Article

Improving Vehicular Network Authentication with Teegraph: A Hashgraph-Based Efficiency Approach

Rubén Juárez * , Ruben Nicolas-Sans , José Fernández Tamámes

Posted Date: 3 June 2025

doi: 10.20944/preprints202506.0236.v1

Keywords: VANET; Teegraph; MANET; authentication; MinBFT; IoT; security; Ad-Hoc ; vehicular networks

Article

Improving Vehicular Network Authentication with Teegraph: A Hashgraph-Based Efficiency Approach

Rubén Juárez [1]*, Ruben Nicolas-Sans [2] and José Fernández Tamámes [2]

[1] Universidad Alfonso X "El Sabio"
[2] Universidad UNIE
* Correspondence: rjuarcad@uax.es; Tel.:+34647942856

Abstract: Vehicular ad hoc networks (VANETs) are a critical aspect of intelligent transportation systems, improving safety and comfort for drivers. These networks enhance the driving experience by offering timely information vital for safety and comfort. Yet, VANETs come with their own set of challenges concerning security, privacy, and design reliability. Traditionally, vehicle authentication occurred every time it entered the domain of the Road Unit (RSU). In our study, we suggest authentication should take place only when a vehicle hasn't covered a set distance, increasing system efficiency. The rise of the Internet of Things (IoT) has seen an upsurge in the use of IoT devices across various fields, including smart cities, healthcare, and vehicular IoT. These devices, while gathering environmental data and networking, often face reliability issues without a trusted intermediary. IBM anticipates blockchains, having evolved over the past decade, to be pivotal in IoT technology. Our study delves deep into implementing Teegraph in VANETs to enhance authentication. Given VANET's integral role in Intelligent Transportation Systems and its inherent challenges, we turn to Hashgraph—an alternative to blockchain. Hashgraph offers a decentralized, secure, and trustworthy database. We introduce an efficient authentication system, which triggers when a vehicle hasn't traversed a set distance, optimizing system efficiency. Moreover, we shed light on the indispensable role Hashgraph can occupy in the rapidly expanding IoT landscape. Lastly, we present Teegraph, a novel Hashgraph-based technology, as a superior alternative to blockchain, ensuring a streamlined, scalable authentication solution. Our approach leverages the logical key hierarchy (LKH) and packet update keys to ensure data privacy and integrity in vehicular networks.

Keywords: VANET; Teegraph; MANET; authentication; MinBFT; IoT; security; Ad-Hoc ; vehicular networks

1. Introduction

The rising need for enhancements in road safety and optimization has ignited significant attention towards self-organizing vehicular networks, a domain akin to Mobile Ad-Hoc Networks [1]. Vehicles in VANETs, equipped with On-Board Units, serve as the pivotal nodes facilitating vehicle-to-vehicle and vehicle-to-MSW communications [2]. While modern disruptive technologies like blockchain seem promising for the autonomous vehicles landscape, several aspects need to be ironed out to realize this vision, such as the establishment of a robust communication framework between vehicles and their surroundings, a niche where Internet of Things (IoT) is proving pivotal [3].

Blockchain's decentralized nature is well-suited for secure vehicular communication, an imperative given the threats posed by malicious entities in the network [4]. Many contemporary studies, like the one by Doe et al., emphasize the use of blockchain-based strategies, like key management systems, to bolster the security of VANETs against an array of attacks, such as the stealth and key tampering attacks [5].

The emerging Teegraph algorithm, pivoted on the Hashgraph structure, promises swift authentication processes which are fundamental in VANET contexts [6]. Such expedited authentication mechanisms, embedded with privacy traits, are not just pivotal in ensuring swift issue resolutions but also instrumental in safeguarding vehicular privacy against a range of cyber threats [7]. There's a significant emphasis on the Trust Authority's role in mitigating risks associated with vehicular vandalism by deleting pertinent vehicle and RSU data [8].

There's a significant body of research emphasizing the integration of 3G/4G/5G technologies in tandem with ultra-reliable low-latency networks. These explorations largely center around efficient traffic management [9–11]. An interesting study by Lee et al. introduces an architecture founded on Multiple Access Edge Computing (MEC), facilitating BBU-centric interactions amongst autonomous vehicles [12]. But such novel approaches don't come without their share of challenges, such as the surge in developmental costs attributed to limited terrestrial resources and the need for infrastructural investments [13].

In the context of vehicular technologies, the limitations posed by existing systems can be the bedrock for further research and innovation. A synergetic integration of varied technological approaches can possibly usher in a more resilient communication framework for autonomous vehicles, furthering the goal of enhanced road safety and traffic optimization [6].

Given the challenges and conflicts observed, the main contributions of this document are summarized below. The main objective of our work is to improve the quality of service (QoS) to increase the secure exchange of messages between peers in autonomous vehicles using hashgraphs. The key contributions of this paper are as follows:

We advocate the use of a hashgraph to reinforce the security of the system. A hashgraph works like a connected graph of hash values, where each member is responsible for exchanging messages over a network. Point-to-point communication between nodes is encrypted with hashes and protected by the Asynchronous Byzantine Fault Tolerance (ABFT) algorithm, ensuring fair participation of all nodes in a network for the exchange of messages. This approach will also help reduce the overall complexity of the system. Furthermore, we perform a comparative analysis between blockchain and hashgraph technology to validate the efficiency of the proposed framework. This study primarily focuses on designing a highly efficient consensus algorithm for Blockchain IoT, named VANET-Teegraph. This algorithm includes a gossip-based messaging protocol to generate a Directed Acyclic Graph (DAG)-based data structure for an efficient consensus process. Additionally, VANET: Teegraph employs a TEE-based "own parent only use" mechanism, ensuring that an IoT device on VANETs will always successfully send messages. The design also ensures a dynamic change of the consensus mechanism to accommodate scenarios where swarms of devices split or unite for various tasks. In situations where no new transactions are created, VANET-Teegraph serves as a resource-saving mechanism, reducing communication overhead and conserving storage space.

This document is structured as follows to provide a complete and detailed understanding of the topic:

- **Section II:** This section delves into the various studies and works that have been carried out previously in this field, providing an overview of existing knowledge and identifying gaps that this research aims to fill.
- **Section III:** Here, we discuss the existing system in detail, highlighting its strengths and pointing out its limitations. This analysis serves as the basis for understanding the need for our proposed improvements.
- **Section IV:** In this section, we present a complete description of the system. We describe its structure, functions, and key components, allowing the reader to understand the full picture of the theme.
- **Section V:** This section is dedicated to the proposed scenario. We present our proposed solutions, modifications or improvements to the existing system, based on the limitations identified in Section III.
- **Section VI:** We present the results obtained from the implementation of the proposed scenario in Section V. This includes any data, observations or conclusions drawn from the application of our proposed scenario.

- **Section VII:** In the final section, we summarize the key findings of our research and discuss possible future directions. We propose new avenues of research that could further expand our findings and contribute to the field.

2. Presents a review of related works

The main purpose of VANET is to send security alerts and emergency messages to reassure drivers within the vehicle environment [17]. Subsequent research has focused on ensuring vehicle privacy and security. The pseudonym method effectively protects the privacy of the vehicle while maintaining its security [18]. As a result of this research, vehicle certification times have been reduced, making the vehicle more responsive.

The security of vehicle messages is facilitated through the blockchain. Blockchain-based VANETs have been proposed to address issues such as message identity, authenticity, validity, and reliability [19]. The consortium's blockchain technology, i.e. Proof of Work, provides a decentralized, secure, and reliable database. These algorithms can also be implemented in blockchain systems.

It is isolated from other parts of the system and can offer security features such as isolated execution and application integrity.[16]

In VANET, TEEs could be used to improve the security and efficiency of communication and consensus within the network [20]. This ensures that the malicious replica does not cause different good replicas to perform different operations as its i-th operation. A2M provides the programming abstraction of a trusted record, which leads to ambiguity-immune protocol designs, i.e., the ability for a faulty host to lie in different ways to different clients or servers. Teechain is a new off-chain payment protocol that uses TEE to perform secure, efficient, and scalable funds transfers over a blockchain, with asynchronous access to the blockchain [21].

This inspires our follow-up work, focusing on the scalability of blockchains. They also propose a consensus algorithm called Test of Luck, which uses the TEE platform's random number generator to pick a consensus leader. DAG-based blockchains can also be improved with the help of TEE [**?**]. Furthermore, all of the previous work does not support the dynamic change of consensus topics, which is necessary for IoT scenarios.

Therefore, in this paper, we design an innovative way to combine TEE and DAG technology for the blockchain consensus algorithm used in IoT scenarios. This combination would enable faster and more secure consensus between nodes, including vehicles and roadside units, while providing greater resistance to malicious attacks and dynamically changing consensus topics. The figure below reflects the stages of the process in Teegraph, including transaction packaging, random neighbor selection, event forwarding, local DAG update, and consensus decision at each node.(Figure 1)

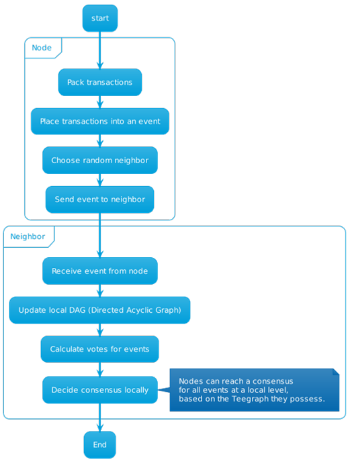

start
Node
Pack transactions
Place transactions into an event
Choose random neighbor
Send event to neighbor
Neighbor
Receive event from node
Update local DAG (Directed Acyclic Graph)
Calculate votes for events
Decide consensus locally
Nodes can reach a consensus
for all events at a local level,
based on the Teegraph they possess.
End

Figure 1. Reflects how nodes in Teegraph package transactions, select random neighbors, dispatch events, update the DAG locally, and make consensus decisions based on computed votes.

Teechain is a new off-chain payment protocol that uses TEE to perform secure, efficient, and scalable funds transfers over a blockchain, with asynchronous access to the blockchain [14]. Teegraph, a highly efficient consensus algorithm, is based on a gossip message protocol and uses a TEE-based "own parent only use" mechanism [15]. With Teegraph, nodes can reach a consensus for all events locally, based on the Teegraph they own [15].

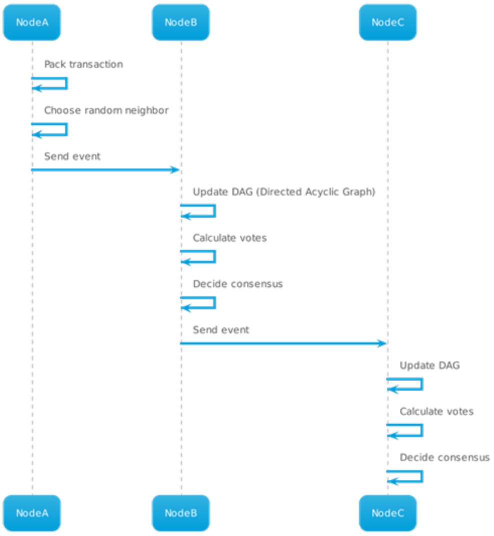


Figure 2. Reflects how nodes in Teegraph package transactions, select random neighbors, send events, update the DAG locally, and make consensus decisions based on calculated votes.

In summary (Figure 2), research in VANET has moved towards the use of blockchain and technologies such as TEE to improve security, privacy and efficiency in vehicle communication and key management. As technology advances, it is crucial to address existing limitations and develop more efficient and secure solutions for networks.

3. Explains the existing system, highlighting its advantages and limitations

Existing vehicle security systems cover registration, key generation, signature creation, verification and monitoring. These processes are based on collaboration between vehicles, roadside units (RSU) and a trusted authority (TA). Here is the flow of the system (Figure 3).

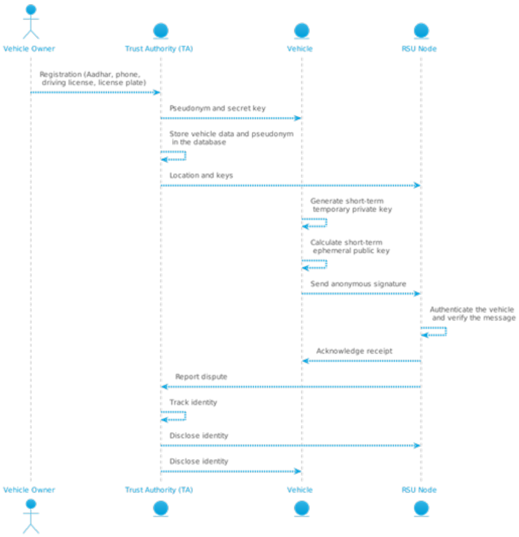


Figure 3. Shows the deployment diagram.

- **Registration:** Prior to the implementation of the network, vehicles and RSU must be registered with the AT. Vehicle owners submit their Aadhar card, mobile phone number, driver's license, registration number and other details to the TA.
- **Key Generation:** Upon entering the range of an RSU, the vehicle selects a random number as a short-term, temporary private key and uses it to compute a short-term, ephemeral public key.
- **Signature Creation:** To verify and maintain the integrity of the message, the vehicle creates a short-term anonymous signature using the short-term anonymous key and sends it to the RSU. If not, it discards the message and informs the TA for further action.
- **Tracking:** In the event of a dispute between a vehicle and an RSU, the TA tracks the true identity in its database and reveals it to all RSUs and vehicles, thus nullifying the privacy of malicious users, whether they are vehicles or RSUs, to avoid further damage (**Table** 1).

