A Privacy-Preserving Authentication Scheme based on Elliptic Curve Cryptography and using Quotient Filter in Fog-enabled VANET

Shidrokh Goudarzi[a], Seyed Ahmad Soleymani[a,*], Mohammad Hossein Anisi[d], Mohammad Abdollahi Azgomi[a], Zeinab Movahedi[a], Nazri Kama[e], Hazlifah Mohd Rusli[e] and Muhammad Khurram Khan[f]

[a] *School of Computer Engineering, Iran University of Science and Technology, Hengam St., Resalat Sq., 16846-13114, Tehran, Iran.*

[d] *School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, United Kingdom.*

[e] *Fakulti Teknologi & Informatik Razak, Universiti Teknologi Malaysia, 54100 Kuala Lumpur, Malaysia.*

[f] *Center of Excellence in Information Assurance (CoEIA), College of Computer & Information Sciences, Building 31, King Saud Universaity, P.O. Box 92144, Riyadh 11653, Kingdom of Saudi Arabia.*

ARTICLE INFO

Keywords:
Authentication
Privacy
Quotient Filter
Fog Computing
Big Data
VANET

ABSTRACT

Security and privacy are considered as two main challenges in Vehicular Ad Hoc Network (VANET). To cope with these challenges and in order to improve the safety of VANET, we propose a secure and privacy-preserving authentication scheme. In the proposed scheme, Quotient Filter (QF) is used to address node authentication while message authentication is done based on Elliptic Curve Cryptography (ECC). Besides, each vehicle is mapped to a different pseudo-identity to preserve privacy in VANET. Moreover, due to the higher computational capabilities of Fog Nodes (FN) compared to Road-Side Units (RSUs), they are distributed over the side of the road to minimize the latency of the security model and enhance the system throughput. Our security analysis demonstrates that the proposed scheme is able to identify illegitimacy vehicle nodes and invalid messages when the fog-enabled VANET is exposed to attacks. Furthermore, the performance evaluations prove the effectiveness of our work compared to the existing studies.

1. Introduction

The smart city's purpose is to create smart and intelligent environments using enabling technologies such as the Internet of Things (IoT) to enhance the life quality of people. In a smart city, IoT characterizes a cyber-physical paradigm, where a wide range of real physical elements is capable to interact with each other autonomously. This type of consistent network is an empowering agent for Intelligent Transportation Systems (ITS) as well. [1]. ITS aims to improve the safety, mobility, and efficiency of transportation. VANET, as a key component of ITS, has obtained great attention from both industry and research communities. It is also an important network infrastructure in the Industrial Internet of Things (IIoT) [2], that creates an intelligent space for vehicular communications. VANET has several applications in IIoT and ITS applications such as weather reporting using the vehicles, road safety improvement and infotainment dissemination [3].

In VANET, vehicles and infrastructures are connected through an open wireless channel and data transmission between such entities is performed over a public channel. Due to the open nature of wireless communication, all entities and data can be threatened by different types of attacks which raises concerns on security and privacy. In such areas, the malicious attackers can modify the messages sent from a vehicle, and/or disguise themselves as vehicles if there are no adequate security considerations for VANET. Private and sensitive data such as driving route or identity can also leak when there is a lack of a proper privacy scheme [4].

To address the security and privacy concerns, it is required to protect the network using robust, secure and efficient authentication and privacy-preserving schemes [5]. An authentication scheme can certify the legitimacy of vehicle nodes and integrity of the message while the privacy-preserving can maintain the sensitive information protected and private [6]. A number of studies such as [7, 8] have focused on the security of data-in-transit over the communication channel in VANET. However, due to the short communication range and high speed of vehicle nodes as well as big data generated on the edge of the network, they couldn't satisfy the trade-off between performance and security. Therefore, minimizing communication and computation overhead as well as latency using a robust and secure scheme is essential.

Probabilistic Data Structure (PDS) and fog computing are two technologies that could be used to deal with the aforementioned issues. PDS is a data structure that is particularly suitable for big data as it is able to reduce analytical procedures and latency [9]. Fog computing also reduces latency by moving the part of

[*] Corresponding author
s.soleymani@surrey.ac.uk (S.A. Soleymani)

the computational power to the edge of the network [10]. Besides, fog computing can address big data issues by providing elastic resources with low latency for the systems that produce large-scale data [3]. In our previous work [4], we proposed a privacy-preserving node and message authentication scheme along with a trust model.

