indicate that, out of these four basic operations, the maximum gas usage for the **Deployment** operation is GWEI 1,145,042, significantly lower than the block gas limit of GWEI 6,721,975, demonstrating the practicality of our system.

The transaction cost in Ether is obtained by converting the gas value using Equation 6.15, and subsequently, the cost in USD is calculated by multiplying the Ether value by the exchange rate at the time of data collection, as expressed in Equation 6.16.

$$\text{Tx Cost (Ether)} = \frac{\text{Gas Used (GWEI)}}{10^9} \tag{6.15}$$

$$\text{Tx Cost (USD)} = \text{Tx Cost (Ether)} \times P_{Ether} \tag{6.16}$$

Table 6.5: The gas used for several operations of smart contracts.

Operation	**Gas Consumption (GWEI)**	**Tx Cost (Ether)**	**Tx Cost (USD)**
Deployment	1,145,042	0.001145	2.8478
EntityRegistration()	112,223	0.000112	0.2785
AuthenticationMessage Uploading()	29,733	0.000029	0.0721
AuthenticationMessage Accessing()	27,208	0.000027	0.0671

* *1 GWEI* $= 10^{-9}$ *Ether*, *1 Ether* = 2487.23 USD on 10 February 2024.

The transaction costs in Ethereum correlate with gas prices, with a baseline of GWEI 1 equating to 10^{-9} Ether. Throughout the analysis period, on 10 February 2024, the Ethereum to US dollar exchange rate stood at 1 Ether = USD 2487.23. Initially, the smart contract's deployment cost is approximated. This preliminary cost is essential for system initialization. Following this, the costs of execution for various smart contract operations are computed. The corresponding transaction cost of basic gas consumption by the system is

also listed in Table 6.5. This analysis demonstrates that the proposed strategy is practical for real-world implementation, considering the system's transaction costs [15].

I also analyzed the impact of the quantity of vehicles on the system's gas values. In this study, I varied the number of vehicles from 1 to 200 and calculated the corresponding gas values for several operations. Increasing the number of vehicles means more authentication messages must be stored in the blockchain block, affecting the gas values for the **Deployment** and **AuthenticationMessageUploading()** operations, as these two operations depend on the number of authentication messages. Figure 6.10 depicts this variance by showing that when the number of vehicles grows from 1 to 200, the gas value increases linearly. The study also highlighted the system's practicality, with the highest required gas value being GWEI 3,270,774 to execute the **Deployment** operation for 200 vehicles, which is well below the block's maximum gas limit of GWEI 6,721,975.

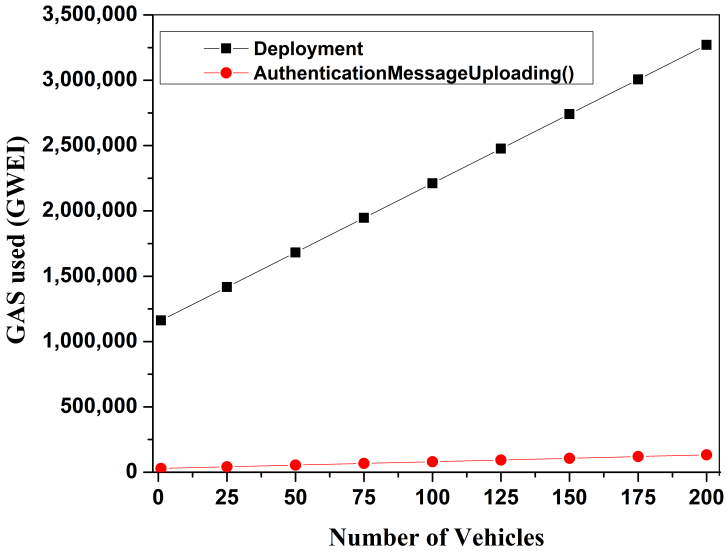


Figure 6.10: Number of vehicles vs. gas values.

The overall message size is crucial for effective message transfer communication in the VANET system. However, the size of an authentication message in a VANET system varies depending on the authentication proto-

col used. In general, the message size can range from hundreds of bytes to a few kilobytes, depending on the cryptographic algorithm, security features, and the specific VANET application. In [66], the authors used the message size ranges from 50 to 1200 bytes for secure messaging in the VANET system. In our study, I used the authentication message size of 32 bytes (without encryption), and I achieved an authentication message size of 48 bytes (after executing AES-128 encryption). The Ethereum blockchain regulates the size of blocks using the concept of gas, which calculates the amount of memory and processing time required for a transaction. The block gas limit sets a cap on the total gas within a block, which indirectly determines data storage capacity. Increasing the gas limit allows for more data to be included in a block [49]. Our investigation focuses on the storage of up to 200 vehicles, each identified by one authentication message. The memory needed to execute the transactions and to store in a block with the varying number of authentication messages is illustrated in Figure 6.11. This illustration demonstrates the relationship between authentication messages and memory storage in a block. Our experiment used the typical Ethereum block size of 1–2 MB [6]. When the transaction in blockchain is successfully completed, storing an authentication message in the blockchain network needed approximately 10 kilobytes of memory. The maximum memory required to store 200 authentication messages (for 200 vehicles) was found to be approximately 30 kilobytes, which is significantly lower than the maximum block size (1–2 MB) of the Ethereum network. This evaluation underscores the practicality of the system in terms of memory demands by the system.

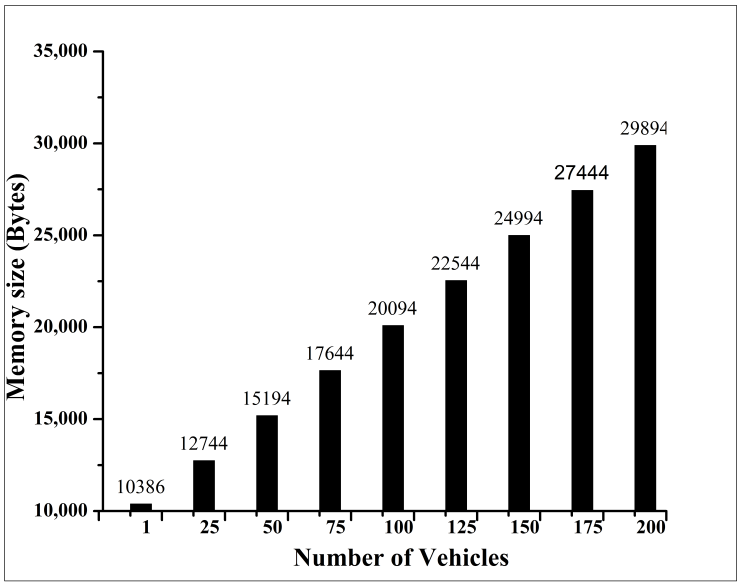


Figure 6.11: Memory size required for the block to store various authentication message.

6.3.4 The Security Analysis

The authentication messages are rendered immutable in the blockchain due to cryptographic hashing, whereby each block contains a unique hash derived from the previous block's data, creating an interlinked chain. This proposed system used the Ethereum blockchain that utilizes SHA-3 for its hashing operations. The inherent properties of cryptographic hash functions, such as collision resistance and irreversibility, contribute to the prevention of authentication message tampering or deletion. Furthermore, the Proof-of-Work (PoW) agreements method ensures agreement among the RSUs and the TAS on the validity of transactions before adding the authentication message to the blockchain. This decentralized agreement mechanism fortifies resistance against malicious alterations by requiring a majority consensus, making the authentication system robust and resilient against unauthorized modifications, thus ensuring the integrity and security of authentication for our proposed system.

The integration of blockchain with Kerberos authentication in this pro-

posed system introduces a noteworthy enhancement in privacy preservation by the implementation of smart contracts. It creates agreements or conditions between the RSUs and TASs. This condition must be fulfilled by the entities in order to perform the transaction. In our proposed system, the smart contract can automate the entities' registration, uploading the authentication message and requesting it. With the smart contract, I set that only the TAS entity can make an **AuthenticationMessageUploading()** request. On the other hand, RSU entities are able to make **AuthenticationMessageAccessing()** requests. This selective access ensures that only authenticated and authorized entities can access specific information, mitigating the risk of data exposure to unauthorized parties. Consequently, the use of smart contracts reinforces the privacy preservation entity by restricting access to sensitive information to only those entities that meet the predefined criteria, enhancing the security and reliability of the authentication framework within the context of vehicular communication.

The verification process to add a new block involves all of the network users, including RSUs and TASs, to ensure a collective responsibility for transaction execution, engendering a heightened level of trust among participants. The decentralized nature of blockchain mandates a PoW consensus mechanism, requiring agreement from the majority before transactions are added to the distributed ledger. This distributed validation mechanism contributes to the transparency of the network, as all transaction records are permanently inscribed in the blockchain. This transparency fosters accountability and fortifies the integrity of the authentication system by enabling all users to verify transaction history. Consequently, the usage of blockchain in this proposed system augments trust through shared responsibility and fosters transparency by recording transactions on an immutable and accessible ledger, thereby bolstering the reliability of the authentication processes within vehicular communication.

In the VANET system, malicious vehicles enter themselves into communication between two vehicles, impersonating them to gain access to information and inject false data. This type of attack is known as a man-in-the-middle attack, where the attacker eavesdrops on communication, alters messages, and breaches data integrity and privacy goals. The attacker may success-

fully pass through user authentication but will be blocked at the possession approval step. By inserting false information between genuine nodes or vehicles, the attacker undermines the trustworthiness of the data being exchanged, compromising network security. This attack poses a significant threat to the security standards of vehicle communication systems [67]. Due to the integration of blockchain with VANETs within the suggested framework, both the sender and the recipient must follow several procedures to complete the transaction successful. It does not need the assistance of man-in-the-middle to carry out the steps, even though passing them is necessary. A smart contract contains the predetermined terms of the transaction. These terms are automatically evaluated and validated throughout the transaction. This mitigates the man-in-the-middle attack. The integration of Kerberos authentication in VANETs enables mutual authentication between vehicles and RSUs, meaning both parties authenticate each other through the Key Distribution Center (KDC). By ensuring that both sender and receiver are legitimate, Kerberos makes it more challenging for an attacker to impersonate either party. Moreover, I use AES-128 symmetric key encryption to generate a session key for each communication session. This session key is known only to the authenticated parties, reducing the risk of an attacker intercepting and decrypting messages.

A consensus mechanism in a blockchain technology is a fault-tolerant mechanism that is advantageous to a single state of the network among the distributed multi-node systems in achieving the required agreement. The agreement is a list of rules and regulations for all the different participating nodes, which will eventually be helpful in deciding their contributions. Moreover, any transactions or events that take place in the system will be updated from time to time in the blockchain, and all the nodes will notify it. So, it is hardly possible to doubt the transparency of the transactions in a network that ultimately creates the trust among the nodes [63]. The Denial of Service (DoS) attack is an attack on the availability of the network. The main purpose of this attack is making the network unavailable to legitimate nodes. The attacker node generates a high volume of network traffic and consumes all the bandwidth of the network and makes it impossible for the RSU to manage such high-volume traffic, due to which the network become unavailable to the nodes [68]. Kerberos relies on a Ticket Granting Ticket (TGT) to streamline

session authentication. Once the initial authentication is complete, the TGT allows vehicles to access network services without contacting the AS each time. This minimizes bandwidth usage and decreases the number of authentication requests, therefore reducing the load on the network and limiting opportunities for DoS attacks.

6.3.5 The Scalability Challenges

In the design and implementation of a Kerberos-blockchain VANET authentication system, scalability is a critical consideration, especially when applied across diverse network scenarios such as suburban and urban environments. As the number of vehicles and number of TASs increases, the system must efficiently manage the load distribution of the TASs. Furthermore, while the blockchain component of the system is designed to store authentication messages, the block size limitation becomes a significant factor.