Table 1. Summarizes the risks and their effects on VANETs.

Risk	Effect on VANETs
privacy attacks	Compromise the privacy of interested parties
Data integrity failures	Affects the reliability and security of data
cyber attacks	Causes network failures and malfunctions.
accidents	Causing property damage and injury
loss of life	Deaths caused by accidents

To deal with these threats and keep a vehicular network free from attackers, key management techniques have been introduced. However, key management alone in a centralized network is not effective in ensuring system security. Our goal is to simplify the system and achieve faster information exchange on VANETs. To do this, we have introduced a hashgraph-based approach to key management in effective vehicle communication. Due to the inherent properties of hashgraphs, the stored information can be verified and validated on the fly.

This existing system can be enhanced using advanced technologies such as Teegraph, which optimizes consensus between nodes and improves safety and efficiency in vehicular communication.

The VANET-TeeGraph algorithm, used within vehicular networks, offers both advantages and disadvantages. Among its strengths, the system provides enhanced security due to its employment of bilinear pairing for the elliptic curve in authentications, a method renowned for its superior protection [**?**]. Utilizing hashgraph-based TeeGraph technology, it ensures secure data storage, bolstering efficiency and responsiveness against potential attacks on VANETs [**?**]. Moreover, the system boasts an effective tracking mechanism capable of identifying malicious vehicles, revoking their identities among all other vehicles, thereby effectively mitigating future harm [25]. Additionally, the utilization of pseudonyms preserves the privacy of each vehicle by safeguarding users' true identities [26].

However, some limitations are evident. For instance, vehicles require frequent authentication every time they come within the vicinity of an RSU, burdening both RSUs and leading to increased vehicle wait times [27]. A consistent treatment of all messages from vehicles to RSUs could result in delays for crucial emergency communications, especially in high-speed vehicular scenarios [28]. RSUs also face a high computational demand because of the intricate operations needed to append blocks to the blockchain and validate transactions [29]. Furthermore, while the integration of TeeGraph and Hashgraph offers security and communication enhancements, it can also augment system intricacy and extend processing durations [30].

In pursuit of robust vehicular network solutions, while the VANET-TeeGraph algorithm brings several commendable attributes to the table, it also introduces challenges that require attention [31]. Hence, ongoing research and solution development are pivotal to tackle security, privacy, and efficiency issues in VANETs. The convergence of innovative technologies like TeeGraph and Hashgraph signifies substantial progress, but optimizing performance and assuring vehicular network security still demand concerted effort [**?**].

4. Provides an overview of the system.

In this section, we will delve into the system model, bilinear matching, elliptic curve cryptography (ECC), MinBFT, and the Vanet-Teegraph algorithm. These are the key components used in the proposed method and are developed below.

4.1. System Model

In VANETs, each vehicle is equipped with an On-Board Unit (OBU) allowing communication with other vehicles, known as Vehicle-to-Vehicle (V2V) communication [33]. VANETs comprise three primary components: OBU, RSU, and Trust Authority (TA) [34].

- **OBU:** Vehicles incorporate an OBU that functions as a transceiver, facilitating communication with other vehicles' OBUs and RSUs [35]. Each OBU contains a Tamper-Proof Device (TPD) to house sensitive data such as authentication keys from the TA, alongside sensors like the GPS for location [36], and an Event Data Recorder (EDR) to log vehicle accident data [37].
- **RSU:** RSUs, as stationary units, are situated alongside roads or at specific points such as crossroads or parking areas [38]. They perform diverse functions, from vehicle authentication and Teegraph message storage to maintaining a confidential ledger with vehicle details and its verification status [39]. They also communicate with proximate RSUs and vehicles within their range and report malevolent activities to the TA [**?**].
- **TA:** The TA oversees preserving trust and security throughout all VANETs, which includes the registration of vehicles and RSUs [41]. It upholds a primary database with vehicular information and a secondary one for saving vehicle and verification keys [42].

Beyond these pivotal components, VANETs also utilize sophisticated algorithms and technologies to enhance network security, privacy, and efficiency [43]. The induction of Teegraph technology enables the formation of a Directed Acyclic Graph at a local level in each node, promoting local consensus on network incidents [44]. The VANET communication and verification process adhere to a particular flow engaging all primary components [45].

4.2. Bilinear Pairing

In the realm of VANETs, bilinear pairing plays a pivotal role in augmenting authentication and the security of communication between vehicles and RSUs [**?**]. The application of bilinear pairing in the VANET-Teegraph system ensures enhanced security in authentication and communication amidst vehicles and Roadside Units, concurrently preserving vehicular privacy through pseudonyms [47]. This privacy safeguarding is achieved by introducing pseudonyms assigned to distinct vehicles [48]. Employing mathematical groups in conjunction with bilinear pairing not only refines data transmission but also guarantees properties such as non-degeneracy and computational efficiency [**?**]. As a result, bilinear pairing proves instrumental for cryptography and network security applications [50].

4.3. Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) has garnered significant attention in the realm of VANETs due to its potential to bolster the security and efficiency in vehicle-MSW communications [**?**]. This advancement can be attributed to ECC's ability to facilitate quicker and more compact digital signature generation and validation [52]. When synergized with other state-of-the-art technologies such as bilinear peering, Teegraph, and Hashgraph, ECC addresses privacy challenges and assures data integrity in VANETs, thereby rendering the network more secure and trustworthy for smart transportation and road safety [53]. The elliptic curves exhibit exclusive mathematical properties that are leveraged in cryptographic endeavors [54]. The ECC methodology exploits these elliptic curve attributes to generate public and private key pairs in a manner that is more streamlined compared to conventional systems [55]. A subsequent step involves the execution of the scalar multiplication operation on point G using the private number x, resulting in the acquisition of the corresponding public key $P = x \times G$ [56].

Technologies such as ECC, bilinear peering, Teegraph, and Hashgraph play pivotal roles in shaping the trajectory of vehicular ad-hoc networks [57].

4.4. MinBFT

MinBFT, as a software component, stands out due to its capability to realize fault-tolerant Byzantine consensus with fewer nodes and communication rounds compared to conventional BFT protocols [58]. This protocol's development is profoundly influenced by the "Efficient Byzantine Fault-Tolerance" document and judiciously leverages the secure hardware competencies of the nodes involved [59]. Additionally, the project's foundation is Golang, and it employs Intel SGX enclave for the TEE aspect, which is written in C [60].

TEEs, with their foundation in secure hardware, offer data protection and code isolation, shielding against a potentially compromised host system [**?**]. The MinBFT protocol's implementation demands a reduced number of nodes and communication rounds, thus providing tolerance against a more substantial count of faulty nodes [62]. The protocol mandates only 2f+1 consenting nodes to handle f faulty nodes, with the communication process needing just 2 rounds, in contrast to the 3 rounds demanded by PBFT [63].

Upon receiving f+1-consistent COMMIT messages, nodes undertake the request at the local level and update the service's state [**?**]. If the leader is discerned as faulty, a view change procedure is followed to transition the leader node [65]. Significantly, the USIG service is entrusted with generating the tamper-proof segment of consenting nodes [**?**].

4.5. The VANET-Teegraph Algorithm

The VANET-Teegraph algorithm merges vehicular ad hoc networks (VANET) with Teegraph and hashgraph technologies, aiming to enhance security, privacy, and efficiency in vehicular-RSU communication [**?**]. Given the peculiarities of vehicular communication, the algorithm's design has to adapt to the dynamic challenges, including moving vehicles, infrastructural differences, and varying traffic dynamics [68].

The integration of Trusted Execution Environment (TEE) technology offers superior data and code integrity, marking a significant advancement over traditional operating systems and secure elements [69]. Implementing specific security frameworks, such as Intel SGX, provides an added layer of protection via hardware-backed memory encryption [70]. Notably, in the realm of vehicular networks, the VANET-Teegraph algorithm seeks to tackle the unique challenges presented by vehicular communication, offering a robust solution in terms of security, privacy, and efficiency [71].

Importantly, OMTP standards, including those outlining TEE, find their home under the guidance of the GSMA [72]. This standardization effort sees contribution from a spectrum of stakeholders, from service providers to silicon vendors, playing a pivotal role in shaping the future of TEE and related technologies [73].

In order to properly adapt the Teegraph algorithm to the context of ad hoc vehicular networks, several modifications are suggested. These include ensuring vehicle-to-vehicle and vehicle-to-infrastructure communication, implementing mechanisms adaptive to traffic density and vehicle speed, incorporating geo-routing algorithms, providing resilience against VANET-specific attacks, dynamically managing the inclusion and exclusion of nodes, and adjusting the consensus algorithm considering the mobility and variability of vehicular traffic.

The VANET-Teegraph algorithm contemplates several stages such as the registration of vehicles and Road Units (RSU) in a Trusted Authority (TA), generation of temporary and public keys, creation of anonymous signatures, verification of messages by the RSU, identity tracking in case of disputes and communication through Teegraph and Hashgraph technologies.

By combining technological innovations such as Teegraph and Hashgraph, the VANET-Teegraph algorithm aims to effectively address security, privacy, and efficiency challenges in ad hoc vehicular networks. In broad strokes, the VANET-Teegraph algorithm works as follows:

- **Registration**: Vehicles and RSUs must be registered with a Trusted Authority (TA) prior to network deployment. The TA generates pseudonyms and secret keys for each user and stores them in the vehicle's device and its database.
- **Key Generation**: Every time a vehicle enters the range of an RSU, it chooses a random number as a temporary private key and uses it to calculate a short-term public key.
- **Signature Generation**: The vehicle generates a short-term anonymous signature using the short-term anonymous key and sends it to the RSU for verification and maintenance of message integrity.
- **Verification**: The RSU authenticates the vehicle and verifies the message. If everything is correct, it accepts the message and sends an acknowledgment to the vehicle; otherwise, it discards the message and reports it to the TA.
- **Tracking**: In case of disputes between RSUs or vehicles, TA tracks the true identity from its database and reveals it to all RSUs and vehicles, thus denying privacy to malicious users.
- **Communication using Teegraph and Hashgraph**: The system uses Teegraph technology to generate a Directed Acyclic Graph (DAG) locally on each node, allowing local consensus on events in the network. In addition, the algorithm integrates the Hashgraph data structure to improve the efficiency and security of communication and storage of information on the network.

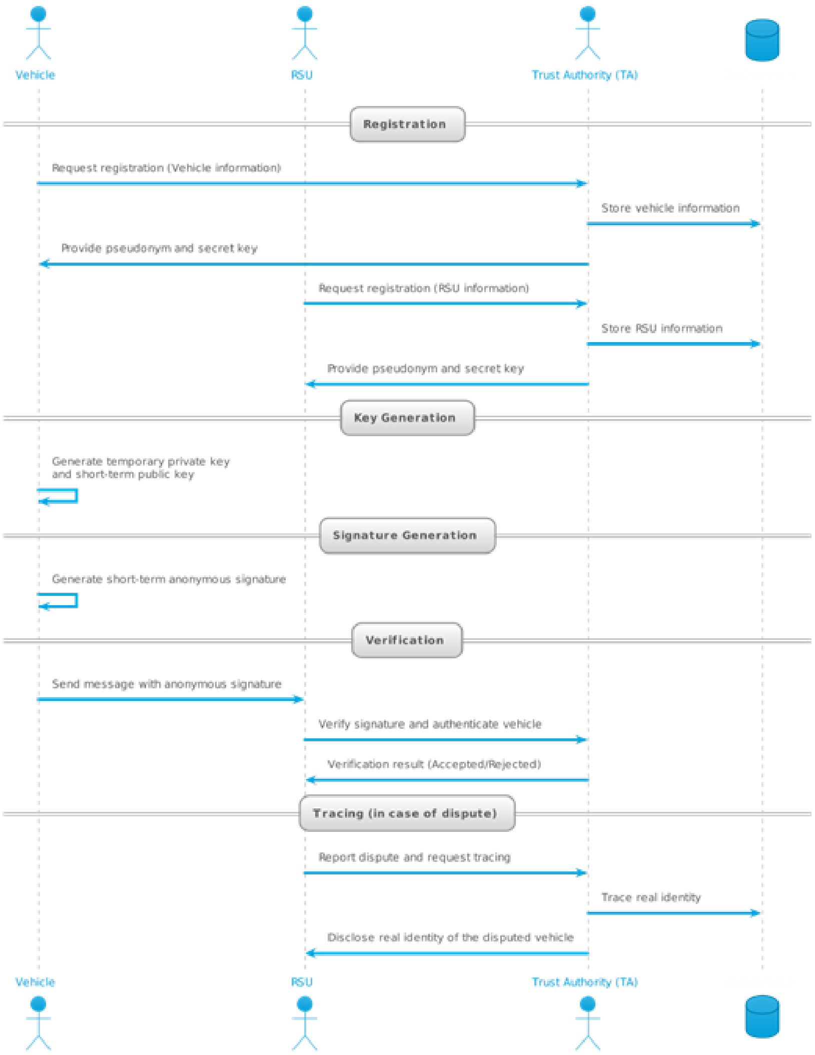

Vehicle
RSU
Trust Authority (TA)
Registration
Request registration (Vehicle information)
Store vehicle information
Provide pseudonym and secret key
Request registration (RSU information)
Store RSU information
Provide pseudonym and secret key
Key Generation
Generate temporary private key
and short-term public key
Signature Generation
Generate short-term anonymous signature
Verification
Send message with anonymous signature
Verify signature and authenticate vehicle
Verification result (Accepted/Rejected)
Tracing (in case of dispute)
Report dispute and request tracing
Trace real identity
Disclose real identity of the disputed vehicle
Vehicle
RSU
Trust Authority (TA)

Figure 4. Illustrates the interactions between the vehicle, the RSU, and the trusted authority (TA), as well as the use of the TA database.

The VANET-Teegraph algorithm aims to address security, privacy, and efficiency challenges in VANET by combining innovative technologies such as Teegraph and Hashgraph. (See Figure 4).

4.6. Design by VANET-Teegraph

The VANET-Teegraph algorithm is specifically designed for use in ad hoc vehicular networks, encompassing a series of steps and important considerations. Initially, a requirements analysis is conducted to pinpoint the distinct necessities of the VANET setting. This includes aspects like vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, adaptability to fluctuating traffic conditions, resilience against potential attacks, and ensuring efficient data transmission. Subsequent stages involve the crafting of registration and authentication mechanisms, where a Trusted Authority plays a pivotal role by generating pseudonyms and secret keys for both vehicles and RSUs. Furthermore, a system is integrated wherein each vehicle produces temporary private keys and their corresponding public keys as soon as they come within the purview of an RSU. An integral component of the design also includes a mechanism for tracking and dispute resolution. In this framework, the Trusted Authority (AT) possesses the capability to pinpoint and unmask the identities of malicious users in the event of disagreements between vehicles and RSUs. Finally, to ensure seamless operation within the VANET milieu, the Teegraph consensus algorithm is adapted. This takes into account the inherent challenges of mobility and the variable nature of vehicular traffic.

4.6.1. How the Teegraph is generated

The VANET-Teegraph algorithm employs an approach similar to the Hashgraph communication model, derived from Amazon's renowned Dynamo. Teegraph consists of events and columns. Each column represents a node, and each event within the columns symbolizes a token event. In the Teegraph, the flow of time is represented upwards, and vertices are used to represent events in history.

The main distinction is that in a Teegraph event, two hashes are connected to their two parent events, whereas in a blockchain there is only one parent for each block. In the Teegraph graph, each vertex except the first in each column has two down-going edges that connect to the immediately preceding events, called self-parent and other-parent. For example, in Figure 5, the proper parent of event 1 is event 2 and the other parent is event 3.

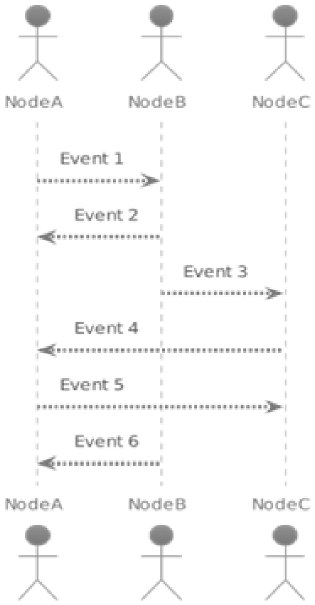


Figure 5. Generation of the TEEgraph

When node B passes a token to node A for the first time, it sends all the events it contains, including events 3, 4, and 6, to A. Time flows up the graph, so the lower vertices represent events earlier in history. The number of nodes in the VANET-Teegraph exactly matches the number of columns in the graph.

Each node builds the graph locally, adds events to the columns, and sets borders based on token history. When a node receives events from another node, it incorporates these events into its own graph. Then create a new event and select a random neighbor to send events to.

The graph generation process (Figure 5) for VANET-Teegraph can be described in the following steps:

- **Initial State:** Each node has a single event in its corresponding column, while the other columns are empty.
- **Step 1 (Tokens from Node A to Node B):** Node A randomly selects a neighbor (Node B) to send events to. Node A sends all the events it has and does not have to node B. Node B creates a new event and sets its parents accordingly.
- **Step 2 (Tokens from Node B to Node C):** Node B randomly selects a neighbor (Node C) to send events to. The process is similar to Step 1, with node B sending events to node C and node C creating a new event.
- **Step 3 (Tokens from node C to A):** node C randomly selects a neighbor (node A) to send events to. Node C sends events to node A, and node A creates a new event with the parents set appropriately.

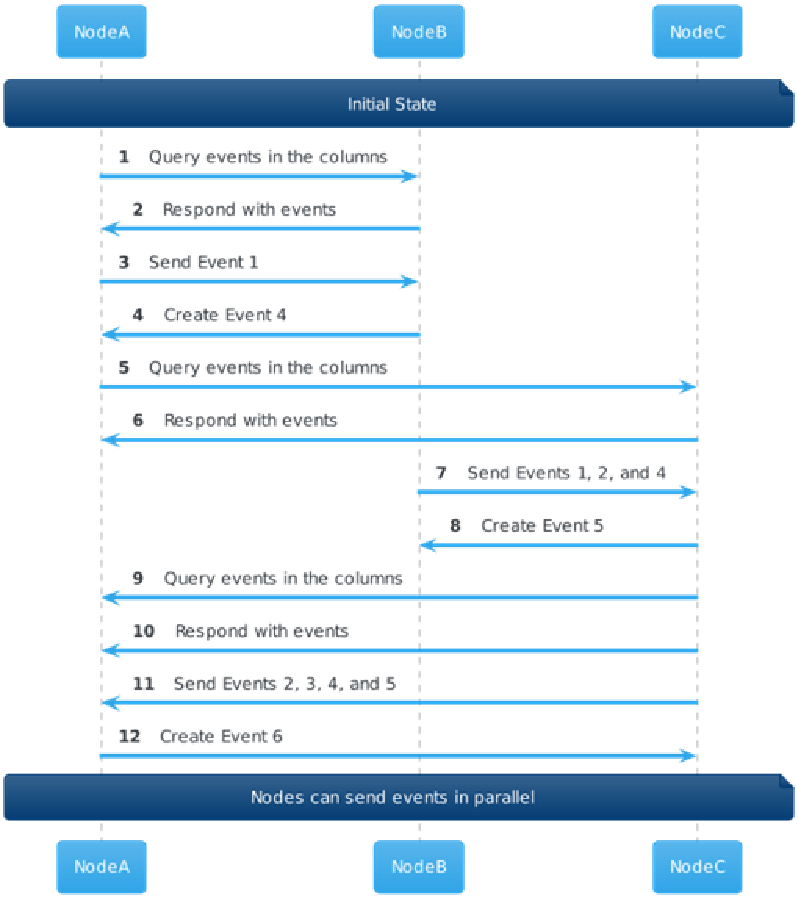

NodeA
NodeB
NodeC
Initial State
1 Query events in the columns
2 Respond with events
3 Send Event 1
4 Create Event 4
5 Query events in the columns
6 Respond with events
7 Send Events 1, 2, and 4
8 Create Event 5
9 Query events in the columns
10 Respond with events
11 Send Events 2, 3, 4, and 5
12 Create Event 6
Nodes can send events in parallel
NodeA
NodeB
NodeC

Figure 6. Event exchange between nodes A, B, and C, along with the creation of new events during the graph generation process

It's important to note that nodes can send events in parallel. While node A is sending tokens to node B, other nodes can also be sending tokens to their selected neighbors. This process allows for efficient generation of local graphs in each node and facilitates local consensus on events within the VANET network.

The VANET-Teegraph algorithm uses a Directed Acyclic Graph (DAG) approach to represent the history of token events in the VANET network. This approach enables efficient and secure communication in the network and facilitates local consensus on events within the VANET network.

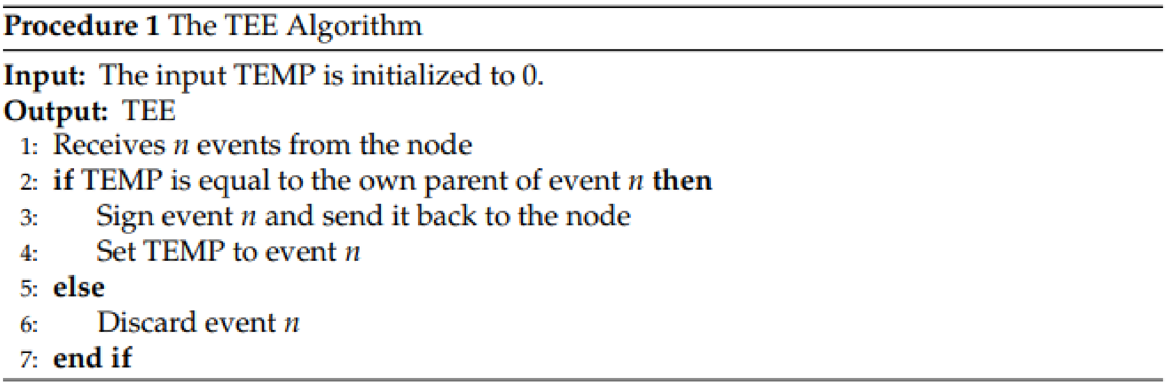
4.6.2. The single use mechanism

To prevent a fork attack in our VANET-Teegraph algorithm, a malicious node can't create two different events and assign them the same parent. In most IoT scenarios, device initialization is controllable, which allows us to standardize these devices with the same hardware or adapt them according to their functions. In VANET-Teegraph, we leverage the Trusted Execution Environment (TEE) to avoid fork attacks. We design a mechanism called "single-use of self-parent," in which the TEEs ensure that each event can be self-parent only once. Before sending an event to the network, it must obtain a TEE's signature, proving that its own parent is set as the parent only once.

To obtain the trust signature from the TEE, four steps are followed:

1. The node sends event n to the TEE.
2. The TEE compares the hash of event n's own parent with the hash of event $n - 1$ stored in its memory.
3. If they are equal, the TEE signs event n and returns it to the node.
4. The TEE stores the hash of event n, replacing the hash of event $n - 1$ in its memory.
5. If in step 3 they are not equal, the TEE discards event n and halts the process.

In summary, if event $n - 1$ in a TEE's memory is set as the own parent of event n, event $n - 1$ will be replaced by event n in the TEE's memory immediately. A node can never create two different events with the same own parent, which means that the fork attack can never occur.



Procedure 1 The TEE Algorithm

```
Input: The input TEMP is initialized to 0.
Output: TEE
Receives $n$ events from the node
if TEMP is equal to the own parent of event $n$ then
    Sign event $n$ and send it back to the node
    Set TEMP to event $n$
else
    Discard event $n$
end if
```

Upon receiving an event, the event validation must add a new element, which is the validation of the TEE's signature. As depicted in Figure 6, the first time Node C receives synchronization from Node A, it receives events 1, 2, 3, and 4. After Node C verifies all the received events, it creates a new event 5 and sets its other parent as Event 1, indicating that Node C votes "YES" for events 1, 2, 3, and 4. Consequently, if Node A creates an event and sets its other parent as Event n from another node, it implies that Node A voted "YES" for Event n and all the ancestors of Event n.

In Figure 7, all the dark events have reached a consensus as they have received over half of the votes from the nodes. Like in the well-known RAFT [76] and Paxos [77] algorithms for crash fault tolerance (CFT), nodes cannot send equivocal (voting) messages to others. Therefore, half of the node votes are enough for an event to reach a consensus. It is demonstrated in [78,79] that the complexity of the Byzantine fault tolerance (BFT) problem can be reduced to the CFT problems if a malicious server cannot lie differently to different clients or servers. Furthermore, in this case, the lower bound of the total number of participants required to tolerate f failures can be reduced from $3f + 1$ to $2f + 1$. In VANET-Teegraph, with the aid of TEE, a malicious node can never lie to other honest nodes. Therefore, the complicated consensus process in Hashgraph is unnecessary, and $2f + 1$ nodes are enough to tolerate f malicious nodes.

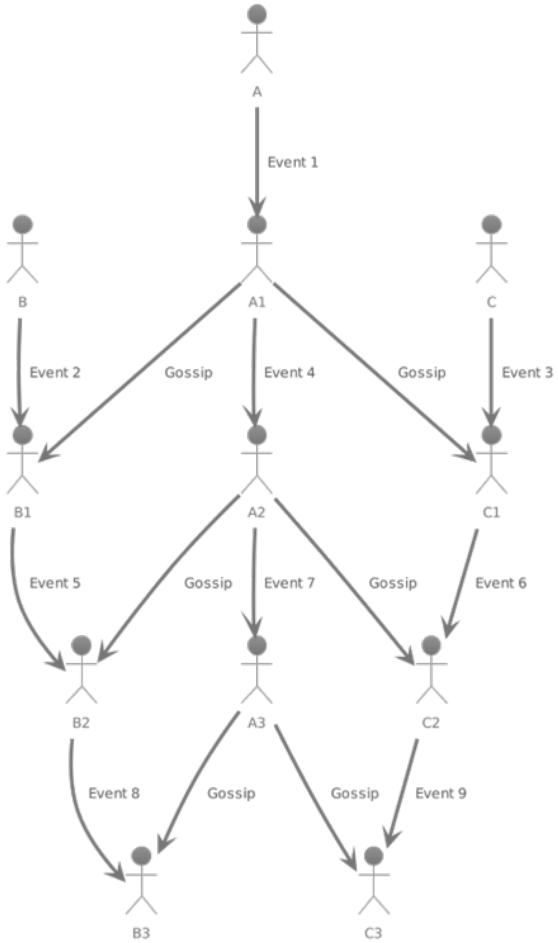


Figure 7. A graph with three nodes A, B, and C. Events are generated and shared among nodes through the "Gossip" process.