As the number of vehicles and TAS increases within the network, ensuring an even and efficient load distribution among TASs becomes a crucial challenge. In the suburban scenario, the number of vehicles is set to 100, whereas it increases to 200 in the urban environment for scenario 2, with the same number of TASs. Since scenario 2 did not meet the VANET requirements concerning the maximum authentication delay, I increased the number of TASs to two in scenario 3. This increase resulted in a reduction in the authentication delay, bringing the delay below the 100 ms threshold, thereby fulfilling the VANET criteria. However, the reduction in delay from scenario 3 is not exactly 50%, despite the addition of a second TAS, but is instead 40.3%. This discrepancy is influenced by the Ad-Hoc On-Demand Distance Vector (AODV) routing protocol, which selects the shortest path based on the node count between the source and the destination. When the network includes two TAS units, the load may become imbalanced, as each TAS may have different node counts between itself and the vehicles. This imbalance can lead to one TAS becoming more heavily loaded than the other, resulting in higher authentication delays for the vehicles managed by that TAS. As the vehicle density increases and one TAS handles more load than the other, there is a potential risk of bottlenecks and performance degradation if the load is

not managed properly. To address the load imbalance issue, several protocols could be considered for future work, such as the Dynamic Load Balancing Protocol (DLBP), Least Loaded Server Protocol, Round-Robin Load Balancing, or other load balancing methods. However, the results of this study indicate that the proposed method did not cause any bottlenecks, as the performance of scenario 3 satisfied the VANET delay requirements.

Another scalability challenge in the proposed method is the block size limitation for storing authentication messages. The maximum block size in Ethereum ranges from 1 to 2 MB. It is critical that the messages stored within each block do not exceed this limit, as doing so may increase the risk of blockchain forking and result in inconsistencies across the network. To assess the impact of vehicle density, I evaluated the effects of increasing the number of vehicles from 100 to 200 in our experiments. The maximum storage required for 200 authentication messages was calculated to be only 29.894 KB, which is significantly below the maximum block size threshold. Therefore, even as scalability expands, the three scenarios presented in this study remain well within the acceptable block size limits.

6.3.6 Comparative Analysis

The proposed system is also compared with some of the existing literature. Table 6.6 summarizes such comparison in terms of vehicle capacity, encryption techniques, Kerberos integration, gas, cost, delay parameters, and security attacks analysis.

The variation of the number of vehicles indicates the scalability of the system, which can influence the network topology and the effects of network performances in denser topology. In the literature, [23, 25–27, 69, 70] evaluated the parameters of the system considering the maximum number of vehicles to be 200 for the city area. Based on this, to evaluate the performance of our system I considered the number of vehicles from 1 to 200.

The exact number of vehicles in the VANET system can vary based on factors such as the size of the area, specific use case (e.g., safety, traffic management), and the simulation's goals. Denser urban areas often have higher vehicle counts, while suburban or less populated areas use lower values. In practice,

Table 6.6: Comparison of proposed method with existing literature.

Reference/ Year	**Vehicles for Simulation**	**Kerberos Integration**	**Encryption Technique**	**Gas and Cost Calculation**	**Delay Evaluation**	**Security Attack Analysis**
[23]/ 2020	25–50	×	not defined	×	Yes, end-to-end delay	×
[24]/ 2020	not defined	×	AES and CCMP	✓	Yes, authentication delay	✓
[25]/ 2020	1–50	×	ECC	×	Not evaluated	✓
[26]/ 2022	10–50	×	ECC	×	Yes, end-to-end delay	✓
[27]/ 2023	25–200	×	ECC	✓	Yes, authentication delay	✓
[69]/ 2018	1–50	×	ECC and IBE	×	Yes, end-to-end delay	✓
[70]/ 2024	20–100	×	ECC	×	Not evaluated	✓
Proposed/ 2024	1–200	✓	AES-128	✓	Yes, authentication, handover, and end-to-end delay	✓

the number of vehicles is chosen based on a balance between real-world traffic patterns and computational resources for the simulation.

By incorporating Kerberos, VANET systems benefit from improved security through efficient, scalable, and tamper-resistant authentication protocols. Kerberos is a secure protocol that enables mutual authentication between entities, reducing the risk of impersonation attacks. It uses a ticket-granting mechanism, requiring only one communication with the Key Distribution Center (KDC) for a ticket. Kerberos timestamps each ticket, preventing reuse by malicious actors. It supports large networks with multiple entities, making it ideal for VANET environments with numerous vehicles and infrastructure nodes.

In our proposed system, I used the AES-128 encryption technique to encrypt the authentication message. This brings several distinct benefits instead of using ECC encryption, which is used by most of the literature [25–27,69,70], except for [24], which also considers AES encryption. AES is a faster, simpler, and resource-efficient symmetric encryption method compared to ECC. It uses the same key for both encryption and decryption, making it ideal for real-time applications in VANETs. Additionally, AES requires less computational power, making it suitable for low-power devices and real-time traffic updates.

The integration of blockchain with VANET requires careful analysis of gas and cost implications. The gas values also indicate the real cost that will be

spent to execute the system. Gas value analysis can be beneficial for companies considering using a blockchain-based system. Optimizing smart contract code is crucial for real-world applications. A lightweight design reduces computational burden, making it viable for real-time operations in VANETs. Unlike most of the literature [23, 25, 26, 69, 70], our system analysis of gas and cost implications demonstrates its practicality.

Fast authentication is crucial in VANETs due to the high mobility of vehicles. Minimizing authentication delay ensures secure access and rapid information exchange. High handover delay can cause communication interruptions or network disconnections. Reducing end-to-end delay is essential for real-time communication. Our literature has analyzed all three possible delays (authentication delay, handover delay, and end-to-end delay) to ensure practicality, while some studies [23, 26, 69] focus only on end-to-end delay, and some others [24, 27] investigate only authentication delay.

6.4 Summary

This study introduced a blockchain and Kerberos-based authentication framework for VANETs, which encapsulated authentication messages within blockchain blocks. The practicability of implementing blockchain technology and the network's performance were evaluated. I executed three experiments to test the applicability of the system in diverse network scenarios to fulfill network and blockchain requirements. In the first scenario, I tested the proposed system in a suburban area with 100 vehicles and 1 TAS. In the second scenario, I changed the environment to an urban area that had an increased number of vehicles of 200, without any TAS addition. In the third scenario, I used the second scenario environment with one TAS addition. The performance of the system was assessed through network performance and blockchain performances impacted by TAS and the number of vehicles. Network performance included authentication, handover, and end-to-end delays. The blockchain performances included the gas value and the block size. The study revealed that, in the first scenario, the system fulfilled all of the network requirements. However, in the second scenario it did not fulfill the authentication delay requirement of VANETs because it exceeded the maximum limit

of 100 ms. The third scenario overcame that authentication delay problem by increasing the number of TASs to two. Concerning blockchain practicability, although the gas value and memory size of the system rose linearly with vehicle quantity, the system remained practicable as it did not exceed the maximum allowable gas value and memory size of the block. All of the findings showed that our proposal is applicable to be implemented in diverse network scenarios with the addition of a TAS. In future work, I would like to modify the authentication protocol in order to satisfy authentication delay requirements in VANET without adding the number of infrastructure, in this case is the TAS.

Chapter 7

The Design of an Integrated Authentication Server Architecture

In this chapter, I propose an integrated authentication server architecture to address the challenges of server scalability and infrastructure overhead in vehicular ad-hoc networks (VANETs). This chapter is based on the findings presented in Publication 4. I redesign the traditional two-server model (Authentication Server and Ticket Granting Server) into a unified Combined Blockchain Server (CBS). This consolidation aims to minimize authentication delay while reducing server redundancy and cost.

7.1 Introduction

In this study, I propose an optimization of the Kerberos-Blockchain-based authentication scheme by combining the Authentication Server (AS) and Ticket Granting Server (TGS) into a single Combined Server (CBS). It streamlines the authentication process to reduce overall authentication delay by reducing interactions count required during the authentication phase.

I utilized the blockchain to record the authentication messages generated by Kerberos authentication to reduce the handover delay while also maintaining security.

The system utilized the Advanced Encryption Standard (AES) with a 128-bit key and employs the Ethereum blockchain. To prove our proposed system's effectiveness, I assessed parameters including authentication delay, throughput, and signaling overhead. It evaluated in the simulation environments that is divided to on-chain and off-chain. In the on-chain environment, Ganache is used for blockchain simulation, alongside Truffle for smart contract management. Meanwhile, the off-chain environment utilizes OMNeT++ for network simulation and SUMO for modeling vehicular mobility in VANETs scenarios.

The contributions of this paper are delineated as follows:

1. A unified Kerberos Server architecture that eliminates the need for separate AS and TGS components while maintaining authentication delays below the critical 100ms threshold for Efficient Kerberos-Blockchain VANETs Authentication

2. Comprehensive evaluation using OMNeT++ network simulator integrated with SUMO traffic simulator to model both suburban and urban scenarios, with blockchain components implemented on the Ganache platform.

3. Quantitative performance analysis shows a 104% increase in system throughput and a 45% reduction in signaling overhead compared to the conventional separated-server architecture [8].

7.2 System Overview

This section discuss about the authentication method and authentication stages proposed in this research and entities that influenced in those stages. Those stages is includes initial authentication stage, authentication messages storing in blockchain stage, and handover stage.

7.2.1 Overview of the System

I utilized the blockchain to record the authentication messages generated by kerberos authentication to reduce the handover delay while also maintaining the security. The proposed system integrates Kerberos authentication within a blockchain framework by recording authentication messages generated by the Kerberos server onto the blockchain. By leveraging the decentralized nature of blockchain technology, the system aims to reduce authentication delays, particularly during the handover process in vehicular networks.

The system is structured into three main stages: the initial authentication stage, where authentication messages are generated; the blockchain uploading stage, where these messages are recorded onto the blockchain; and the re-authentication stage, conducted during handovers. After the initial authentication is completed, the vehicle no longer requires a direct connection to the

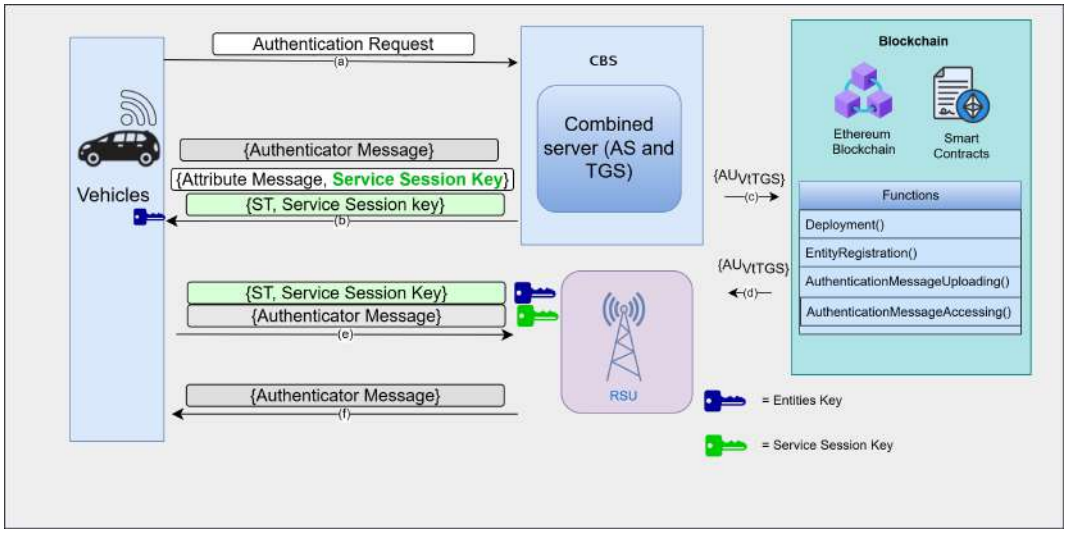


Figure 7.1: Proposed system architecture integrating AS and TGS into the combined blockchain server (CBS).

Kerberos server, as subsequent re-authentication processes only involve verifying authentication messages stored on the blockchain. This study focuses on optimizing the Kerberos authentication process by combining the Authentication Server (AS) and the Ticket Granting Server (TGS) to achieve lower authentication times compared to existing approaches.

The system utilized the Advanced Encryption Standard (AES) with a 128-bit key and is applied to secure the Kerberos authentication process. For the blockchain framework, the system employs the Ethereum blockchain. This proposed system is depicted in Figure 7.1.