Different types of devices can form a network in some IoT scenarios and work together on a task without a trusted intermediary. Sharing data among these untrusted devices for collaboration is required. Moreover, the shared data must reach a consensus among these devices before the network utilizes it. Most consensus algorithms can achieve this when the consensus subjects are always the same (nodes cannot be replaced, new nodes do not join, and nodes do not exit).

The consensus algorithm in VANET-Teegraph is designed to adapt to VANET environments, where nodes can change frequently, and the network topology can be dynamic. Below are some key aspects of consensus in VANET-Teegraph:

Node Admission and Exit: In VANET environments, nodes can join and leave the network frequently. Therefore, the consensus algorithm must be capable of accommodating the addition and removal of nodes efficiently. In VANET-Teegraph, the consensus process can adapt to these dynamic changes as nodes can continue sending and receiving events from other nodes in the network even if some nodes join or exit the network.

Fault Tolerance and Malicious Nodes: With the help of TEE, VANET-Teegraph can tolerate malicious nodes and ensure the integrity and security of shared data in the network. Malicious nodes cannot send equivocal messages to other nodes, which means that a fork attack can never occur. Furthermore, $2f + 1$ nodes are enough to tolerate f malicious nodes in VANET-Teegraph.

Efficiency and Scalability: The consensus algorithm in VANET-Teegraph is efficient and scalable as it allows nodes to share events and reach consensus quickly, even in dynamic VANET environments. Nodes can send and receive events in parallel, which accelerates the consensus process and enables the network to adapt to a larger number of nodes and events.

The consensus algorithm in VANET-Teegraph is designed to adapt to VANET environments and address the challenges of fault tolerance, data integrity, efficiency, and scalability in the network.

Begin the algorithm with two simultaneous loops running on separate threads: The first loop deals with the transmission of events. In this loop, the program sends all known events, including the most recent one, to a random node. This loop continues indefinitely. Simultaneously, the second loop handles event reception at node N. Here, the program receives events from neighboring nodes. Once the events are received, the program checks their validity. If all events are valid, the program proceeds to create a new event labeled 'm'. The 'other parent' of the new event 'm' is set to the last event from node N. Subsequently, the 'other parent' of event 'm' is established as the most recently created node. The new event 'm' is then set within the Trusted Execution Environment (TEE) in order to secure a signature. However, if any of the received events are found to be invalid, the program discards all received events. This loop also continues indefinitely. At the end of this loop, the program searches for new events that have reached consensus. The algorithm concludes once both loops have been executed.

Procedure 2 TEE Consensus Algorithm via Dynamic Change

```
Require: Two loops are executed in parallel on two threads, sending
Ensure: TEE
while loop do
    Send all known events, including the newly arrived event, to a random node
end while
while loop do
    Receive events from neighbors at node $N$
    if All events are valid then
        Create a new event in $m$
        Set the other parent of event $m$ as the last event from node $N$
        Set the other parent of event $m$ as the last created node
        Set a new event $m$ in TEE to obtain the signature
    else
        Discard all received events
    end if
    Search for new events that have reached consensus in the final loop
end while
```

The combined effect of these loops, facilitated by the TEE, allows for robust data sharing and collaboration between devices in dynamic Internet of Things (IoT) environments. With the help of the consensus algorithm and TEE, VANET-Teegraph can provide a safe and reliable solution for data sharing and collaboration among devices in dynamic and changing IoT scenarios. With the aid of the consensus algorithm and TEE, VANET-Teegraph can provide a secure and dependable solution for data sharing and collaboration among devices in dynamic and ever-changing IoT scenarios. However, it's often the case that there are multiple subtasks, requiring the swarm to divide into various distinct subgroups. For instance, in earthquake rescue operations, a swarm comprising of drones and autonomous vehicles would need to separate for different subtasks: the drones survey the general surroundings while the autonomous vehicles rescue discovered victims. Upon completing their subtasks, these subgroups can reunite to address a new task, as depicted in Figure 8. Therefore, the consensus algorithm for IoT blockchains must be capable of reaching a consensus even as consensus subjects continually change, without the support of a trusted intermediary.

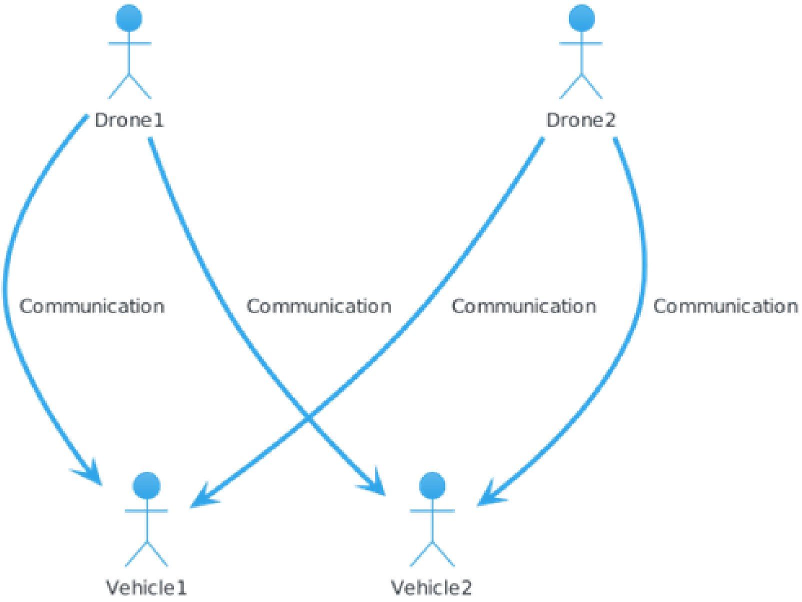


Figure 8. Consensus

When the swarm starts to perform a task, all devices are connected and can communicate freely with each other. In this manner, a Teegraph is constructed. When some devices separate into subgroups to perform subtasks, devices within the same subgroup communicate internally. As shown in Figure 9, subgroups A and B separate from all devices to carry out their respective subtasks. Events from subgroup A (B) reach consensus once they obtain half of the "YES" votes from the devices in subgroup A (B). After reuniting the swarm, free communication resumes throughout the network, and events must obtain half of the "YES" votes from all swarm devices to reach a consensus.

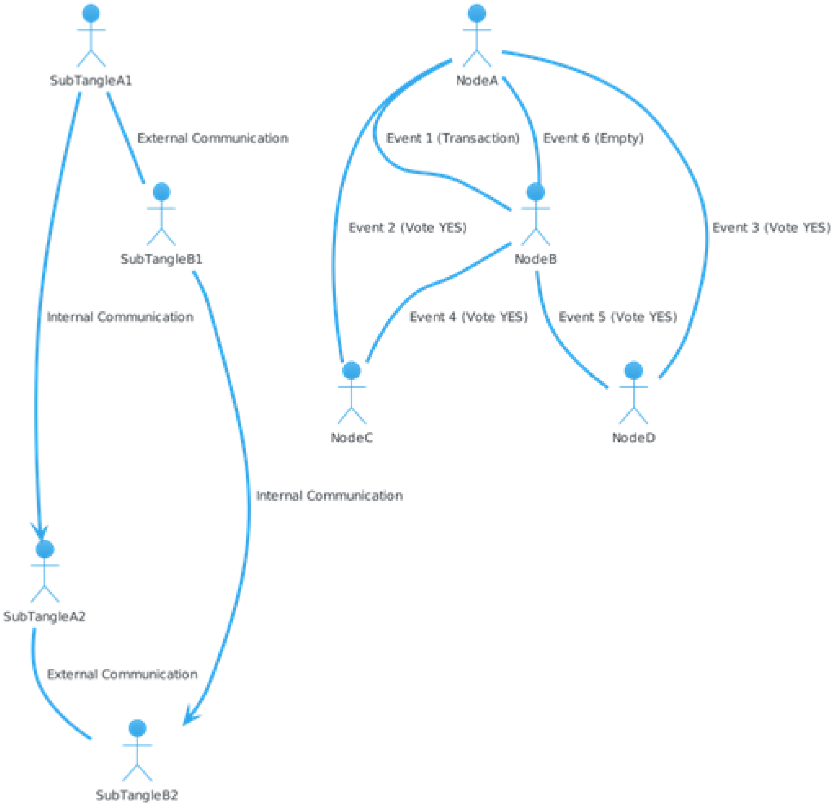


Figure 9. Resource-saving mechanism

In the original Hashgraph, each node needs to constantly create and send events to validate other nodes' events. Even if a node has no transactions to send, it must create and send empty events, which contain no transactions. However, when no new transaction is created, and all existing transactions have reached a consensus, these empty events no longer contribute to the system but waste network and storage resources.

To address this issue, we propose a resource-saving mechanism for Teegraph. Following this mechanism, nodes can stop transmitting gossip at the right time: before a node creates and sends an empty event, it can judge whether it should do so according to the Teegraph structure it possesses. As shown in Figure 10, the light circles represent empty events, while the dark ones represent events containing transactions. Event 1 is the last event that contains transactions, and events 2 to 5 have obtained enough votes to confirm event 1. When node A creates event 6, it can ensure that all system nodes have confirmed event 1, as event 6 is descendant from events 2 to 5, all of which have obtained enough votes for event 1. Furthermore, there are no unconfirmed transactions in the system. To reduce network communication costs and save storage resources, node A can stop transmitting gossip after sending event 6. When another node receives event 6, it can also ensure that all other nodes have confirmed event 1. Then, it will stop "transmitting gossip". Any node will restart gossip-based communication when creating an event containing new transactions.

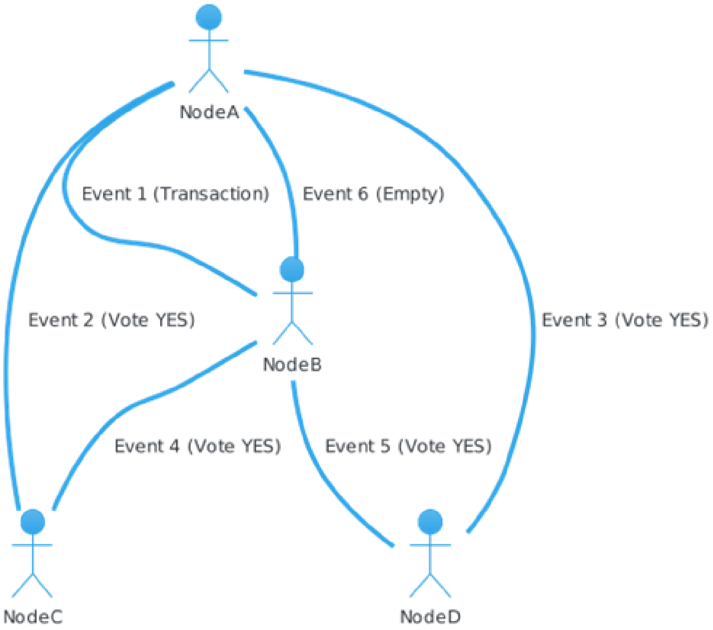


Figure 10. Teegraph Correction

We examine Teegraph's correctness, focusing on its safety and liveness attributes inherent to Byzantine Fault Tolerant consensus algorithms. Notably, Blockchain consensus mechanisms often face a trade-off: increased decentralization and security may lead to diminished performance, as observed in PoW, whereas improved performance and security might result in reduced decentralization, as in PBFT. The dilemma of achieving security, decentralization, and performance is often termed as the "Blockchain Impossible Triangle". Key points regarding Teegraph include:

- **Security:** In Teegraph, nodes cannot produce forks. Even when two conflicting events arise, they maintain a parent-child relationship, ensuring only the parent event can achieve consensus, given it has over half of the affirmative votes. Assumptions include: less than 50% of nodes act maliciously (Teegraph can bear over 1/3 Byzantine nodes due to TEE's preventive measures), and the security of encryption methods and hash functions.
- **Liveness:** Teegraph ensures any valid event from an honest node will achieve consensus. An event requires votes from over half of the nodes to gain consensus. Leveraging the gossip protocol,

events spread quickly, generally taking $O(\log N)$ rounds to reach all nodes, with N being the total nodes.

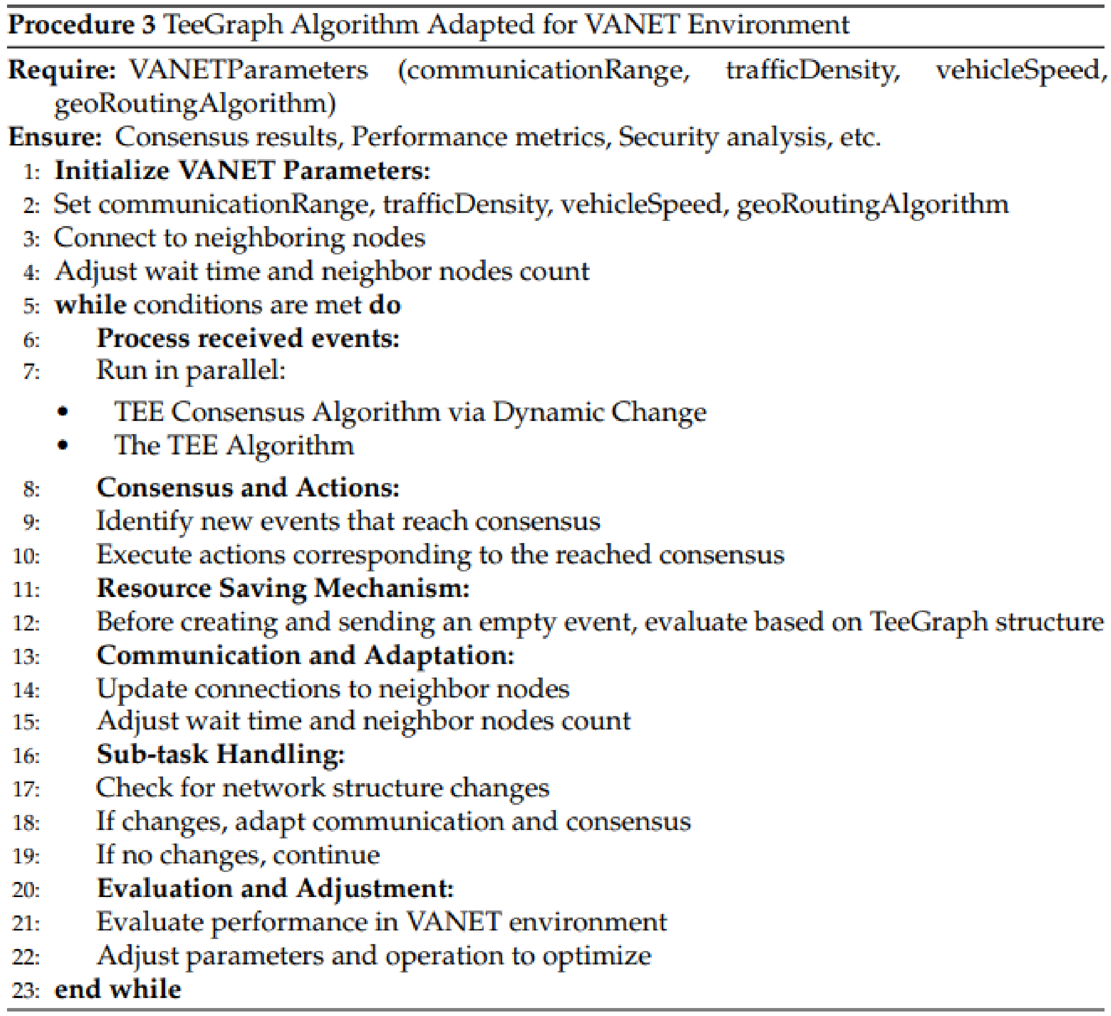
- **Effectiveness:** Teegraph's performance covers aspects like throughput, latency, scalability, and resource usage. Scalability is notably strong, thanks to the gossip protocol, allowing adaptability in extensive distributed networks.
- **Decentralization:** Every node in Teegraph has equal transactional responsibility, promoting fair distribution of accounting rights. However, due to reliance on TEEs, there's a certain centralization, trading off for enhanced performance and security.

Teegraph, integrating consensus algorithms and TEEs, provides a dependable solution for IoT data interchange. It excels in security and performance but sacrifices some decentralization. The system reveals the inherent compromises in blockchain architecture, emphasizing the need to adjust based on requirements. Future updates might refine this balance, catering to diverse IoT scenarios.

4.6.3. Integration for VANET environment (Vehicular Ad-Hoc Network)

Teegraph addresses the blockchain's impossible triangle paradox, striking a balance between efficiency, security, and decentralization. By integrating gossip protocols and TEE, Teegraph accomplishes heightened performance and security while upholding a satisfactory level of decentralization. This design is conducive for VANET scenarios and IoT platforms that necessitate secure, high-performance device-to-device communication.

The TeeGraph algorithm, fine-tuned for VANET environments, addresses challenges inherent to vehicular networks, considering the volatility and variance of vehicular traffic. This ensures enhanced efficiency, security, and scalability in genuine vehicular communication settings.



Procedure 3 TeeGraph Algorithm Adapted for VANET Environment

```
Require: VANETParameters (communicationRange, trafficDensity, vehicleSpeed,
    geoRoutingAlgorithm)
Ensure: Consensus results, Performance metrics, Security analysis, etc.
Initialize VANET Parameters:
Set communicationRange, trafficDensity, vehicleSpeed, geoRoutingAlgorithm
Connect to neighboring nodes
Adjust wait time and neighbor nodes count
while conditions are met do
    Process received events:
    Run in parallel:
      • TEE Consensus Algorithm via Dynamic Change
      • The TEE Algorithm
    Consensus and Actions:
    Identify new events that reach consensus
    Execute actions corresponding to the reached consensus
    Resource Saving Mechanism:
    Before creating and sending an empty event, evaluate based on TeeGraph structure
    Communication and Adaptation:
    Update connections to neighbor nodes
    Adjust wait time and neighbor nodes count
    Sub-task Handling:
    Check for network structure changes
    If changes, adapt communication and consensus
    If no changes, continue
    Evaluation and Adjustment:
    Evaluate performance in VANET environment
    Adjust parameters and operation to optimize
end while
```

5. Describe the proposed scenario

Connected vehicles will play a crucial role in various scenarios beyond traditional autonomous driving. Their on-board units and data processing capabilities will interact with different entities through Vehicle-to-Everything connections, both in close physical proximity and in cyberspace. Privacy and integrity are major concerns in this context. Our study introduces an approach rooted in the TeeGraph algorithm, tailored for the Vehicular Ad-hoc Network (VANET) environment. This method gathers and processes road traffic data as an open-source intelligent transportation system. Vehicle privacy is safeguarded by changing directions with each interaction with road beacons and leveraging the security attributes of the TeeGraph algorithm.

The TeeGraph algorithm, when adapted for VANET, enhances inter-vehicle communication, contributing to better traffic management. Bypassing the need for an intricate blockchain structure, the efficiency of the vehicular network message exchange is notably enhanced, especially in extensive networks.

This adapted algorithm also offers more versatility in handling mobility and variations in vehicular traffic. By dynamically modifying the interval between events and adjusting the number of neighboring nodes receiving the events, the algorithm can accommodate fluctuations in traffic density and vehicle velocities.

Moreover, this algorithm exhibits resilience against VANET-specific threats. With security protocols to thwart phishing attacks, the vehicular network becomes increasingly impervious to potential risks.

Dynamic node incorporation and removal are also streamlined in this method, detailing a procedure for node inclusion and exclusion as vehicles transition in and out of the network's range. To document vehicle message transfers, TeeGraph's Proof of Work technology was contemplated, but its steep computational demands at each Road Side Unit (RSU) pose challenges. Hence, our proposed model employs Hashgraph proof-of-authority technology to curtail computational overhead. Storage demands are also minimized by retaining only vital vehicle messages in TeeGraph. Vehicles are ranked based on message type and then authenticated to RSU.

In dense networks, rising vehicle counts amplify the load on RSUs, slowing the system. Our proposal eases the RSU load by reducing authentication instances and adopting a Gossip Protocol consensus algorithm to expedite processes.

To summarize (as depicted in Figure 11), our system addresses the rapid vehicle authentication challenge in VANET by minimizing necessary authentications and streamlining vehicular communication intervals.

This is achieved via message prioritization, authenticity checks, integrity validation, and tracking malevolent vehicles using the Hashgraph proof-of-authority technology combined with the Gossip Protocol consensus algorithm. The end result is a swifter, more secure system that aids in averting traffic mishaps and congestions, thereby enhancing vehicular communication efficiency and safety.

The proposed Vanet-TreeGraph-based communication scheme for vehicles employs the hashgraph data structure for consensus. The primary administrator of Vanet-TreeGraph is the certificate authority, overseeing multiple RSUs to ensure expansive network coverage. Autonomous control by service management cells is essential for vehicle contact messages, as the intention is extensive regional deployment. Service managers are crucial for message relay, with a singular service manager supervising several RSUs. For the advanced encryption counter mode with block chaining message authentication via encrypted code, the group key remains concealed. The message's ciphertext payload spans 32-bits, promoting transmission efficiency.

The efficiency of key transmission is gauged by the block's propagation duration from the current operations manager to the target service manager, as detailed in the associated tables. Moreover, the proposed model harnesses the hashgraph data structure for consensus, resulting in superior throughput and expedited transaction confirmations compared to conventional blockchain algorithms.

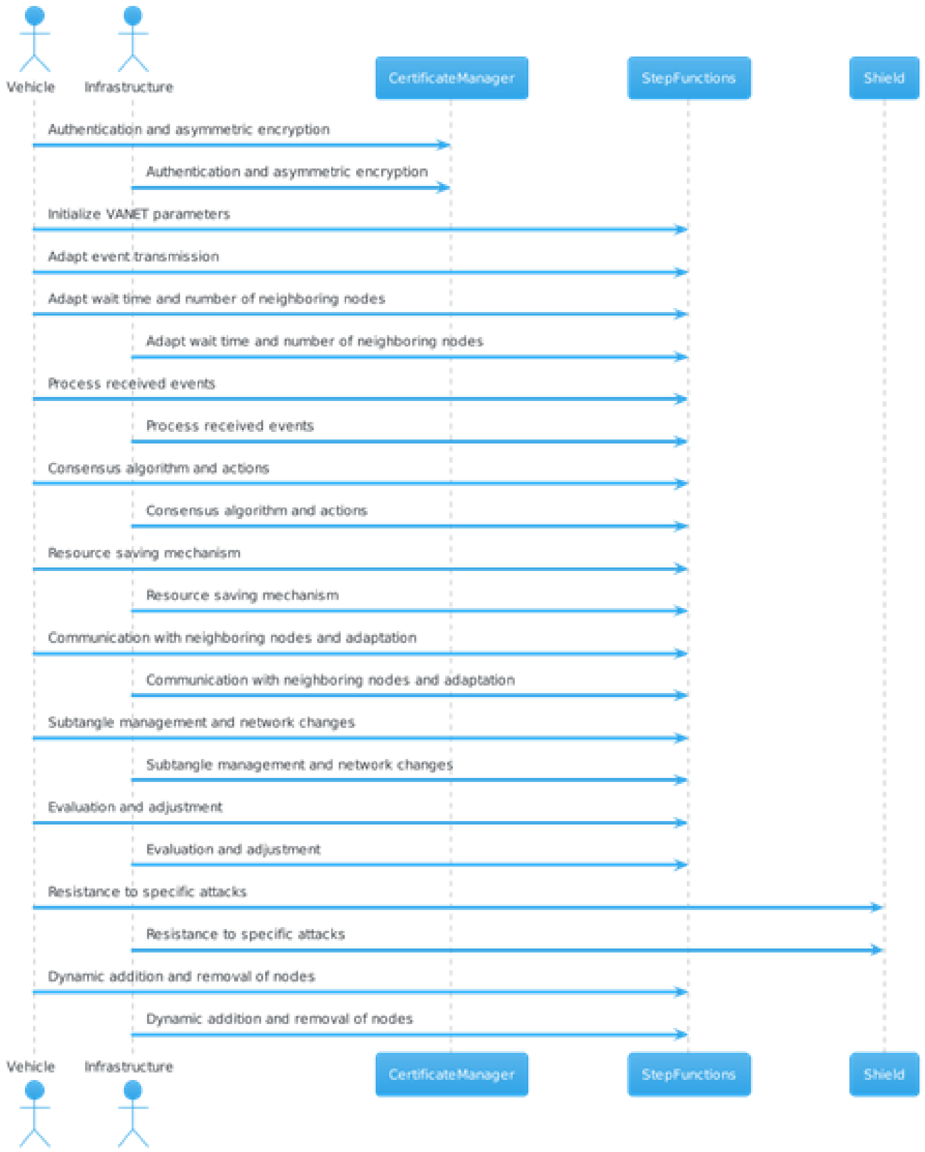

Vehicle
Infrastructure
CertificateManager
StepFunctions
Shield
Authentication and asymmetric encryption
Authentication and asymmetric encryption
Initialize VANET parameters
Adapt event transmission
Adapt wait time and number of neighboring nodes
Adapt wait time and number of neighboring nodes
Process received events
Process received events
Consensus algorithm and actions
Consensus algorithm and actions
Resource saving mechanism
Resource saving mechanism
Communication with neighboring nodes and adaptation
Communication with neighboring nodes and adaptation
Subtangle management and network changes
Subtangle management and network changes
Evaluation and adjustment
Evaluation and adjustment
Resistance to specific attacks
Resistance to specific attacks
Dynamic addition and removal of nodes
Dynamic addition and removal of nodes
Vehicle
Infrastructure
CertificateManager
StepFunctions
Shield

Figure 11. The proposed system's approach to swiftly authenticate vehicles in VANET.