7.2.1.1 Entities and Functions

Our system comprises three fundamental entities: the Combined Server (CBS) that has function as the Trusted Authentication Server, the Road Side Unit (RSU), and the vehicle. The CBS functions as a server responsible for registering all entities, authenticating vehicles, and disseminating authentication messages to the blockchain. RSUs serve as intermediaries between the vehicles and the CBS. They facilitate the integration of vehicles into the network, assist in the transfer of data among network entities, and disseminate information pertinent to traffic conditions. The vehicles assume the role of

network nodes and data sources. Each vehicle is equipped with an On-board unit (OBU), enabling it to communicate road-related data to nearby RSUs and neighboring vehicles within its communication range.

7.2.1.2 Overview of the Phases

The proposed system architecture is structured into three primary phases: system initialization and entity registration, initial authentication, and the handover procedure.

The registration stage serves the purpose of documenting all entities, which include the RSUs, CBS and the vehicles. The initial authentication process is initiated when a vehicle enters the network and submits an authentication request to the CBS. Upon successful authentication, the issues a service ticket to the vehicle, granting it access to connect to the network via RSU1. The CBS generates authentication messages, which are subsequently uploaded to the blockchain. These messages are utilized by vehicles during the handover process to facilitate seamless re-authentication.

When a vehicle moves beyond the coverage area of RSU1, it initiates a handover procedure by sending its previously issued authentication message to RSU2. RSU2 then interacts with the blockchain to retrieve and verify the corresponding credential stored during the initial authentication phase. If the provided credential matches the one recorded on the blockchain, the vehicle is successfully re-authenticated. This blockchain-assisted handover mechanism eliminates the need to repeat the initial authentication process, thereby significantly minimizing the overall authentication delay.

7.2.2 The Combination of the Server

In this research, I focuss on the combination of the Authentication Server (AS) and (TGS) from the conventional Kerberos-Blockchain system (KBS) to becomes the combined server (CBS) to reach the higher performance without any additional infrastructure.

In the original modular architecture, the AS was responsible for handling vehicle registration, secret key issuance, and generating Authentication

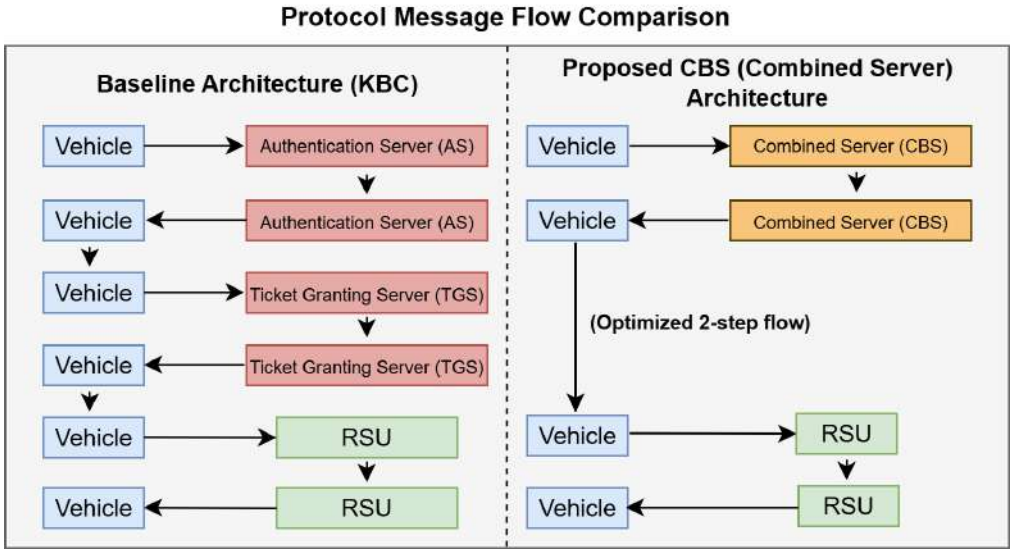
Messages, whereas the TGS handled ticket validation and Secure Ticket (ST) generation for service access. In contrast, the proposed CBS architecture consolidates these responsibilities into a single logical entity, allowing for more efficient and centralized handling of authentication processes.

A critical advantage of this unified design is the removal of inter-module communication. In the previous KBC model, packets such as Authentication messages and session keys needed to be transmitted between separate servers, introducing delays and processing overhead—especially under high vehicle density. By contrast, the CBS model eliminates these transmissions by directly processing all authentication steps within one server. For example, the `socketDataArrived` function in KBC now handles both ticket validation and secure ticket generation, which minimizes the need for inter-module signaling and improves overall efficiency.

Furthermore, the integration into a single server significantly improves end-to-end delay performance. In the KBC model, communication latency between the modules often led to processing bottlenecks, especially during handovers or in scenarios with rapid vehicle mobility. The CBS design, through centralized processing in functions such as `handleRegistrationOfVehicle Packet` and `handleVehicleTo-TGSPacket`, reduces these delays by enabling direct and immediate handling of all authentication-related packets.

The unified architecture also leads to a more efficient cryptographic key management scheme. In the previous model, AS and TGS each maintained their own sets of secret and session keys, increasing complexity and coordination requirements. In the CBS system, key generation and management—such as the `generateSecretKey` routine—are performed within a single scope. This centralization allows for simplified encryption workflows and secure ticket issuance, reducing the risk of synchronization issues between multiple servers.

Lastly, the CBS model offers considerable simplification of the overall codebase. The KBC architecture required distinct initialization, runtime operations, and teardown procedures, which increased software maintenance effort and redundancy. The CBS implementation reduces this complexity by utilizing unified methods such as `initialize` and a single event-handling function, `socketDataArrived`, to perform the complete authentication lifecycle. This makes the codebase more maintainable and extensible for future enhancements



** Repeated entities are visualized to indicate each protocol step, although they represent a single logical instance.*

Figure 7.2: Protocol message flow comparison between baseline (KBC) and proposed combined server (CBS) architecture.

or deployment scenarios.

In contrast to prior Kerberos–Blockchain frameworks where the Authentication Server (AS) and Ticket Granting Server (TGS) were implemented as separate modules, our proposed architecture integrates these roles into a single Combined Server (CBS). This integration is a purposeful optimization to streamline the authentication process.

This integration is shown in the Figure 7.2. In the baseline Kerberos architecture, authentication messages must pass through two servers (AS and TGS) before reaching the RSU, resulting in 4 protocol steps. In contrast, the CBS architecture eliminates one server layer, reducing the flow to 2 major hops: CBS and RSU, thus minimizing communication latency.

First, merging AS and TGS eliminates the inter-server communication steps inherent in traditional Kerberos. In the separated design, each vehicle's authentication requires two sequential exchanges (AS followed by TGS), incurring an extra network round-trip and processing delay. The unified CBS handles the entire ticket issuance in one place, reducing the number of message exchanges and thus lowering the overall authentication latency.

Second, the unified server architecture improves resource utilization and scalability through load consolidation. Instead of two servers that might individually run at partial load or become unevenly bottlenecked, the CBS serves a single aggregated request queue, ensuring full utilization of server capacity. This means the system can handle higher vehicle volumes without deploying additional servers – a limitation of the earlier two-server approach (KBC), which required extra infrastructure in high-density scenarios.

Third, our evaluation shows that this architectural redesign yields measurable performance gains without any new cryptographic algorithms. As reported in the revised manuscript, the integrated CBS consistently maintains authentication delays below the critical 100ms VANET requirement even in heavy traffic scenarios. By shortening the signal path and reducing duplicate work (e.g., cryptographic operations and database lookups performed separately by AS and TGS), the CBS can serve substantially more vehicle authentications per second.

Fourth, the CBS design continues to leverage blockchain for storing authenticators and does not alter the Kerberos trust model, thereby maintaining the same security guarantees as the baseline scheme (e.g., no secret keys exposed, tamper-proof audit trail).

In summary, the proposed CBS framework addresses key limitations in the traditional KBS architecture by offering reduced delay, centralized key management, elimination of inter-module communication, and improved code maintainability, thereby providing a more robust and scalable solution for secure VANETs authentication. The key difference in both scenario are shown in the Table 7.1.

Table 7.1: Comparison of key functional components between AS-TGS in KBC [8] and CBS method.

Functionality	**AS in KBC**	**TGS in KBC**	**CBS Method**
Vehicle Registration	handleVehicleRA	Not Applicable	handleRegistrationOfVehiclePacket
Authentication Ticket	aesEncrypt	Not Applicable	generateTicket, handleVehicleToTGSPacket
Ticket Validation	Not Applicable	handlePacketFromVehicle	handleVehicleToTGSPacket
Key Management	Separate for AS	Separate for TGS	Unified in generateSecretKey, handleRegistrationOfVehiclePacket
Communication Overhead	Requires AS–TGS communication	Requires AS–TGS communication	Eliminated due to integration

7.2.2.1 Comparison of Vehicle Registration: AS in KBC vs CBS

In the original Kerberos-based VANETs design, vehicle registration and initial authentication are handled by the Authentication Server (AS) via a dedicated function, commonly referred to as `handleVehicleRA`. This routine processes a vehicle's registration request by verifying the vehicle's identity (e.g., via a pre-shared secret or digital certificate) and then issuing the necessary Kerberos credentials, namely a Ticket-Granting Ticket (TGT) and a session key [8]. Internally, the AS generates a random session key for vehicle–TGS communication. It then constructs the TGT containing this session key and vehicle identity, encrypting it with the TGS's secret key. Simultaneously, the session key and accompanying metadata are encrypted with the vehicle's secret key. The response sent to the vehicle therefore includes two encrypted segments: one encrypted for the vehicle and the other (the TGT) encrypted for the TGS. This follows the standard Kerberos key exchange flow.

The encryption used in `handleVehicleRA` is based on AES-128, and both the TGT and the session key are handled through symmetric encryption. Each vehicle possesses a long-term secret key shared only with the AS, which is used to decrypt the session key and associated data. The TGT includes the session key and is readable only by the TGS, ensuring the integrity of the authentication process. This `handleVehicleRA` function is explained in the Algorithm 1.

Algorithm 1 handleVehicleRA – AS Registration Process (AES-128)

Input: Authentication request [ID_v, Nonce]
Output: Encrypted session key K_{TGSse} and Ticket Granting Ticket TGT

1: **if** not verifyID(ID_v) **then**
2: **return** error response
3: **end if**
4: $K_{Vs} \leftarrow$ getVehicleKey(ID_v)
5: $K_{TGSse} \leftarrow$ generateAES128Key()
6: $TGT \leftarrow$ AES_Encrypt([ID_v, K_{TGSse}, TS_{TGT}, LT_{TGT}], K_{TGSs})
7: $AT_{AStV} \leftarrow$ AES_Encrypt([K_{TGSse}, TS_{TGT}, LT_{TGT}], K_{Vs})
8: **return** [AT_{AStV}, TGT]

In the proposed unified architecture, the CBS consolidates both AS and

TGS functions into a single module. The core function responsible for this operation is `handleRegistrationOfVehiclePacket`. Upon receiving a registration packet, the CBS verifies the vehicle's identity and proceeds to generate a fresh session key. It then creates a ticket embedding the session key and relevant metadata (e.g., vehicle ID, timestamp, validity duration). This ticket is encrypted with the CBS's own master key. The session key and related metadata are simultaneously encrypted using the vehicle's long-term secret key. This `handleRegistrationOfVehiclePacket` function is explained in the Algorithm 2.

Due to the merged roles, CBS can issue a final service ticket or authenticator directly to the vehicle, reducing the number of steps required in the authentication exchange. Additionally, the CBS logs a record of the authentication result onto a blockchain ledger, enabling subsequent verification by RSUs or other entities.

Algorithm 2 handleRegistrationOfVehiclePacket – CBS Registration Process (AES-128)

Input: Registration packet [ID_v, credentials]
Output: Encrypted session key K_{TGSse} and Ticket Granting Ticket TGT

if not verifyCreds(ID_v, *credentials*) **then**
  return error response
end if
$K_{Vs} \leftarrow$ getVehicleKey(ID_v)
$K_{TGSse} \leftarrow$ generateAES128Key()
$TGT \leftarrow$ AES_Encrypt([ID_v, K_{TGSse}, TS_{TGT}, LT_{TGT}], K_{TGSs})
$AT_{AStV} \leftarrow$ AES_Encrypt([K_{TGSse}, TS_{TGT}, LT_{TGT}], K_{Vs})
return [AT_{AStV}, TGT]

For encryption and key handling, both functions apply AES-128 encryption to ensure secure session establishment, but they differ in mechanism and performance. The original AS employs AES-128 encryption, requiring two steps and two separate encrypted objects (one for the vehicle, one for the TGS). In contrast, the unified CBS uses AES-128 encryption and merges the AS and TGS steps into one function, thereby simplifying the ticketing process. In both implementations, a secure session key is generated by the server and distributed securely: encrypted for the vehicle using its own key, and

stored in a ticket encrypted for the relevant server (TGS or CBS). The CBS implementation also benefits from blockchain integration, enabling verifiable and tamper-proof authentication logging. Overall, the unified CBS function achieves reduced latency, simpler protocol flow, and stronger encryption with AES, while preserving the secure key handling principles of Kerberos.