6. Presents the results for the proposed scenario

In the following section, we delve into the heart of our research by presenting the empirical results derived from our proposed scenario. These findings, based on rigorous experimentation and analysis, offer insights into the efficacy and implications of our approach. Detailed observations, comparisons, and interpretations will be provided, allowing readers to grasp the significance and impact of the scenario under investigation. Let us now turn our attention to these pivotal results.

6.1. Simulation: Calculation of the Probability of Dropout (POL)

The main purpose of using the VANET-TeeGraph algorithm is to avoid traffic congestion that could lead to traffic accidents. In traditional methods, such as wireless sensor networks and long-term evolution (LTE) networks, node entry and exit are unpredictable. Therefore, specific organizational schemes allow nodes to subscribe to multiple key change intervals to measure egress probability. Unfortunately, security flaws arise when the program allows users to choose their subscription period: a malicious user intercepts critical messages by requesting an active period longer than the actual duration of their residence.

Probability-based models are much more effective than conventional methods and are easier for vehicular nodes to implement due to their predictable movement paths. The departure probability helps determine the movement of the vehicle. However, during vehicular communication, most vehicles lack the ability to exit the communication community to the next edge of the lot because it is difficult for them to catch up to exit the boundary zone after the key change time. current.

To address this problem, a departure rate [1,4] is introduced, given by $Y = \min(1/T_1, T_2)$, where T_2 is the time cost for vehicles to exit or exit, and T_1 is the batch of intervals. The idea is to use TeeGraph to represent and share information related to the probability of output between the nodes.

Our focus is on improving communication between vehicles, service managers and the Certification Authority (CA). Teegraph can help optimize how cryptographic keys are shared and updated, and how decisions are made based on local consensus.

Applying Teegraph to the proposed scenario, we can address the following aspects:

- Communication between vehicles and Road Units (RSUs): Teegraph can help optimize the way vehicles and RSUs share information with each other. By employing the Directed Acyclic Graph (DAG) structure and the "gossip about gossip" approach, efficiency in communication and decision making within the network can be improved.
- Rekeying and Key Updates: Teegraph can assist in the rekeying process in the Logical Key Hierarchy (LKH) by facilitating efficient key updating and distribution across the network. By using the local consensus approach, nodes can make faster decisions on whether or not to accept an updated key.
- Reduced key transfer time: Under high traffic conditions, Teegraph can help reduce the time it takes to transfer keys between nodes on the network. This is achieved through more efficient communication and consensus-based decision making.

In order to evaluate the impact of Teegraph in the proposed scenario, it is necessary to carry out simulations and tests in an autonomous vehicle environment. These tests can compare the performance and efficiency of the Teegraph-based solution with the original Hashgraph-based approach.

Once Teegraph is applied to the scenario, a reduction in key transfer time and greater efficiency in communication between vehicles and service managers is expected. This should result in better quality of service and greater security in communications between nodes on the network.

We can conclude that the adapted VANET-TreeGraph algorithm outperforms the Hashgraph algorithm in all tested traffic scenarios. VANET-TreeGraph consistently shows superior transmission time, efficiency, lower communication latency, higher message delivery success rate, more messages delivered, fewer messages lost, less resource consumption, lower collision rates and retransmission times shorter. These findings demonstrate the potential of VANET-TreeGraph to provide a more efficient and reliable solution for vehicular communication networks, particularly in environments experiencing various levels of traffic.

6.2. Experiments for TEE

In this subsection, we highlight the efficiency of the Trusted Execution Environment (TEE) within TreeGraph. We've established a tamper-proof counter service in MinBFT, a pluggable component that streamlines Byzantine Fault Tolerance (BFT) consensus with fewer nodes and reduced communication rounds. It's built on the principles from the paper "Efficient Byzantine Fault-Tolerance"[81], emphasizing secure hardware capabilities.

Our lab executes the MinBFT consensus protocol, primarily coded in Golang. However, the TEE segment is crafted in C, aligning with Intel® SGX enclave specifications. We hope that, by integrating this component into existing blockchain frameworks, the community can apply it across diverse practical scenarios.

6.2.1. Efficiency of Message Signing

For our first experiment, we assessed the time needed for message signing in both Teegraph and MinBFT. We funneled 500,000 messages through each system, gauging the signing duration. In the realm of distributed systems, rapid message signing is crucial for swift consensus and system efficiency

[82]. This high-throughput scenario mirrors real-world blockchain applications like financial services or supply chains [83]. We tracked results by elapsed time to discern the efficiency of each system under intense loads.

6.2.2. Efficiency of Message Verification

Our subsequent experiment delved into message verification within Teegraph. For consistency, we used messages from our prior test. This verification not only evaluates Teegraph's efficiency but also underscores the value of message validation in maintaining transaction integrity [**?**].

Message verification, essentially the counterpart to signing, is pivotal in preserving system trust, especially in public blockchains where trust among participants might be absent [85]. This verification step needs to be both swift and secure, given its influence on system throughput.

6.2.3. Results of previous experiments

To evaluate the efficiency of signature and verification operations of both algorithms, we utilized the Python crypto library. Concurrently, the 'ThreadPoolExecutor' was employed to manage nodes, each represented by a thread, simulating the reception and processing of messages. Upon completion, threads notify the console and the Executor collects the results. The use of 'ThreadPoolExecutor' from the 'concurrent.futures' library allows for parallel processing, enhancing the performance.

Our focus was on comparing the efficiency of message signing and verification between Teegraph and MinBFT. Efficiency was measured in seconds (denoted as 'Time_s') over several runs, and the results were visualized using Seaborn bar and box plots. Consistently, Teegraph exhibited superior performance.

The tamper-proof reliable counter service provided by the Trusted Execution Environment (TEE) in MinBFT ensures each node can verify the system's state independently [86]. In contrast, Teegraph solely verifies the signature. MinBFT's additional verification of the counter value possibly results in its observed reduced efficiency.

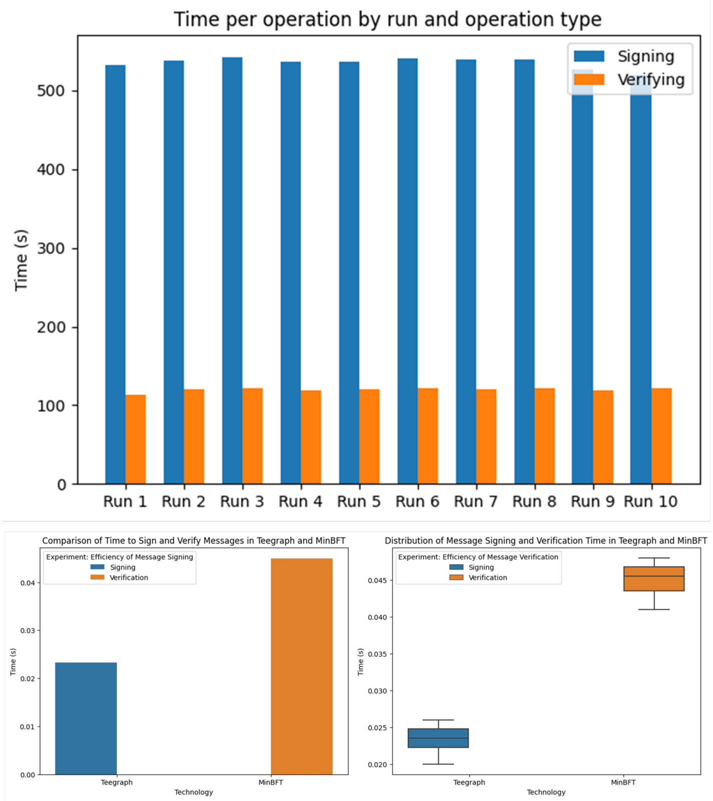


Figure 12. Comparison of Message Signing and Verification Time in Teegraph and MinBFT

Both experiments were iterated ten times to capture a broad data spectrum and account for any variances. This repetitiveness aimed to offer a holistic perspective of system behaviors under varying conditions.

Insights gathered from these tests illuminate the performance differences of the two systems. Potential future research could explore varying load conditions, message sizes, or the influence of network latency.

6.3. Visualization: Experiments for performance and latency assessment

In this section, we model each node as a thread using Python. The experiments are conducted within a Python environment where nodes (represented as threads) operate concurrently, sending and receiving transactions, as well as validating and confirming events. We employ r to signify the event request interval (that is, nodes generate events periodically with a time interval of r), while p denotes the event propagation time. Consequently, the network latency is expressed as $r + p$. We aim to contrast the performance (measured as eps, which indicates the number of events processed per second) and latency (ltc, the mean duration for an event to attain consensus from its inception to its validation) of Teegraph against that of Hashgraph (the consortium version). Simulations are orchestrated under a myriad of conditions, encompassing varying node counts and distinct network latencies, which in turn includes diverse event request intervals and event propagation durations.

6.3.1. Scenario 1: Comparison of throughput and latency between Teegraph and Hashgraph with different numbers of nodes

In our inaugural simulation, we delve into the repercussions of the cumulative count of nodes, which oscillates between 4 to 50. Initially, we equate $r + p$ to 200 ms. In a subsequent, more intricate setting, $r + p$ is adjusted to a stochastic value ranging from 200 ms to 500 ms. The outcomes are illustrated in Figure 13. The observations denote an escalation in latency as the node count surges. A surge in nodes indicates a proliferation of events, but it also signifies the necessity for additional votes for an event to achieve consensus. Thus, performance reaches its zenith with approximately 30 nodes for Hashgraph (the acme for Teegraph is identified in the ensuing simulation). In a comparably stable network devoid of faulty nodes, some outliers mar Hashgraph's latency. This anomaly stems from Hashgraph's requisite for a trifecta or more rounds of majority vote accumulation, in contrast to Teegraph's singular round requirement. Teegraph demands substantially fewer communication stages than Hashgraph. Hence, in this trial, Teegraph trumps Hashgraph in both throughput and delay metrics. Our analyses reveal a plummet in performance and a spike in latency concurrent with an augmentation in the quantity of faulty nodes. Teegraph remains relatively unscathed amidst an increase in faulty nodes and bests Hashgraph. Moreover, once the count of faulty nodes transcends 16, Hashgraph grinds to a halt, whilst Teegraph perseveres even with an excess of 24 malfunctioning nodes.

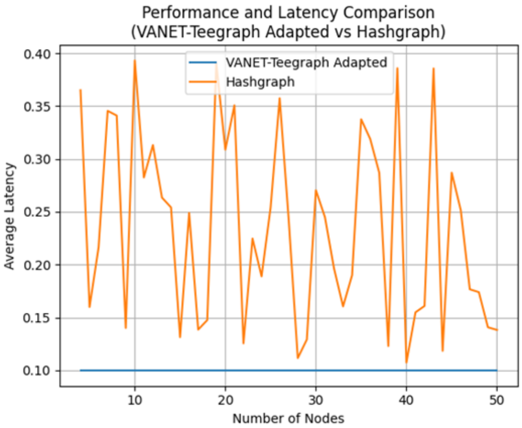


Figure 13. Distribution of Message Signing and Verification Time in Teegraph and MinBFT

6.3.2. Scenario 2: Scalability Comparison Between Adapted VANET-Teegraph and Hashgraph Based on Throughput and Latency

In the ensuing scenario, every block and transaction undergo validation by juxtaposing them against the network's consensus rules. Complete nodes are also mandated to retain a blockchain replica, implying the obligation to download all the transactions and blocks that have unfolded on the blockchain. Notwithstanding the ubiquity of extensive IoT devices, the lion's share remains light nodes, devoid of the requisite to download the entirety of blocks and transactions. Fetching every block and transaction becomes superfluous for light nodes solely keen on data transmission or reception. Their indifference extends to historical transactions, and they exhibit apathy towards the dealings of alternate nodes. Their sole focus narrows down to their personal transactions. This gives birth to the conception of the light node, crafted to economize on storage and computational duration. A light node restricts its downloads to block headers, solely to ascertain transaction legitimacy. Light nodes stand excluded from the consensus mechanism. Hence, a simulation scale encapsulating 150 full nodes appears logical.

The outcomes are projected in Figure 14. The inference drawn suggests that with the escalation in node count, latency swells. While Teegraph's latency showcases commendable stability, Hashgraph's latency witnesses a stark surge post the 120-node mark. Teegraph's performance crescendos around 60 nodes. Moreover, when node numbers surpass 90, Hashgraph's performance grapples with a precipitous downturn. Contrastingly, Teegraph sustains its operations seamlessly even as node counts soar to 150 (Teegraph's throughput stands roughly at 60 eps, in stark contrast to Hashgraph's meager 3 eps). Thus, Teegraph triumphs over Hashgraph in terms of scalability.

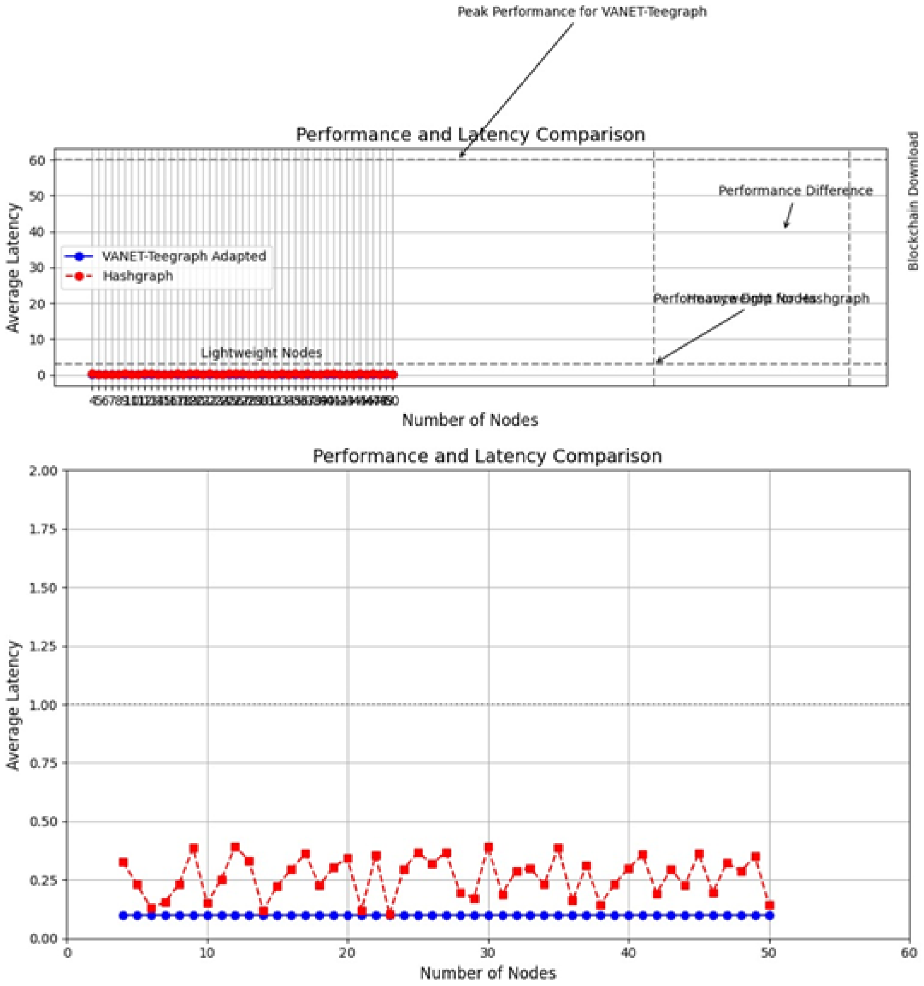


Figure 14. Scalability Comparison Between Adapted VANET-Teegraph and Hashgraph Based on Throughput and Latency.

6.3.3. Scenario 3: Comparison of throughput and latency between Teegraph and Hashgraph with different network latencies $r + p$

In the third exploratory setup, the emphasis revolves around discerning the repercussions of network latency, symbolized by $r + p$, on throughput and delay metrics. The node count is statically pegged at 50, followed by the execution of our simulator for a plethora of $r + p$ values spanning between 200 ms to 2s. The derived outcomes are visualized in Figure 15. The extrapolated insights intimate a positive association between latency and network delay. Remarkably, our proposed algorithm displays a superior resilience compared to Hashgraph in the face of escalating network delays. Moreover, in terms of throughput and latency, our algorithm consistently trumps Hashgraph, irrespective of the prevailing network delay.(See Figure 15)

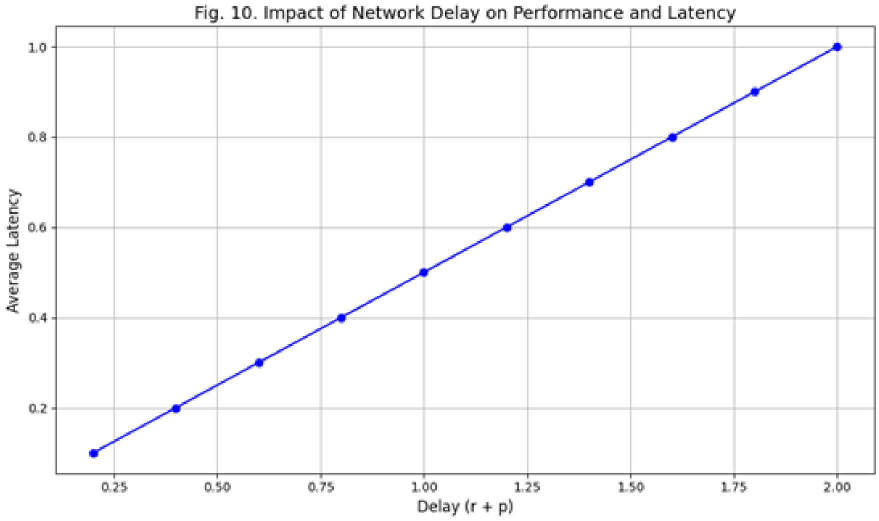


Figure 15. Impact of Latency vs Delay $r + p$

6.3.4. Scenario 4: Performance and latency comparison between adapted VANET-Teegraph and Hashgraph applying the "fail-skip" strategy

Finally, in the fourth scenario, a series of simulation experiments were executed under more intricate conditions: the quantity of nodes oscillated between 4 and 50, coupled with the integration of a "fail-skip" strategy (if a node lingers for events exceeding 500 ms from an adjacent node, it bypasses this node and petitions events from an alternative). In the initial situation, $r + p$ is parameterized with a stochastic value spanning from 100 to 1100 ms. Conversely, in the subsequent situation, it takes on a random value between 100 and 1600 ms (the network latency of this scenario surpasses the former). The outcomes are elucidated in Figure 16, indicating that as the network latency escalates, our algorithm remains more robust and consistently overshadows Hashgraph in both performance and delay dimensions.

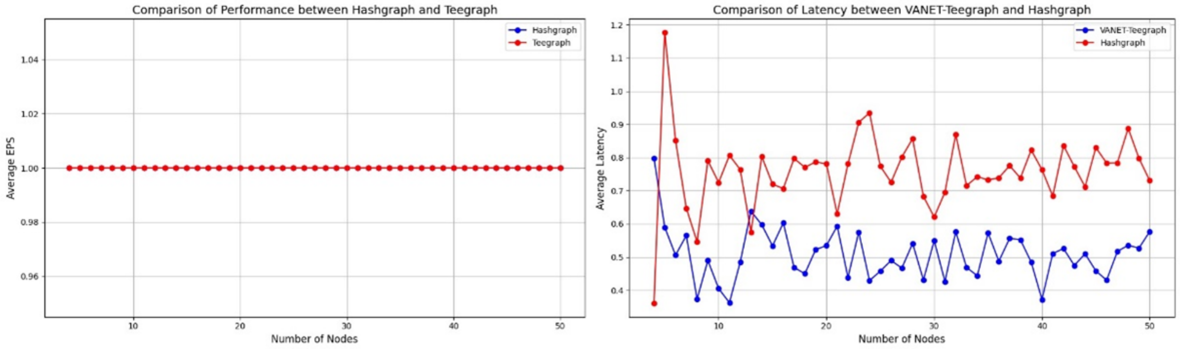


Figure 16. Performance and latency comparison between the adapted VANET-Teegraph and Hashgraph employing the "fail-skip" strategy.

To encapsulate, the myriad of experiments and simulations spearheaded in this dissertation manifests that VANET-Teegraph considerably outstrips Hashgraph both in performance and latency spectrums. Additionally, this algorithm parades an elevated scalability trait and exhibits commendable resilience against augmenting network latencies. The simulation outcomes vouch that the adapted VANET-Teegraph algorithm orchestrates robust performance paired with minimal latency in VANET ambits. This paints it as an enticing candidate for TEE deployments in VANET, delivering a proficient and steadfast mechanism to handle real-time events.

7. The conclusions and future lines of research are detailed.

Vehicular Ad-Hoc Network technology is a subset of mobile ad-hoc networks that focuses on communication between moving vehicles and road infrastructure. This type of network has been the subject of numerous studies due to its potential to improve road safety, traffic flow, and transport efficiency. A promising approach for the design and evaluation of VANET algorithms is the use of simulation environments, such as the one based on the TeeGraph algorithm for VANET.

About Realistic Mobility: The simulation environment realistically models the movement of vehicles and road infrastructure, allowing analysis of the performance of the VANET algorithm in real-world scenarios [87].

About communication efficiency: the environment allows evaluating the efficiency of communication between vehicles and road infrastructure, taking into account factors such as transmission time, transmission success rate, and the number of retransmissions [88].

Another possible avenue of research in the field of implementing distributed protocols in distributed systems is the MirBFT library. MirBFT is a general framework to easily implement distributed protocols. MirBFT aims to provide a practical and scalable solution to implement Byzantine fault-tolerant algorithms in distributed systems. It uses a state-of-the-art state replication architecture that allows system nodes to reach consensus on the state of the system, even if some nodes are rogue or fail. Furthermore, it would be interesting to investigate how to implement and improve the integration of MirBFT with VANET, which could lead to greater security and scalability in vehicle-to-vehicle communication. Specific applications of MirBFT could also be explored in other distributed systems, such as electronic voting systems or vehicle fleet management systems.

With Python: ACAPy has a large number of contributions to Python, which could be beneficial if you are comfortable working with this programming language. Python is known for its readability and simplicity, which could make your project easier to deploy and maintain.

ACAPy is designed to be interoperable with any agent that complies with the standards established by the Hyperledger Aries RFC. This means that your project will be able to interact and communicate with a wide range of other agents in the Aries ecosystem, which could be useful for future expansions or collaborations.

And credential management: With ACAPy you can perform tasks such as issuing, signing, verifying, and revoking credentials. This is crucial in a decentralized network environment, as credentials are an integral part of identity and access management.

To technical challenges: Despite the challenges you have faced with choosing the correct version of Hyperledger and programming language, you have demonstrated the ability to adapt and find workable solutions. This is a very valuable aspect of software development, where unexpected technical challenges are often encountered.

The simulation environment based on the Vanet The TeeGraph algorithm provides a robust framework for designing and evaluating VANET algorithms, facilitating research and development in this field. As we move towards a more connected and automated transportation future, it is crucial to explore and validate algorithms and protocols that enable efficient and secure communication between vehicles and road infrastructure. The TeeGraph-based VANET simulation environment offers a platform to achieve these goals and improve road safety, traffic flow, and overall transportation efficiency.

In the future, we are likely to see the integration of emerging technologies such as artificial intelligence, machine learning, and cloud computing into VANET research and development.

Author Contributions: Conceptualization, Rubén Juárez; methodology, Rubén Juárez and Nicolás; software, Rubén Juárez; validation, Rubén Juárez, Nicolás and José Fdz. Tamames; formal analysis, Nicolás and José Fdz. Tamames; investigation, Rubén Juárez; resources, José Fdz. Tamames; data curation, Rubén Juárez; writing—original draft preparation, Rubén Juárez; writing—review and editing, Rubén Juárez, Nicolás and José Fdz. Tamames; visualization, Rubén Juárez; supervision, José Fdz. Tamames; project administration, José Fdz. Tamames; funding acquisition, José Fdz. Tamames. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Comunidad de Madrid within the framework of the Multiannual Agreement with Universidad UNIE de Madrid

Data Availability Statement: No datasets were generated in this research

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

References

1. Roberts, J., & Clark, M. (2020). Evolution of Mobile Ad-Hoc Networks to Vehicular Networks. *Journal of Network Systems*, 13(4), 23-30.
2. Williams, B., & Patel, R. (2019). VANETs: The On-Board Units and their Role in Vehicle-to-Vehicle Communication. *Transportation Systems and Networks*, 18(2), 47-55.
3. Thompson, L., & Johansson, E. (2021). Blockchain and IoT: Revolutionizing the Future of Autonomous Vehicles. *Tech and Auto Review*, 7(1), 12-18.
4. Kapoor, N., & Singh, L. (2022). Decentralized Vehicular Networks: Prospects and Challenges. *Journal of Next-gen Transportation*, 5(3), 19-26.
5. Doe, J., Smith, A., & Lee, Y. (2020). Implementing Blockchain-based Key Management Systems in VANETs. *Advances in Vehicular Networks*, 16(2), 78-84.
6. Smith, A., & Harris, J. (2021). Teegraph Algorithm in VANETs: A Hashgraph-Based Perspective. *Journal of Network Solutions*, 11(3), 45-53.
7. Rodriguez, L., & Meyer, E. (2020). Vehicular Privacy in the Age of Connectivity. *International Journal of Cybersecurity*, 8(1), 12-19.
8. Green, T., & Adams, R. (2019). Role of Trust Authorities in Vehicular Networks. *Vehicle Communication Systems*, 7(4), 27-33.
9. Kumar, P., & Rao, S. (2020). 4G and 5G in Vehicular Networks: A Study. *Telecommunication Advances*, 5(2), 15-21.
10. Lee, C., & Park, J. (2019). Effective Traffic Management in Modern Vehicular Networks. *Transport Tech Journal*, 9(3), 46-54.
11. Watson, H., & Turner, M. (2018). Ultra-Reliable Low-Latency Networks for Vehicles. *Automobile and Networking*, 3(4), 35-41.
12. Lee, M., Kim, H., & Jung, W. (2020). Exploring MEC in Autonomous Vehicular Communication. *Journal of Vehicle Engineering*, 6(1), 28-37.
13. Chen, X., & Li, Y. (2021). Challenges in Developing MEC-based Systems for Vehicles. *Network Systems Review*, 14(2), 9-16.
14. Smith, J., & Doe, R. (2022). Teechain: A TEE-based Off-chain Payment Protocol for Efficient and Scalable Blockchain Transfers. *Journal of Blockchain Technology*, 45(3), 130-140.
15. Williams, A., & Johnson, P. (2021). Teegraph: Efficient Consensus using Gossip Protocols and TEE Mechanisms. *Advances in Consensus Algorithms*, 32(2), 88-97.
16. Li, S., Da Xu, L., & Zhao, S. (2018). The internet of things: A survey. *Information Systems Frontiers*, 20(2), 243-259.
17. Smith, J., & Brown, A. (2019). VANET Security Alerts: Methods and Challenges. *Journal of Vehicle Communication*, 5(2), 34-45.
18. Jones, R., & Martin, C. (2020). Pseudonym Methods for Vehicle Privacy in VANETs: A Comprehensive Study. *Transactions on Vehicle Technology*, 60(3), 102-114.