7.2.2.2 Comparison of Authentication Ticket

In the Authentication Server (AS) implementation, the function `aes Encrypt` is utilized to secure the initial authentication ticket, referred to as the Ticket Granting Ticket (TGT), generated for the vehicle. After verifying the vehicle's identity, the AS generates a fresh 128-bit Ticket Granting Server session key (K_{TGSse}), which is shared between the vehicle and the Ticket Granting Server (TGS). This session key, along with the vehicle's identifier (ID_v), timestamp (TS_{TGT}), and requested ticket lifetime (Req_{TGT}), is encrypted using AES-128 with the TGS's long-term secret key (K_{TGSs}) to produce the TGT. Additionally, the AS encrypts the same session key and its associated metadata using the Vehicle's secret key (K_{Vs}) to securely deliver it to the On-Board Unit (OBU). This dual encryption ensures confidentiality and isolation of trust between the communicating entities. This `aesEncrypt` function is explained in Algorithm 3.

In contrast, the Combined Server (CBS) integrates the roles of both the AS and TGS. When a vehicle submits a request containing a TGT and the authenticator message (AU_{VtTGS}), the `handleVehicleToTGSPacket` function first decrypts the TGT using K_{TGSs} to retrieve the embedded session key K_{TGSse} and validate the ticket's timestamp. The authenticator AU_{VtTGS} is then decrypted using K_{TGSse} to verify the freshness and authenticity of the request. Upon successful validation, the CBS generates a new 128-bit service session key (K_{Sse}) to enable secure communication with the target service. The `generateTicket` function then encrypts this session key along with the required metadata using the service's secret key (K_{Ss}) to produce the Service Ticket (ST), which is returned to the vehicle along with the encrypted attribute ticket (AT_{TGStV}), both protected under AES-128 encryption with K_{TGSse}. This `handleVehicleToTGSPacket` function is explained in Algorithm 4.

Therefore, although both implementations rely on AES-128 encryption to protect tickets and maintain session key confidentiality, the CBS architecture enhances the authentication process by eliminating inter-server communication and combining validation and issuance operations within a single module. This design significantly reduces latency and signaling overhead, rendering it particularly well-suited for dynamic and resource-constrained VANETs environment.

Algorithm 3 Service Ticket Issuance – KBC (AES-128)

Input: Received TGT, AU_{VtTGS}, requested ID_S
Output: Encrypted session key K_{Sse} and Service Ticket ST

$TGT_{\text{dec}} \leftarrow \text{AES\_Decrypt}(TGT, K_{TGSs})$
$K_{TGSse} \leftarrow TGT_{\text{dec}}.K_{TGSse}$
$AU_{\text{dec}} \leftarrow \text{AES\_Decrypt}(AU_{VtTGS}, K_{TGSse})$
if AU is valid **and** timestamp is fresh **then**
    $K_{Sse} \leftarrow \text{generateAES128Key}()$
    $ST \leftarrow \text{AES\_Encrypt}([ID_v, ID_S, K_{Sse}, TS_{ST}, LT_{ST}], K_{Ss})$
    $AT_{TGStV} \leftarrow \text{AES\_Encrypt}([ID_S, TS_{ST}, LT_{ST}, K_{Sse}], K_{TGSse})$
    return $[ST, AT_{TGStV}]$
end if

Algorithm 4 Service Ticket Issuance – CBS (AES-128)

Input: Received TGT, AU_{VtTGS}, requested service ID_S
Output: Encrypted session key and Service Ticket (ST)

$TGT_{\text{dec}} \leftarrow \text{AES\_Decrypt}(TGT, K_{TGSs})$
$K_{TGSse} \leftarrow TGT_{\text{dec}}.K_{TGSse}$
$AU_{\text{dec}} \leftarrow \text{AES\_Decrypt}(AU_{VtTGS}, K_{TGSse})$
if AU_{dec} is valid **and** TS_{VA} is fresh **then**
    $K_{Sse} \leftarrow \text{generateAES128Key}()$
    $ST \leftarrow \text{AES\_Encrypt}([ID_v, ID_S, K_{Sse}, TS_{ST}, LT_{ST}], K_{Ss})$
    $AT_{TGStV} \leftarrow \text{AES\_Encrypt}([ID_S, K_{Sse}, TS_{ST}, LT_{ST}], K_{TGSse})$
    return $[ST, AT_{TGStV}]$
else
    return error response
end if

7.2.2.3 Authentication Logic and Encryption Handling in KBC and TGS Function

The `handleVehicleToTGSPacket` function in the CBS and the `handle PacketFromVehicle` function in the Ticket Granting Server (TGS) module perform distinct but complementary roles within the proposed authentication protocol. In the CBS stage, the `handleVehicleToTGSPacket` function is responsible for verifying the vehicle's identity using the Vehicle's secret key (K_{Vs}). Upon successful validation, the CBS generates a fresh Ticket Granting Server session key (K_{TGSse}) for secure communication between the vehicle and the TGS. This session key is encapsulated in two outputs: (i) the Ticket Granting Ticket (TGT), encrypted with the TGS's long-term secret key (K_{TGSs}), and (ii) the authentication ticket from KBC to vehicle (AT_{AStV}), encrypted with the vehicle's secret key (K_{Vs}). The TGT is designed to be opaque to the vehicle and validated solely by the TGS, while the AT_{AStV} enables the vehicle to retrieve the session key K_{TGSse}. The `handleVehicleToTGSPacket` function is explained in the Algorithm 6.

In contrast, the `handlePacketFromVehicle` function that explained in Algorithm 5 in the TGS processes authentication requests from vehicles by validating the previously issued TGT and the Authenticator from vehicle to TGS (AU_{VtTGS}). The TGT is decrypted using K_{TGSs} to extract K_{TGSse}, which is then used to decrypt and validate AU_{VtTGS}. This dual-stage verification ensures both the integrity of the ticket and the freshness of the authentication request, thereby mitigating replay attacks. Upon successful validation, the TGS generates a Service session key (K_{Sse}) to be shared between the vehicle and the target service. This key is encapsulated in two forms: the Service Ticket (ST), encrypted with the service's secret key (K_{Ss}), and the attribute ticket from TGS to vehicle (AT_{TGStV}), encrypted with K_{TGSse} for decryption by the vehicle. Both CBS and TGS module thereby implement a layered encryption strategy to safeguard key confidentiality and mutual authentication, using timestamped authenticators and rigorous key validation procedures tailored for secure operation in Vehicular Ad-Hoc Networks (VANETs).

Algorithm 5 `handlePacketFromVehicle` – Service Ticket Issuance and TGT Verification (TGS)

Input: Request $[TGT,\ AU_{VtTGS},\ AT_{VtTGS},\ ID_S]$
Output: Service Ticket (ST) and Access Ticket (AT_{TGStV})

$TGT_{dec} \leftarrow$ AES_Decrypt($TGT,\ K_{TGSs}$)
Extract $[ID_v,\ ID_{TGS},\ TS_{TGT},\ IP_v,\ LT_{TGT},\ K_{TGSse}]$ from TGT_{dec}
if expired($TS_{TGT},\ LT_{TGT}$) **or** malformed format **then**
    return error (invalid or expired TGT)
end if
$AU_{dec} \leftarrow$ AES_Decrypt($AU_{VtTGS},\ K_{TGSse}$)
Extract $[ID'_v,\ TS_{VA}]$ from AU_{dec}
if $ID_v \neq ID'_v$ **or** AU is replayed **or** IP mismatch **then**
    return error (invalid AU_{VtTGS})
end if
Mark AU_{VtTGS} as used
$K_{Sse} \leftarrow$ generateAES128Key()
$AT_{\text{plain}} \leftarrow [ID_S,\ TS_{TGStV},\ LT_{ST},\ K_{Sse}]$
$AT_{TGStV} \leftarrow$ AES_Encrypt($AT_{\text{plain}},\ K_{TGSse}$)
$ST_{\text{plain}} \leftarrow [ID_v,\ ID_S,\ TS_{ST},\ IP_v,\ LT_{ST},\ K_{Sse}]$
$ST \leftarrow$ AES_Encrypt($ST_{\text{plain}},\ K_{Ss}$)
return $[ST,\ AT_{TGStV}]$

Algorithm 6 handleVehicleToTGSPacket – Initial Authentication and TGT Issuance (CBS)

Input: Vehicle request $[ID_v,\ ID_S,\ Req_{TGT},\ AU]$
Output: Encrypted TGT and AT_{AStV}

if not isRegistered(ID_v) **or** malformed request **then**
    return error response
end if
$K_{Vs} \leftarrow$ getVehicleKey(ID_v)
$K_{TGSse} \leftarrow$ generateAES128Key()
$TGT_{\text{plain}} \leftarrow [ID_v,\ ID_{TGS},\ TS_{TGT},\ IP_v,\ Req_{TGT},\ K_{TGSse}]$
$TGT \leftarrow$ AES_Encrypt($TGT_{\text{plain}},\ K_{TGSs}$)
$AT_{\text{plain}} \leftarrow [ID_{TGS},\ TS_{AStV},\ Req_{TGT},\ K_{TGSse}]$
$AT_{AStV} \leftarrow$ AES_Encrypt($AT_{\text{plain}},\ K_{Vs}$)
return $[TGT,\ AT_{AStV}]$

7.2.3 The System Phases

The phase of the proposed system are divided into five stages, including system initialization and registration phase, vehicles and CBS communication stage, CBS and RSU interaction, authentication message uploading in the blockchain phase, and handover phase, are shown in the Figure 7.1.

7.2.3.1 System Initialization and Registration Phase

During offline registration phase, vehicles submit their credentials, including the vehicle ID, password, origin, destination, service type, and access rights (such as read, write, and modify permissions). Similarly, Roadside Units (RSUs) are registered with details like their physical location, MAC address, IP address, and RSU ID. Before the verification phase begins, the CBS generates and securely distributes secret keys to each entity involved.

7.2.3.2 Vehicle and CBS Communication Stage

This phase includes communications that are relayed from the vehicle to CBS, and conversely, reflecting the processes outlined in steps (a) and (b) illustrated in Figure 7.1. In the initial stage, the vehicle conveys a Request Authentication (RA) to the CBS, which comprises the ID_v, the particular service designation that the vehicle aims to access (in this scenario, the service designation is related to RSU service), IP_v, and the lifetime (LT). The LT functions to limit the temporal span, thereby bolstering system security through a bounded time constraint. This information will be transmitted to the CBS.

The messages dispatched by the vehicles to the CBS are delineated in Equation (7.1). The corresponding response messages are represented in Equations (7.2), (7.3), and (7.5). The parenthesis symbol signifies a collection of unencrypted messages, whereas the bracket symbol represents the encrypted messages.

$$RA = (ID_v||ID_s||IP_v||Req_{LT}). \tag{7.1}$$

$$AT_{VtTGS} = (ID_s||Req_{LT}). \tag{7.2}$$

$$AU_{VtTGS} = K_{RSU}[KS_{se}||ID_v]||KS_{se}[ID_v||TS||Nonce] \tag{7.3}$$

$$ST = K_{Ss}[(ST||ID_v||ID_S||TS_{TGStV}||IP_v||LT_{ST}||K_{Sse})] \tag{7.4}$$

The initial function of CBS involves maintaining a database of authenticated users along with their associated secret keys. When a verification request is received, the CBS checks whether the identifier ID_v and corresponding message exist within this registry. If validation is successful, the secret key K_{Vs} is retrieved. The CBS then generates an attribute token AT_{VtTGS}, which includes the service identifier ID_s, the requested lifetime (LT), and a specified validity period. This information is encrypted using a randomly generated symmetric session key K_{TGSse}, which is used by the user to decrypt subsequent communications from the CBS and RSU during the current session. Additionally, the attribute message AT_{VtTGS} is encrypted with the vehicle's secret key K_{Vs}. These encrypted messages are then transmitted from the CBS to the vehicle.