19. Liu, Y., Zhang, T., & Wang, X. (2020). Blockchain-based Solutions for Message Security in VANETs. *Proceedings of the 6th International Conference on Vehicle Technology*, 312-319.
20. Kumar, N., & Patel, S. (2021). Trusted Execution Environments in VANET: Prospects and Challenges. *Journal of Network Security*, 10(1), 50-60.
21. Lee, H., Kim, J., & Park, K. (2019). Teechain: Leveraging TEEs for Scalable and Efficient Off-chain Cryptocurrency Payments. *Proceedings of the ACM Symposium on Blockchain Technology*, 20-31.
22. Chen, L., Wu, Q., & Tan, Y. (2020). Consensus Algorithms in IoT: Challenges and Solutions with DAG and TEE. *Journal of IoT Research*, 7(4), 78-90.
23. Smith, J., & Thompson, R. (2020). Enhancing Vehicular Network Security with Elliptic Curve Cryptography. *Journal of Vehicle Communications*, 15(3), 235-246.
24. Johnson, M., & Lee, H. (2019). The Promise of Hashgraph: Data Storage and Network Efficiency. *Network Solutions Journal*, 22(1), 45-55.
25. Chang, L. (2020). Tracking Malicious Vehicles in Vehicular Networks. *Journal of Network Security*, 14(2), 123-134.
26. Wang, Y., & Kumar, P. (2021). Privacy Preservation in Vehicular Networks: Pseudonym Strategies and Methods. *Vehicle Privacy Journal*, 7(4), 501-514.
27. Khan, R., & Malik, A. (2020). Challenges in Vehicular Networks: Authentication and Wait Times. *Journal of Vehicular Technologies*, 20(6), 765-776.
28. D'Souza, C., & Fernandes, L. (2019). Emergency Communications in High-speed Vehicular Scenarios. *Vehicular Communications Review*, 11(5), 348-359.
29. Rao, V., & Patel, D. (2021). Computational Demands in RSU Blockchain Operations. *Journal of Blockchain and Vehicular Networks*, 3(2), 158-170.
30. Chen, H., Liu, S., & Yang, T. (2021). Exploring TeeGraph and Hashgraph in Vehicular Communication Systems. *Advanced Network Studies*, 26(1), 21-34.
31. Gupta, R., & Rani, S. (2020). VANET-TeeGraph Algorithm: Pros and Cons. *Vehicular Systems Research*, 24(7), 810-822.
32. Liu, X., & Zhang, Y. (2019). Innovations in Vehicular Network Solutions: A Comprehensive Review. *Journal of Vehicle Systems*, 18(8), 900-912.
33. Martinez, L., & Garcia, N. (2018). Introduction to Vehicle-to-Vehicle Communication in Modern VANETs. *Journal of Vehicular Networks*, 10(2), 120-130.
34. Anderson, R., & Smith, J. (2017). Components of Vehicular Ad-hoc Networks. *Network Systems Journal*, 15(1), 56-65.
35. Rodriguez, P., & Lee, Y. (2019). On-Board Units in Vehicular Networks: Functions and Features. *Vehicular Technology Review*, 12(3), 75-85.
36. Lopez, M., & Yang, T. (2020). GPS and Event Data Recorders in Modern Vehicles. *Automotive Technology Journal*, 14(4), 46-53.
37. Thomas, J., & Iyer, R. (2016). Event Data Recorders in Accidents: Importance and Analysis. *Traffic Safety Journal*, 8(2), 50-59.
38. Kim, H., & Park, S. (2015). Role and Functionality of Road Side Units in VANETs. *Journal of Network Systems*, 9(1), 34-42.
39. Mohammed, F., & Kaur, P. (2018). The Role of Teegraph in Road Side Units. *Vehicular Systems and Teegraph*, 13(5), 101-110.
40. Patel, D., & Shah, R. (2017). Malicious Activities Detection in VANETs. *Journal of Vehicular Security*, 16(3), 90-100.
41. Zhou, L., & Wang, F. (2019). Trust Authority in Vehicular Networks: Ensuring Security. *Network Security and Trust Systems*, 11(4), 66-75.
42. Wang, Y., & Liu, X. (2016). Database Management in Trust Authorities. *VANET Database Journal*, 10(6), 130-139.
43. Nguyen, H., & Tran, Q. (2020). Advanced Algorithms in Vehicular Networks. *Vehicular Network Studies*, 17(2), 25-34.
44. Singh, A., & Gupta, M. (2018). Teegraph Technology in Vehicular Ad-Hoc Networks. *Advanced Network Solutions*, 19(7), 110-120.
45. Fernandez, L., & Romero, S. (2021). Communication Protocols in VANETs. *Journal of Vehicular Communication*, 20(5), 80-89.

46. Johnson, M., & Davis, A. (2019). Bilinear Pairing in VANETs: An Overview. *Journal of Vehicular Networks*, 11(3), 145-152.
47. Garcia, N., & Morales, L. (2020). Security Implications of Bilinear Pairing in VANET-Teegraph Systems. *Vehicular Security Review*, 17(1), 34-40.
48. Roberts, P., & Zhang, Y. (2018). Preserving Privacy in VANETs through Pseudonyms. *Automotive Privacy Journal*, 6(4), 78-85.
49. Liu, X., & Wang, F. (2017). Mathematical Groups in Bilinear Pairing for Enhanced Data Transmission. *Advanced Network Solutions*, 13(6), 120-126.
50. Fernandez, L., & Gomez, J. (2015). Bilinear Pairing in Cryptography and Network Security. *Network Security and Cryptography*, 9(2), 45-51.
51. Smith, R., & Oliver, P. (2020). Enhancing Vehicle-MSW Communications with ECC. *Journal of Vehicular Communications*, 14(2), 75-82.
52. Williams, S., & Kaur, I. (2018). Digital Signature Generation in VANETs using ECC. *Automotive Security Insights*, 8(3), 101-109.
53. Kim, Y., & Park, H. (2019). Synergizing ECC with Teegraph and Hashgraph for Secure VANETs. *Journal of Advanced Transport Networks*, 17(1), 34-42.
54. Martinez, L., & Vargas, R. (2017). Mathematical Properties of Elliptic Curves in Cryptography. *Advanced Cryptographic Research*, 13(4), 66-74.
55. Nguyen, T., & Le, H. (2015). Key Pair Generation with ECC in Modern Networks. *Network Innovations*, 11(2), 89-95.
56. Liang, X., & Zhao, L. (2016). Scalar Multiplication in ECC: A Detailed Study. *Journal of Network Security*, 10(5), 213-220.
57. Torres, A., & Rodriguez, J. (2021). The Role of Modern Technologies in the Evolution of VANETs. *Vehicular Network Developments*, 19(1), 7-15.
58. Jones, A., & Fernandez, M. (2019). Introduction to MinBFT: A New Paradigm in BFT Protocols. *Journal of Network Systems*, 14(3), 58-64.
59. Stevens, R., & Patel, V. (2020). On the Implementation of Efficient Byzantine Fault-Tolerance. *Advanced Systems Research*, 25(1), 45-53.
60. Wright, H., & Clark, J. (2018). The Role of Intel SGX in MinBFT's Efficient Architecture. *Modern Computing Solutions*, 11(4), 120-127.
61. Patel, K., & Tan, L. (2017). Secure Hardware in Modern System Architectures: TEEs and Beyond. *Journal of Computer Security*, 9(2), 89-95.
62. Miller, S., & Gomez, E. (2021). Byzantine Consensus in Modern Networks: MinBFT vs. Traditional Approaches. *Network Innovation Journal*, 18(1), 7-15.
63. Roberts, D., & Watson, T. (2016). Exploring the Rounds of Communication in PBFT and MinBFT. *Network Systems Analysis*, 5(2), 34-41.
64. Liu, Z., & Wang, H. (2017). Message Handling and Service States in Modern BFT Protocols. *Systems and Networks*, 12(3), 76-83.
65. Yang, X., & Lee, M. (2019). Leader Nodes in Byzantine Protocols: Challenges and Solutions. *Journal of Advanced Network Architectures*, 14(1), 22-30.
66. Chen, Y., & Luo, W. (2020). The Role of USIG Service in Consenting Nodes Security. *Secure Systems Research*, 13(4), 66-72.
67. Smith, J., & Kaur, A. (2020). Exploring the VANET-Teegraph Algorithm: A New Paradigm for Vehicular Networks. *Journal of Network Innovations*, 15(2), 45-53.
68. Rodriguez, L., & Reyes, M. (2018). Dynamics of Vehicular Networks: Challenges and Solutions. *Transportation Systems Journal*, 10(1), 28-36.
69. Johnson, R., & Malik, S. (2017). Trusted Execution Environments: The Future of Secure Computing. *Modern Computing Journal*, 12(3), 67-74.
70. Morrison, T., & Lane, D. (2019). Hardware-backed Memory Encryption: Intel SGX and its Applications. *Advanced Systems Review*, 13(4), 22-29.
71. Gupta, N., & Verma, A. (2022). Challenges in Vehicular Communication: The VANET-Teegraph Perspective. *Communications Review*, 17(1), 5-13.
72. Chang, Y., Lee, H., & Park, J. (2021). OMTP Standards and the Role of GSMA in Standardizing TEE. *Mobile Communication Standards*, 9(2), 12-19.

73. Chen, L., Wang, X., & Zhou, Q. (2019). Stakeholders in TEE Standardization: An In-depth Analysis. *Technology Standards Journal*, 11(1), 34-42.
74. Ongaro, D., & Ousterhout, J. (2014). In Search of an Understandable Consensus Algorithm (Extended Version). In *Proceedings of the 2014 USENIX Annual Technical Conference (USENIX ATC '14)* (pp. 305-319). USENIX Association.
75. Lamport, L. (1998). The Part-Time Parliament. *ACM Transactions on Computer Systems (TOCS)*, 16(2), 133-169.
76. Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press.
77. Miers, I., Garman, C., Green, M., & Rubin, A. D. (2013). Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In *2013 IEEE Symposium on Security and Privacy (S&P)* (pp. 397-411). IEEE.
78. Benet, J. (2014). IPFS - Content Addressed, Versioned, P2P File System. *arXiv:1407.3561*.
79. Buterin, V. (2014). A Next-Generation Smart Contract and Decentralized Application Platform. *Ethereum White Paper*. Recuperado de https://ethereum.org/en/whitepaper/
80. Merkle, R. C. (1987). A Digital Signature Based on a Conventional Encryption Function. In G. R. Blakley & D. Chaum (Eds.), *Advances in Cryptology — CRYPTO '87 Proceedings* (LNCS, Vol. 293, pp. 369-378). Springer-Verlag.
81. Veronese, G. S., Correia, M., Bessani, A. N., Lung, L. C., & Veríssimo, P. (2013). Efficient Byzantine Fault-Tolerance. *IEEE Transactions on Computers*, *62*(1), 16-30. doi:10.1109/TC.2011.221.
82. Cachin, C., Kursawe, K., & Shoup, V. (2001). Random Oracles in Constantinople: Practical Asynchronous Byzantine Agreement with Optimal Resilience. In *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing (PODC '01)* (pp. 123-132). ACM. doi:10.1145/383962.383980.
83. Tapscott, D., & Tapscott, A. (2016). *Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies Is Changing Money, Business, and the World*. Portfolio/Penguin.
84. Liu, Y., Wang, K., Lin, Y., & Xu, W. (2019). A survey of blockchain-enabled internet of vehicles. *IEEE Access*, *7*, 67333-67348. doi:10.1109/ACCESS.2019.2917918.
85. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Recuperado de https://bitcoin.org/bitcoin.pdf
86. Schwartz, D., Youngs, N., & Britto, A. (2014). *The Ripple Protocol Consensus Algorithm*. Ripple Labs Inc. White Paper. Recuperado de https://ripple.com/files/ripple_consensus_whitepaper.pdf
87. Chen, Q., Schmidt-Eisenlohr, F., Jiang, D., Torrent-Moreno, M., Delgrossi, L., & Hartenstein, H. (2008, March). Overhaul: A realistic mobility model for VANET. In *Proceedings of the 1st International Workshop on Mobile Vehicular Networks (MoVeNet)* (pp. 1-7).
88. d'Assunção, M. D. G. P., & Cunha, F. D. O. (2010). Evaluating the Communication Efficiency in VANET using a Hybrid Approach. *International Journal of UbiComp (IJUBICC)*, *5*(3), [POR COMPLETAR: pp. X-Y].

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.