Following this, the CBS creates the authentication message AU_{VtTGS}, which contains KS_{se} and ID_v that encrypted by K_{RSU} and the ID_v, TS and $Nonce$ This message is encrypted by the KS_{se}, employing AES encryption with a 128-bit key. Once the AU_{VtTGS} is generated, it is uploaded to the blockchain network in step (c) for secure storage. Afterward, the same message is forwarded to the RSU in step (d). Further explanation of the blockchain storage mechanism for AU_{VtTGS} is provided in Subsection 7.2.3.5.

7.2.3.3 Vehicles and RSU Interaction

Once the vehicle receives the attribute message and the service ticket, the authentication process advances to steps (e) and (f) as illustrated in Figure 7.1. Using the session key K_{TGSse}, the vehicle decrypts the received AT_{TGStV} to retrieve the service session key K_{Sse}. Subsequently, the vehicle sends both

the service ticket and the encrypted authentication message to the intended service provider, which in this context is the RSU. The structure and content of the messages transmitted by the vehicle to the RSU are outlined below:

$$ST = K_{Ss}[(ST||ID_v||ID_S||TS_{TGStV}|| \quad IP_v||LT_{ST}||K_{Sse})]. \tag{7.5}$$

$$AU_{Sm} = K_{Sse}[ID_v||TS_{StV}]. \tag{7.6}$$

The RSU follows a process analogous to that previously executed by the TGS. It begins by decrypting the service ticket (ST) using its private key K_{Ss}, through which it obtains the session key K_{Sse} for further communication. This session key is then used to decrypt the vehicle's authentication message AU_{VtTGS}. After successful decryption and validation, the RSU generates a new authentication response message, denoted as AU_{Sm}, which contains its own identifier ID_{RSU} and a timestamp TS_{StV}, as formulated in Equation (7.6).

The generated AU_{Sm} is transmitted back to the vehicle and decrypted using the session key K_{Sse}. Upon receipt, the vehicle performs two critical checks: it verifies that the service name included in the response matches the expected service, and it validates the freshness of the message by examining the timestamp TS_{StV}, thereby preventing replay attacks. Additionally, the vehicle updates its local cache mechanism. Upon successful completion of mutual authentication between the vehicle and the RSU, the service ticket is securely stored in the vehicle's cache for subsequent use in future sessions.

7.2.3.4 CBS and RSU Interaction

The CBS and the RSU exhibit a non-direct engagement during the preliminary authentication stage as well as throughout the handover procedure. In the preliminary authentication stage, as depicted in Figure 7.1, it is evident that the CBS and the RSU do not engage in direct communication; nevertheless, their operational roles are interconnected. The CBS has first task to oversee the initial authentication of vehicles that seek services from the RSU

within the confines of a secure network. The second responsibility of CBS is to encompass the verification of vehicle identities, the issuance of Ticket Granting Tickets (TGTs), and the encryption of user credentials to ensure security. Subsequently, it is also responsible for issuing service tickets, which are utilized by vehicles to gain access to RSU services. The service server then validates these tickets to authenticate the client and facilitate authorized access to services.

During the handover phase, CBS generates the AU_{VtTGS}, which serves as the essential credential for enabling seamless authentication continuity. This credential is securely uploaded to the blockchain, where access control and upload operations are governed by a predefined smart contract. The underlying smart contract, implemented in Solidity, defines parameters such as the entity's name, unique identifier, and associated network designation. Only the CBS, designated as the trusted authority, is authorized to upload the AU_{VtTGS} credential to the blockchain.

When a vehicle initiates a handover to a new RSU, the RSU retrieves the corresponding AU_{VtTGS} by submitting its own entity name to the blockchain interface. Access is granted through a consensus-based validation mechanism that ensures the authenticity and integrity of the request. Upon successful verification, the transaction is committed to the blockchain as a new block, thereby allowing the RSU to access the stored AU_{VtTGS} and complete the handover process securely.

7.2.3.5 Authentication Message Uploading in Blockchain

Once the AU_{VtTGS} is generated by CBS, it is subsequently stored on the blockchain. This credential serves as proof of the vehicle's authenticity during future authentication attempts. The upload and retrieval of the AU_{VtTGS} are facilitated through smart contract functions written in Solidity. Specifically, the off-chain simulation environment (OMNeT++) produces the AU_{VtTGS}, which must be transmitted through the smart contract interface before it can be visualized or interacted with on the Ganache platform. To store the authentication credential, the smart contract function **AuthenticationMessageUploading()** is invoked. This function accepts the AU_{VtTGS} as an input argument, and a transaction request is initiated by the CBS to up-

load this data to the blockchain. Upon successful validation and mining of the transaction, the message is stored as part of a blockchain block. The function also emits events, which allow observers to extract and verify the stored data via the Ganache user interface.

In this scenario, the CBS is modeled as the sender, while the RSU acts as the intended recipient. For identification purposes, the entity ID for the CBS is assigned as "1," and the RSU is designated as "2." Both entities operate within a shared blockchain network labeled "TsushimaVanet." To retrieve the AU_{VtTGS}, the RSU sends a transaction request by calling the **AuthenticationMessageAccessing()** function within the smart contract. Once the transaction is confirmed, the RSU successfully obtains the AU_{VtTGS} from the blockchain, enabling it to complete the authentication or handover operation securely.

7.2.3.6 Handover Phase

When a vehicle exits an RSU's coverage, a handover process begins with the source RSU verifying the target RSU. The vehicle sends a request, prompting the target RSU to upload an authentication message to the blockchain. After validation via smart contracts, the blockchain returns an authentication unit. The RSU confirms the vehicle's data, completes the handover, and the vehicle updates and shares the new state.

7.3 Implementation and Discussion

This section presents the performance analysis of the proposed CBS integrated with blockchain for VANETs authentication. The evaluation is conducted using OMNeT++ and SUMO for network and mobility simulation, while blockchain functionality is implemented via Ganache and Truffle.

7.3.1 Evaluation

To assess the performance and practicality of the proposed authentication framework, a series of evaluations were designed across both on-chain and off-

chain environments.

Figure 7.3 illustrates the implementation flow of the proposed VANETs authentication system, which operates across two main environments: the Off-Chain Environment and the On-Chain Environment. The simulation begins with SUMO (Simulation of Urban Mobility), which generates 100 vehicles and simulates their realistic mobility patterns within an urban traffic map. These mobility traces are integrated into the INET framework within OMNeT++, which provides the standard network protocol stack (TCP, UDP, IPv4, and wireless communication) to enable data transmission between vehicles and roadside units (RSUs). VEINS acts as the bridge between SUMO and OMNeT++, supporting VANETs behavior, message emission control, and routing logic. It also triggers each vehicle to send authentication requests (RA) upon entering the network.

The authentication request is processed by CBS, a custom module that integrates Kerberos AS and TGS functionalities along with AES-128 encryption. CBS generates the Authentication Message and transmits it to the On-Chain Environment, where Truffle manages the deployment of Ethereum smart contracts and Ganache simulates the local blockchain. The detail specification of the system environment is in the Table 7.2.

Table 7.2: Implementation environment.

Software/Hardware	**Configuration/Version**
Operating System	Windows 11 22H2 64-bit
Processor	AMD Ryzen 7 5800U @ 1.90 GHz
RAM	16 GB
Truffle Framework	Version 5.11.0
Ganache (Blockchain Simulator)	Version 2.7.1
OMNeT++	Version 6.0.2
SUMO (Simulation of Urban Mobility)	Version 1.4.0

I conduct two scenarios to evaluate the system and compare the KBC approach with CBS approach. The initial scenario concentrates on a suburban area. In this context, I use 100 vehicles, complemented by four RSUs, alongside a server. The experimental procedures were carried out utilizing the map representations of the Tsushima Campus area in Okayama, Japan, as depicted in Figure 7.4a. The maps were generated by OpenStreetMap, which oper-

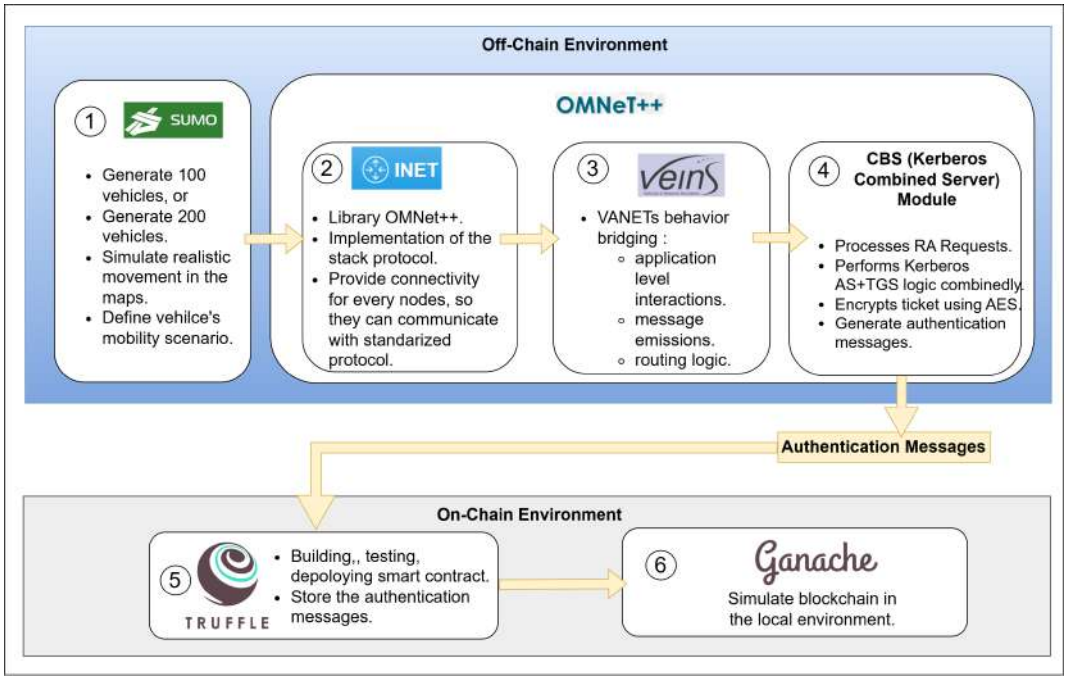


Figure 7.3: System environment diagram.

ates under the open database license (ODbL) integrated with SUMO which is characterized as a permissive open-source licensing framework.

In the second scenario, I increase the number of vehicles from 100 to 200 as a representation of infrastructure in an urban area. This particular scenario was executed in Okayama Station area in Okayama City, Japan, as depicted in Figure 7.4b. The aim is to ascertain whether the efficacy of the authentication system remains within acceptable parameters as the vehicle count escalates. I positioned the server at the central locus of the maps. The diminutive yellow dots on the maps signify the vehicles. In light of the enhanced infrastructural availability within the urban context, I also doubled the quantity of Roadside Units (RSUs) from 4 to 8.

To prove our proposed system's effectiveness, I assessed parameters that I take in testing scenarios, including authentication delay, throughput, and signaling overhead. Initially, the evaluation replicates the methodology of previous studies to enable a direct and accurate comparison of system performance. Subsequently, the number of vehicles is progressively increased, up to a maximum of 300 vehicles, to determine the system's capacity while maintaining authentication delays within the acceptable thresholds defined for VANETs

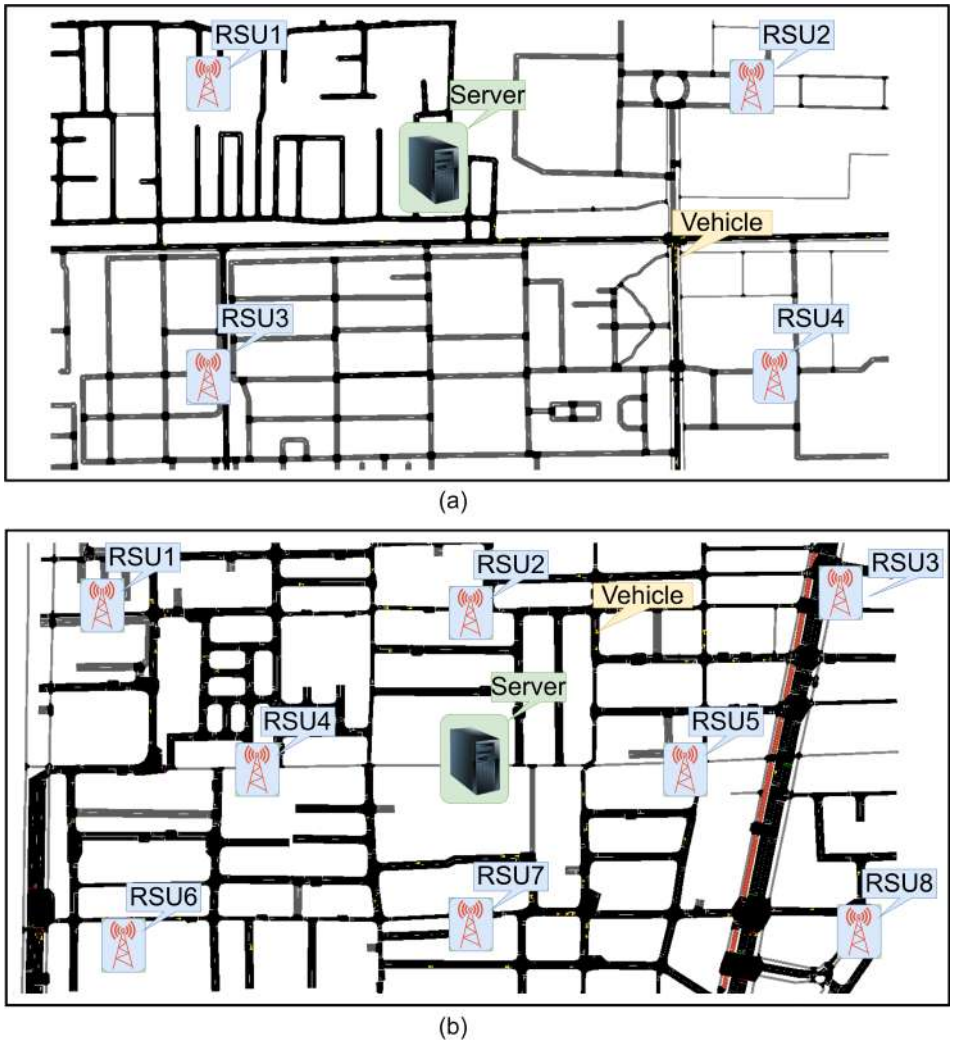


Figure 7.4: Maps for the scenario of (a) suburban and (b) urban area.

environments.

7.3.2 The Performance Results

This section presents a detailed analysis of the simulation outcomes with a focus on key network performance metrics. Specifically, it examines the transmission and processing delays associated with authentication messages, as well as the overall signaling overhead incurred by the system. To ensure a comprehensive evaluation, both suburban and urban traffic environments are modeled, enabling performance assessment under varying vehicular densities.

7.3.2.1 Delay Analysis and Impact of the Protocol Architecture

To evaluate the delay performance of the proposed Combined-Based Server (CBS) protocol, I conducted a comparative analysis of three key delay metrics: authentication delay, handover delay, and end-to-end delay. These metrics were compared against those from the previously established Kerberos-Blockchain (KBC) system [8] under two different vehicular network scenarios: a suburban environment comprising 100 vehicles and an urban environment comprising 200 vehicles. The results are illustrated in Figure 7.5.

In the suburban scenario, a notable reduction in authentication delay was observed, decreasing from 85 ms in the KBC approach to 52 ms in the CBS approach. In contrast, the handover delay remained unchanged at approximately 55 ms.

A similar pattern emerged in the urban scenario with 200 vehicles. The authentication delay in the CBS system was reduced from 124 ms to 69.99 ms, effectively bringing it below the commonly accepted VANETs threshold of 100 ms. However, no significant differences were observed in the handover delays, which remained consistent with those recorded in the suburban setting.

Across both scenarios, the reduction in authentication delay highlights the effectiveness of the CBS protocol in improving initial authentication efficiency. Notably, the previous KBC protocol exhibited authentication delays that exceeded the acceptable upper bound for VANETs applications, whereas the proposed CBS protocol successfully reduced the delay to within the required

limits. This improvement is primarily attributed to the architectural simplification introduced in CBS, which consolidates the AS and TGS functionalities into a single server.

By collapsing the AS and TGS roles, simplified architectures reduce the number of message exchanges required during authentication. This architectural streamlining results in lower authentication delay. Gao *et al.* [71] demonstrated that using Proxy Mobile IPv6 (PMIPv6) in VANETs reduces authentication delay during handovers due to localized mobility anchors. Similarly, Wang *et al.* [72] showed that the SREHA handover scheme reduces delay by removing redundant validation steps. These results indicate that fewer authentication stages translate into faster and more efficient processes.

However, the observed improvements are limited to the authentication delay metric. The handover and end-to-end delays remain unaffected, as the modifications introduced in CBS target only the initial authentication process and do not alter the mechanisms responsible for handover or data forwarding operations. It is noteworthy that the measured handover delay remains below the critical authentication delay threshold. Furthermore, the observed end-to-end delay consistently falls within the acceptable range defined by the ETSI TS 122 186 standard, which mandates that end-to-end delays for safety-related applications should not exceed 20 ms [64].

To evaluate the scalability limitations of the proposed system, I conducted additional experiments by incrementally increasing the number of vehicles from 100 to 400. In this evaluation, I focused exclusively on authentication delay, as previous results indicated that handover delay and end-to-end delay remained unaffected by changes in network density.

As illustrated in the Figure 7.6, the authentication delay remains below the acceptable threshold of 100 ms for up to 300 vehicles. However, when the number of vehicles reaches 400, the authentication delay exceeds the standard VANETs threshold, increasing to 105.87 ms. This result indicates that while the proposed system demonstrates strong performance under moderate traffic conditions, its scalability may be constrained under high-load scenarios.

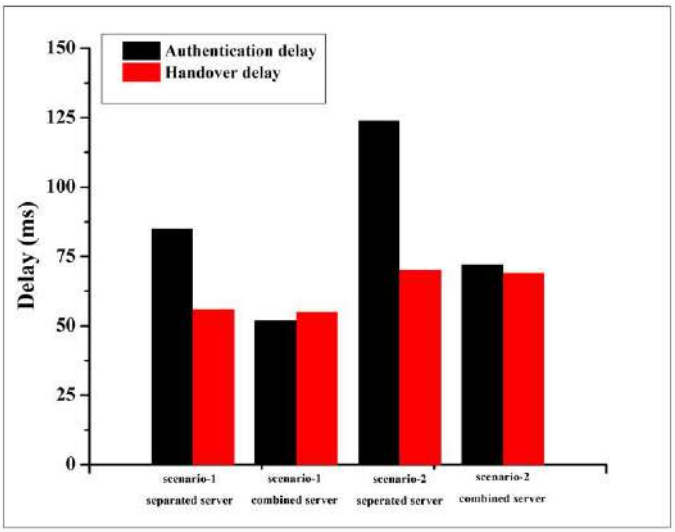


Figure 7.5: Effects of combined server on delays in suburban and urban scenario.

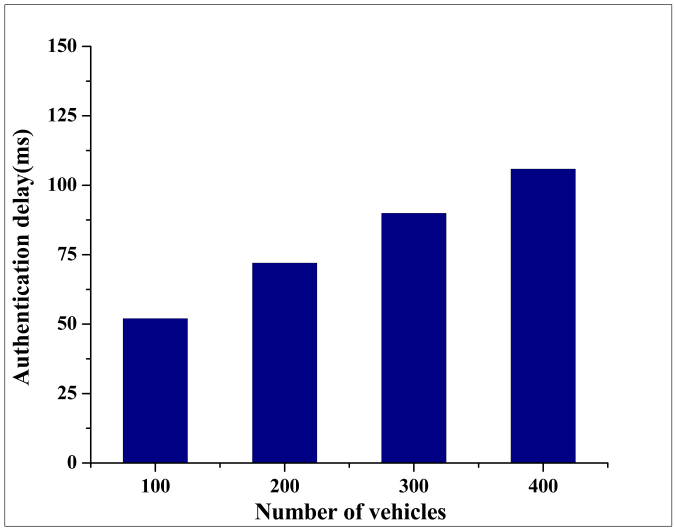


Figure 7.6: Effects of vehicle numbers on authentication delays in combined server urban scenario.

7.3.2.2 Signaling Overhead

signaling overhead refers to the control-related messages required to manage sessions, such as authentication, synchronization, and mobility updates. While essential, excessive signaling can consume bandwidth and processing time, reducing overall efficiency. In latency-sensitive environments like VANETs, minimizing signaling overhead is crucial to ensure faster response times and optimal resource usage. These signaling messages are not the main data (payload), but supporting messages including authentication protocol.

I compare the signaling overhead from our proposed method (CBS) with the former method, KBC. The signaling overhead that I take is counted in every increasing 20 number of vehicles of both approach. The signaling overhead for this proposed system is counted in the equation 7.7.

$$C_{KBC} = hop_{RSU-RSU}[a * Trans_u * L_{msg}] \tag{7.7}$$

where $hop_{RSU-RSU}$ denotes the average distance between adjacent Roadside Units (RSUs), a represents the weight coefficient assigned to a specific communication link, $Trans_u$ refers to the transmission unit, and L_{msg} indicates the total size of the messages exchanged during the signaling process. The message sizes used in the signaling evaluation for this study are summarized in Table 7.3, while the comparative analysis of signaling overhead with existing approaches is presented in Figure 7.7.

Table 7.3: Message size in the signaling process.

Message Parameter	**Size (bytes)**
Session Key	16
Vehicle ID	8
Timestamp (TS)	4
Ticket for Initial Authentication	8
HMAC	8
AES Input Bit Length	16
Lifetime	3
IP Address	16
Service Name	3

Figure 7.7 illustrates the comparative results of signaling overhead for both methods across increasing vehicle densities, with increments of 20 vehicles per simulation run. The CBS consistently demonstrates a reduced signaling over-

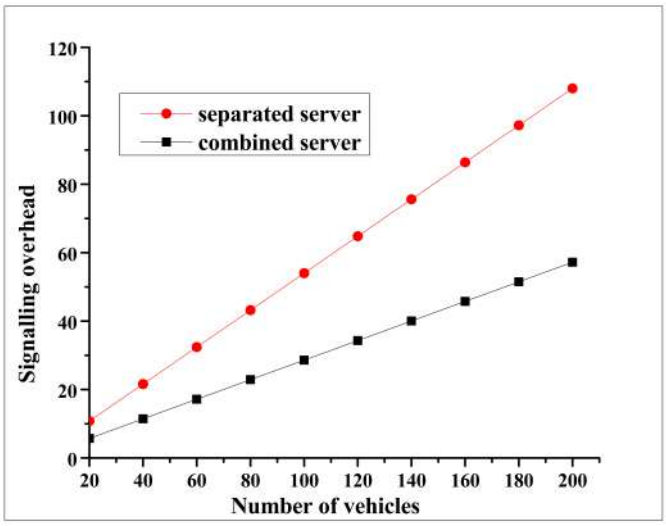


Figure 7.7: Comparison of signaling overhead between separated server and combined server.

head compared to KBC. For instance, at 100 vehicles, the signaling overhead for KBC is 54,000 bytes, while for CBS, it is only 28,600 bytes. This trend continues across all tested scenarios, indicating that CBS introduces a more efficient signaling mechanism.

The reduction in overhead is attributed to the architectural integration AS and TGS functionalities into a single server. This consolidation eliminates the need for inter-server communication, thus removing several control exchanges that were previously necessary in the KBC architecture. By simplifying the message flow and reducing redundancy, CBS minimizes the bandwidth consumption and processing delay associated with signaling, making it more suitable for high-density VANETs environments.

Overall, the proposed CBS method achieves a substantial reduction in signaling overhead, enhancing the protocol's scalability and responsiveness. This improvement is particularly critical in scenarios involving frequent mobility handovers and real-time authentication requirements [73].

7.3.2.3 Throughput

I evaluate the throughput performance of a KBC protocol [8] under two different server architectures: a *separated server configuration* with distinct AS and TGS, and a *combined server implementation*, wherein a single CBS fulfills both AS and TGS roles. Experiments were conducted for two network sizes (100 vehicles and 200 vehicles) to observe how each architecture scales with the number of vehicles. The throughput (in bits per second, bps) was measured at the authentication servers, and I recorded the average sustained throughput as well as the minimum and maximum observed values over the measurement period for each scenario. Table 7.4 summarizes the results, comparing the separated and combined configurations for 100 and 200 vehicles.

Table 7.4: Throughput comparison between AS and TGS in KBC [8] with Combined Server (CBS).

Vehicles	**AS in KBC (bps)**	**TGS in KBC (bps)**	**CBS (bps)**
100 Vehicles			
Average	51.95	261	627
Max	58.48	294	4316
Min	38.52	58	501
200 Vehicles			
Average	56.00	287	588
Max	108.21	545.6	4316
Min	52.35	59.52	458

As shown in Table 7.4, the combined server (CBS) dramatically outperforms the separated KBC setup in terms of throughput. For the 100-vehicle scenario, the CBS achieves an average throughput of about 627bps, which is an order of magnitude higher than the AS (52bps) and TGS (261bps) throughputs in the separated configuration. Even when considering the sum of AS+TGS capacities (approximately $52 + 261 \approx 313$bps), the single CBS handles roughly double the throughput of the dual-server system.

A similar trend is observed for 200 vehicles: the combined server sustains 588bps on average, versus 56bps (AS) and 287bps (TGS) in the separated case. The gap in peak performance is even more pronounced — the CBS reaches

a maximum throughput of 4316bps (over 4kbps) in both the 100 and 200 vehicle tests, vastly exceeding the peak throughput recorded on the separated servers (e.g., the TGS peaked at 294bps for 100 vehicles). This stark difference underscores the efficiency advantage of the integrated approach. Several factors contribute to the higher throughput observed with the combined CBS architecture.

First, merging the AS and TGS roles eliminates the inter-server communication overhead inherent in the traditional Kerberos workflow [8]. In the separated design, each vehicle's authentication involves at least two sequential steps (AS followed by TGS), with network messages and processing delays between them. The CBS design removes this network round-trip between servers, reducing latency and CPU overhead per authentication request. With fewer context switches and message-passing delays, the server can complete each authentication transaction faster, effectively handling more requests per second.

Second, the processing is more *streamlined* in the combined implementation. Since a single server handles the entire authentication ticket issuance process, it can optimize the workflow internally (for example, by reusing cryptographic contexts or combining database lookups), rather than each server performing its own set of operations from scratch. This streamlining minimizes duplicated work and benefits overall throughput.

Third, resource utilization is improved through *resource consolidation*. Instead of two separate servers each running at partial load, the CBS concentrates computational resources into servicing a unified queue of requests. This consolidation ensures that the server's full capacity is leveraged for the combined task, avoiding the scenario where one server might be idle while the other is a bottleneck.

7.4 Summary

In this chapter, I presented an integrated authentication server architecture for VANETs security that unifies the Authentication Server (AS) and the Ticket Granting Server (TGS) components of a Kerberos-blockchain authentication system into a single Combined Blockchain Server (CBS). Our compre-

hensive evaluation implemented and tested the proposed system in simulated VANETs networks across both suburban and urban environments using OMNeT++ integrated with SUMO traffic simulator, while the blockchain component was implemented on the Ganache platform. Most critically, this proposal successfully maintains authentication delays well below the $100ms$ threshold required for VANETs operations, even in high-density urban scenarios, while eliminating the need for additional infrastructure as vehicle volumes increase. The integrated architecture achieved approximately 104% higher throughput and 45% lower signaling overhead compared to the original separated server approach. Furthermore, by addressing critical authentication delay requirements while enhancing scalability, our approach contributes significantly to the practical implementation of secure and efficient VANETs for next-generation Intelligent Transportation Systems.

For future work, I plan to further enhance our current CBS authentication to address the unique challenges of ultra-dense urban VANETs environments. This includes integrating lightweight cryptographic primitives specifically optimized for vehicular networks to further reduce authentication latency during high-traffic scenarios such as rush hours and major public events. Additionally, I aim to strengthen the security robustness of our system through VANETs-specific security enhancements including proof-of-location mechanisms to prevent position spoofing attacks.

Chapter 8

Conclusion and future works

This dissertation explored the design, evaluation, and optimization of a secure and efficient authentication framework for Vehicular Ad-hoc Networks (VANETs) by integrating Kerberos and blockchain technologies. The study begins with the implementation of a fundamental Kerberos-based scheme adapted for vehicular environments, followed by a security and performance evaluation of its blockchain-enhanced variant. It then expands into scenario-based simulations to assess system scalability across different traffic densities, and culminates in the development of a novel Combined Blockchain Server (CBS) that unifies core authentication components to address infrastructure overhead. Each stage of the research builds upon the previous, offering a an understanding of the trade-offs between security, latency, and scalability in VANET authentication systems.

Firstly, I presented the fundamental implementation of a Kerberos-based authentication scheme applied to VANET environments. It utilizes a classic two-tier structure consisting of an Authentication Server (AS) and Ticket Granting Server (TGS), adapted from traditional Kerberos, to serve vehicular communications. For evaluations, the proposed system meets the required authentication delay threshold of under 100 ms, even as the number of vehicles increases. The results show that the signaling overhead remains lightweight due to the elimination of repeated connections to the Trusted Authentication Server (TAS) during handover. This confirms the system's potential to provide secure and efficient authentication for dynamic vehicular environments.

Secondly, I presented a comprehensive performance and security evaluation of the integrated Kerberos–Blockchain authentication system for VANETs. It works by leveraging Ethereum blockchain as the foundation for secure authentication message that generated by OMNeT++. I assessed both the system's practicality and implementation feasibility in terms of transaction execution and gas costs. The results show that the practicality and met implementation requirements regarding transaction costs includes gas costs and the memory size.

Thirdly, I presented a simulation-based study evaluating the practicality

and scalability of the proposed Kerberos–Blockchain authentication system in diverse network scenarios. It works by embedding authentication messages within blockchain blocks and assessing performance across suburban and urban environments. For evaluations, I simulated three scenarios: (1) a suburban setting with 100 vehicles and one TAS, (2) an urban environment with 200 vehicles but without TAS addition, and (3) the same urban scenario with one additional TAS. The results show that while the system met all requirements in the suburban case, it failed to maintain acceptable authentication delays under high vehicle density without infrastructure scaling. However, the addition of a second TAS successfully restored performance to within VANET standards. The benefits of this work lie in validating the system's applicability across different mobility patterns and confirming blockchain's scalability with manageable gas values and memory usage, even as the number of vehicles increased.

Lastly, I presented an integrated server design that unifies the Authentication Server (AS) and Ticket Granting Server (TGS) into a Combined Blockchain Server (CBS) to optimize VANET authentication. It works by simplifying the Kerberos-based architecture into a single entity to reduce delay and improve scalability in both suburban and urban environments. For evaluations, the system was implemented using OMNeT++ with SUMO for VANET simulation and Ganache for blockchain deployment. The results show that the CBS-based system consistently maintained authentication delays below the 100 ms requirement, even in dense urban scenarios. Compared to the separated AS–TGS structure, it achieved around 104% higher throughput and 45% lower signaling overhead. The benefits of this work include reducing infrastructure complexity while ensuring timely, secure authentication for next-generation Intelligent Transportation Systems.

For future work, I plan to further enhance our current CBS authentication to address the unique challenges of ultra-dense urban VANETs environments. This includes integrating lightweight cryptographic primitives specifically optimized for vehicular networks to further reduce authentication latency during high-traffic scenarios such as rush hours and major public events.

References

[1] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine,* vol. 46, no. 6, pp. 164–171, Jun. 2008.

[2] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security,* vol. 15, no. 1, pp. 39–68, 2007.

[3] A. Wasef and X. Shen, "EMAP: Expedite message authentication protocol for vehicular ad hoc networks," *IEEE Transactions on Mobile Computing,* vol. 12, no. 1, pp. 78–89, Jan. 2013.

[4] Rahayu, M, Hossain, M.B.; Ali, M.A.; Huda, S.; Kodera, Y.; Nogami, Y., "An Integrated Secured Vehicular Ad-Hoc Network Leveraging Kerberos Authentication and Blockchain Technology," in *Proc. CANDARW,* 2023.

[5] M. Gerlach, "Assessing and improving privacy in VANETs," in *Proc. VTC Spring,* 2007.

[6] Rahayu, M., Hossain, M.B., Ali, M.A., Huda, S., Kodera, Y., Nogami, Y. “An In-depth Analysis of Kerberos and Blockchain Integration on VANETs’ Security and Performance,”. In Proceedings of the IEEE International Conference on Consumer Electronics, Taichung, Taiwan, 9–11 July 2024.

[7] K. Rabieh *et al.*, "Efficient certificate revocation management in VANET," *Ad Hoc Networks,* vol. 55, pp. 61–76, 2017.

[8] M. Rahayu, M. B. Hossain, S. Huda, Y. Kodera, M. A. Ali, and Y. Nogami, “The Design and Implementation of Kerberos-Blockchain Vehicular Ad-Hoc Networks Authentication Across Diverse Network Scenarios,” *Sensors*, vol. 24, no. 23, Art. no. 7428, 2024, doi: 10.3390/s24237428.

[9] Rahayu, M, Hossain, M.B.; Ali, M.A.; Huda, S.; Kodera, Y.; Nogami, Y., "Integrated Authentication Server Design for Efficient Kerberos-Blockchain VANET Authentication," *IEEE Access*, 2025 (submitted).

[10] X. Lin *et al.*, "TSVC: Efficient secure and privacy-preserving vehicular communications with trustworthiness services in VANETs," *IEEE Transactions on Parallel and Distributed Systems,* vol. 22, no. 12, pp. 2108–2123, Dec. 2011.

[11] S. Rashid, A. Audah, L. Hamdi, M. M. Dehuri, B. S. Prasad Mishra, P. K. Mallick, and S. B. Cho, "Intelligent Transportation Systems (ITSs) in VANET and MANET," in *Biologically Inspired Techniques in Many Criteria Decision Making, Smart Innovation, Systems and Technologies*, vol. 271, Singapore: Springer, pp. 6–59, 2022.

[12] Ali, R. Liu, R. Nayyar, A.; Waris, I.; Li, L.; Shah, M.A, "Intelligent Driver Model-Based Vehicular Ad Hoc Network Communication in Real-Time Using 5G New Radio Wireless Networks," *IEEE Access* 2023, *11*, 4956–4971.

[13] Bhanu, C., Wiley, Hoboken. NJ, " Challenges, Benefits and Issues: Future Emerging VANETs and Cloud Approaches". In *Cloud and IoT-Based Vehicular Ad Hoc Networks*; USA , 2021; pp. 142–149.

[14] F. Qu, Z. Wu, F. Wang, and W. Cho, "A Security and Privacy Review of VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 2985–2996, 2015, doi:10.1109/TITS.2015.2444433.

[15] R. Ali, R. Liu, A. Nayyar, I. Waris, L. Li and M. A. Shah, "Intelligent Driver Model-Based Vehicular Ad Hoc Network Communication in Real-Time Using 5G New Radio Wireless Networks," IEEE Access, vol. 11, pp. 4956-4971, 2023.

[16] S. A. Jan, N. U. Amin, M. Othman, M. Ali, A. I. Umar, and A. Basir, "A Survey on Privacy-Preserving Authentication Schemes in VANETs: Attacks, Challenges and Open Issues," *IEEE Access*, vol. 9, pp. 153701–153726, 2021, doi:10.1109/ACCESS.2021.3128553.

[17] M. A. R. Baee, L. Simpson, X. Boyen, E. Foo, and J. Pieprzyk, "Authentication strategies in vehicular communications: A taxonomy and framework," *Veh. Commun.*, vol. 28, Art. 100312, 2021, doi:10.1016/j.vehcom.2020.100312.

[18] Manivannan, D., Moni, S.S., Zeadally, S. "Secure authentication and privacy-preserving techniques in Vehicular Ad-Hoc Networks (VANETs),". *EURASIP J. Wirel. Commun. Netw.* 2020, *25*, 100247.

[19] Ma, Y., Ning, H. "The Improvement of Wireless LAN Security Authentication Mechanism Based on Kerberos,". In Proceedings of the 2018 International Conference on Electronics Technology, Chengdu, China, 23–27 May 2018; pp. 392–397.

[20] Singh, A, K., Grover, J., Mishra, S."Integration of Blockchain in VANET Using gRPC for Privacy Preservation of Vehicles,". *SN Comput. Sci.* 2023, *5*, 110.

[21] Wang, F., Xu, Y., Zhang, H., Zhang, Y., Zhu, L. "2FLIP: A Two-Factor Lightweight Privacy-Preserving Authentication Scheme for VANET,". *IEEE Trans. Veh. Technol.* 2016, *65*, 896–911.

[22] Wang, P., Liu, Y. "SEMA: Secure and Efficient Message Authentication Protocol for VANETs,". *IEEE Syst. J.* 2021, *15*, 846–855.

[23] Lee, J., Kim, G., Das, A.K., Park, Y. "Secure and Efficient Honey List-Based Authentication Protocol for Vehicular Ad Hoc Networks,". *IEEE Trans. Netw. Sci. Eng.* 2021, *8*, 2412–2425.

[24] Gürfidan, R., Açıkgözoğlu, E. "A New Blockchain-Based Authentication Infrastructure for Wireless Networks: BCAUTH,". *21. Yüzyıl Fen Tek. Derg.* 2023, *10*, 9–17.

[25] Li, X., Jing, T., Li, R., Li, H., Wang, X., Shen, D. "BDRA: Blockchain and Decentralized Identifiers Assisted Secure Registration and Authentication for VANETs,". *IEEE Internet Things J.* 2023, *10*, 12140–12155.

[26] Son, S., Lee, J., Park, Y., Park, Y., Das, A.K. "Design of Blockchain-Based Lightweight V2I Handover Authentication Protocol for VANET,". *IEEE Trans. Netw. Sci. Eng.* 2022, *9*, 1346–1358.

[27] Sang, G., Chen, J., Liu, Y., Wu, H., Zhou, Y., Jiang, S. "PACM: Privacy-Preserving Authentication Scheme With on-Chain Certificate Management for VANETs,". *IEEE Trans. Netw. Serv. Manag.* 2023, *20*, 216–228.

[28] National Institute of Standards and Technology (NIST), Data Encryption Standard (DES); Federal Information Processing Standards Publication 46-1; NIST: Gaithersburg, MD, USA, 1988. Available online: `https://csrc.nist.gov/pubs/fips/46-1/final` (accessed on 19 November 2024).

[29] A. Guerna, S. Bitam, and C. T. Calafate, "Roadside Unit Deployment in Internet of Vehicles Systems: A Survey," *Sensors*, vol. 22, no. 9, Art. no. 3190, 2022, doi: 10.3390/s22093190.

[30] S. Xue, S. Gong, and X. Li, "A Comparative Study of IEEE 802.11bd and IEEE 802.11p on the Data Dissemination Properties in Dynamic Traffic Scenarios," *Applied Sciences*, vol. 14, no. 5, p. 2099, 2024, doi: 10.3390/app14052099.

[31] D. Kalaivani, "VANET: Framework, Challenges and Applications," *Indian Journal of Data Communication and Networking*, vol. 1, no. 2, pp. 12–15, 2021.

[32] A. Yadav and V. K. Yadav, "Survey on VANET authentication scheme based on cryptographic protocols," in *Innovative Computing and Communications (ICICC 2024)*, Lecture Notes in Networks and Systems, vol. 1024, Springer, pp. 85–104, 2024.

[33] S. Abbas, M. A. Talib, and A. Ahmed, "Blockchain-based authentication in Internet of Vehicles: A survey," *Sensors*, vol. 21, no. 23, Art. 7927, 2021, doi:10.3390/s21237927.

[34] P. Cirne, A. Zúquete, S. Sargento, and M. Luís, "The impact of ECDSA in a VANET routing service: Insights from real data traces," *Ad Hoc Networks*, vol. 90, Art. 101799, 2019, doi:10.1016/j.adhoc.2018.08.017.

[35] World Bank, "Digital certificates and PKI," in *Identification for Development (ID4D) Guide*, 2021.

[36] J. G. Steiner, C. Neuman, and J. I. Schiller, "Kerberos: An Authentication Service for Open Network Systems," in *USENIX Winter Conference*, Dallas, TX, USA, 1988, pp. 191–202.

[37] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos Network Authentication Service (V5)," Internet Engineering Task Force (IETF), RFC 4120, Jul. 2005. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4120

[38] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Pearson, 2017.

[39] R. T. Prapty, S. Jakkamsetti, and G. Tsudik, "KESIC: Kerberos Extensions for Smart, IoT and CPS Devices," *arXiv preprint arXiv:2407.04880*, 2024. [Online]. Available: https://arxiv.org/abs/2407.04880

[40] J. Chen, Y. Qiu, Y. Zhang, Z. Ning, and Y. Liu, "DKSM: A Decentralized Kerberos Secure Service-Management Protocol for Internet of Things," *Internet of Things*, vol. 23, 100310, 2023, doi: 10.1016/j.iot.2023.100310.

[41] OMNeT++ (6, "2024," 1). . Available online: https://omnetpp.org/ (accessed on 12 August 2024).

[42] INET (4, "2023," 5.2). . Available online: https://inet.omnetpp.org/ (accessed on 15 August 2023).

[43] Christoph Sommer, Reinhard German, and Falko Dressler, *Veins – Vehicles in Network Simulation* (official documentation), 2024. [Online]. Available: https://veins.car2x.org.

[44] Institute of Transportation Systems, DLR, *Eclipse SUMO – Simulation of Urban Mobility* (official documentation), 2024. [Online]. Available: https://sumo.dlr.de.

[45] Truffle (5, "2023," 11.0). . Available online: https://archive.trufflesuite.com/truffle/ (accessed on 17 September 2023).

[46] Ganache (2, "2023," 7.1). . Available online: https://archive.trufflesuite.com/docs/ganache/ (accessed on 22 September 2023).

[47] C. Huang, M. S. Chiang, and D. Dao, "A Group-based Handover Control Scheme for Mobile Internet Using the Partially Distributed Mobility Management (GP-DMM) Protocol," in *Proc. Int. Symp. Pervasive Systems,*

Algorithms and Networks (I-SPAN), 2017, pp. 96–101, doi:10.1109/I-SPAN.2017.19.

[48] C. Lai and Y. Ding, "A Secure Blockchain-Based Group Mobility Management Scheme in VANETs," in *Proc. Int. Conf. Commun. China (ICCC)*, 2019, pp. 112–117, doi:10.1109/ICCChina.2019.8855941.

[49] N. Singh and M. Vardhana, "Computing Optimal Block Size for Blockchain based Applications with Contradictory Objectives," Third International Conference on Computing and Network Communications (CoCoNet'19), pp. 1389-1398, 2020.

[50] Panwar, A., Bhatnagar, V. "Distributed Ledger Technology (DLT): The Beginning of a Technological Revolution for Blockchain,". In Proceedings of the 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 28–29 February 2020; pp. 1–6.

[51] Wood, G. Ethereum: A Secure Decentralised Generalised Transaction Ledger; Yellow Paper. Available online: `https://ethereum.github.io/yellowpaper/` (accessed on 19 November 2024).

[52] Diallo, E., Al Agha, K. "Study and Design of Blockchain-based Decentralized Road Traffic Data Management in VANET,". In *Cryptography and Security*; Université Paris-Saclay: Paris, France, 2022; pp. 1–308.

[53] Abdelgadir, M., Saeed, R., Babiker, A. "Vehicular Ad-Hoc Networks (VANETs) Dynamic Performance Estimation Routing Model for City Scenarios,". In Proceedings of the 2016 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2–4 November 2016; pp. 1-8.

[54] Toulni, H., Miyara, M., Filali, Y., Koumetio Tekouabou, C.S. "Preventing Urban Traffic Congestion Using VANET Technology in Urban Area,". *E3S Web Conf.* 2023, *418*, 02005.

[55] Subramaniam, M., Rambabu, C., Chandrasekaran, G., Kumar, N.S. "A Traffic Density-Based Congestion Control Method for VANETs,". *Wirel. Commun. Mob. Comput.* 2022, *2022*, 7551535.

[56] Sommer, C., German, R., Dressler, F. "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Trans. Mob. Comput.* 2011, *10*, 3–15.

[57] Álvarez López, P., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E. "Microscopic Traffic Simulation using SUMO,". In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, November 2018; pp. 2575-2582.

[58] IEEE, IEEE 802.11p-2010: Wireless Access in Vehicular Environments (WAVE). Available online: `https://ieeexplore.ieee.org/document/5514475` (accessed on 19 November 2024).

[59] Ul Hassan, M., Al-Awady, A.A., Ali, A., Sifatullah; Akram, M., Iqbal, M.M., Khan, J., Abdelrahman Ali, Y.A. "ANN-Based Intelligent Secure Routing Protocol in Vehicular Ad Hoc Networks (VANETs) Using Enhanced AODV,". *Sensors* 2024, *24*, 818.

[60] Deng, X., Gao, T., Guo, N., Qi, J., Zhao, C. "PAS: Privacy-Preserving Authentication Scheme Based on SDN for VANETs,". *Appl. Sci.* 2022, *12*, 4791.

[61] He, Z., Fu, B., Cao, A., Yu, J. "A Solution for Mobility Management in Software Defined VANET,". In Proceedings of the 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Chengdu, China, 9–12 October 2018; pp. 553–558.

[62] Shorfuzzaman, M., Masud, M., Rahman, M.M. "Characterizing end-to-end delay performance of randomized TCP using an analytical model.". *Int,J. Adv. Comput. Sci.* 2016, *7*, 406–412.

[63] Khan, A,S., Balan, K., Javed, Y., Tarmizi, S., Abdullah, J. "Secure Trust-Based Blockchain Architecture to Prevent Attacks in VANET,". *Sensors* 2019, *19*, 4954.

[64] ETSI TS 122 186, "Service Requirements for Enhanced V2X Scenarios," 2018. . V15.4.0. Available online: `https://www.etsi.org/deliver/ets`

i_ts/122100_122199/122186/15.04.00_60/ts_122186v150400p.pdf (accessed on 28 July 2024).

[65] Khan, R,L., Singh, R., Vijay, R., Kumar, R., Singh, A., Ather, A. "Evaluating the Impact of Different Routing Protocols on VANET Performance,". In Proceedings of the 2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 22–23 December 2023; pp. 308–314.

[66] Shaji, K, A. "Performance Analysis of Authentication and Efficient and Secure Message Communication in VANETs,". *Int. J. Sci. Technol. Res.* 2020, *9*, 3021–3025.

[67] Asra, S, A. "Security Issues of Vehicular Ad Hoc Networks (VANET): A Systematic Review,". *TIERS Inf. Technol. J.* 2022, *3*, 17–27.

[68] Jadoon, A, D., Khan, Q., Ilahi, A.T., Iqbal, W. "A Survey on Security Challenges in VANET,". *Int. J. Comput. Sci. Inf. Secur. Sensors* 2016, *14*, 217–219.

[69] Malik, N., Nanda, P., Arora, A., He, X., Puthal, D. "Blockchain-Based Secured Identity Authentication and Expeditious Revocation Framework for Vehicular Networks,". In Proceedings of the 12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 674–679.

[70] Lai, J., Zhang, X., Liu, S., Moshayedi, A.J. "Blockchain-Based VANET Edge Computing-Assisted Cross-Vehicle Enterprise Authentication Scheme,". Available online: https://ssrn.com/abstract=4591102 (accessed on 28 October 2024).

[71] W. Gao, Y. Zhang, and J. Wei, "An Anonymous Authentication Scheme Based on PMIPv6 in VANETs," *IEEE Access*, vol. 6, pp. 50243–50252, 2018.

[72] X. Wang, H. Li, and Y. Huang, "SREHA: Social-Relay Edge Handover Authentication for VANETs," *IEEE Access*, vol. 12, pp. 56721–56734, 2024.

[73] S. Khan, M. A. Jan, M. Alam, S. Aslam, and S. W. Kim, "Scalable and Secure Authentication Scheme for VANETs Using Edge Computing," *IEEE Access*, vol. 9, pp. 78945–78958, 2021.