In this study, to deal with the security and privacy challenges in VANET, an identity-based authentication scheme with privacy-preserving is designed. In the proposed scheme, fog computing and authentication process are integrated wherein fog nodes are distributed along the side of the road. In the existing works, the authentication and integrity of all event messages received from vehicle nodes and other entities, legitimate or illegitimate, are evaluated by the receivers. However, this increases latency and network congestion due to the large amount of data generated. To address this issue, the proposed work verifies the node's legitimacy prior to initiating the communication. Verification of node authenticity is started by a query on the fog node's QF. Then, the receiver of the message needs to verify the message's integrity by using single/batch signature verification.

The key contributions of this work are as follows:

1) A fog computing-based VANET architecture for latency reduction and throughput improvement is designed.

2) A QF-based node authentication scheme to verify the vehicle node's legitimacy is developed. The aim is to deal with illegal and unauthorized nodes attempting to join the network and share data.

3) A scheme based on ECC is proposed for message authentication in order to guarantee and support the integrity of the event messages. This scheme includes message's signing and single/batch signature verification. Besides, to preserve the privacy of vehicle nodes, the pseudonym is employed.

4) The OMNET++ is used to simulate and measure the impact on transmission delay under different densities and velocities with different percentages of malicious nodes distributed in the network.

The remaining of this article is structured as follows. In Section 2, information and background are provided in detail. Related works on security models for VANET are clarified in Section 3. In Section 4, the network model and security requirements are presented. Section 5 presents the proposed scheme. Section 6 presents the analysis of security proof for the scheme. The performance among state-of-the-art methods is compared in Section 7. Finally, the conclusion and future work are provided in Section 8.

2. Preliminaries

Here, there exists a general definition fog computing, and QF.

2.1. Fog Computing

Fog computing extends cloud computing capabilities to the network edge [11]. It is a virtual platform that connects traditional cloud servers and end-users or things by offering storage, computing, and networking capabilities. Local resource pooling and data processing, cache data management, load balancing, and latency reduction are just a few of the benefits of combining VANET with fog computing [12]. In a fog computing-based VANET, the time-critical data are analyzed locally through the fog node tools which reduces the latency. Notably, fog computing facilitates interactions between vehicle nodes and allows for effective collaboration amongst nodes [13].

2.2. Quotient Filter

QF, as a cache-friendly and space-efficient probabilistic data structure, is useful and beneficial for big data sets since it decreases latency and simplifies the analytical process. It represents a multiset of elements $S \subseteq U$ by storing p bit fingerprint for each element. QF stores the multiset $F = h(S) = \{h(x) \mid x \in S\}$, where $h : U \rightarrow \{0, \cdots, 2p - 1\}$ is a hash function.

It is assumed that there exist a hash table T with $m = 2^q$ buckets for storing F utilizing the quotient method [14]. In this method, a fingerprint f is divided into its r least significant bits, $f_r = f \ mod \ 2^r$ (the remainder), and its $q = p - r$ most significant bits, $f_q = \lfloor f/2^r \rfloor$ (the quotient). For inserting a fingerprint f into F, we store f_r in bucket $T\left[f_q\right]$. Considering a remainder f_r in bucket f_q, the full fingerprint can be exclusively reconstructed as $f = f_q 2^r + f_r$ [15].

3. Related Work

Security and privacy are among the most important issues in to the vehicular networks that have been taken into consideration by many researchers.

A signature authentication scheme based on Public Key Infrastructure (PKI) is developed in [16]. In this scheme, in order to hide the real identity of the vehicle node, they used anonymous certificates. To this purpose, each vehicle has to store a number of anonymous certificates. In authentication process, to avoid traceability, the vehicle uses different private/public key pairs. However, the growth in vehicle numbers and in addition key change frequency, because of the high velocity of vehicles, lead to increasing the number of keys. This issue causes the complexity of the key management and storage and thereby may unable to satisfy the stringent time requirement of the vehicular communication applications.

To tackle issues concerning the PKI-based schemes, an identity-based batch message signature verification scheme for the Vehicle to Infrastructure (V2I) communications named RAISE is presented in [17]. Multiple received messages are simultaneously validated and verified by the RSUs in this approach. Compared to the schemes that each message is verified by RSU separately, the overall authentication overhead is considerably decreased, resulting in improved operating efficiency for VANETs. Although this scheme enhances efficiency, however, it fails for example when the number of vehicles is much because the RSU has to maintain the extra ID-Key table. This issue resulting in more storage costs. Moreover, the key agreement process still executes the exponent operations, which leads to a high computation cost.

In [18] a scheme is developed for message verification in which RSU helps neighboring vehicles to authenticate their messages received. In other words, message verification is one of the main tasks of RSU and it serves as the cloud. To this end, RSU uses batch message verification to authorize multiple messages. When batch verification is completed successfully, all messages in the batch are valid. In contrast, when at least one invalid message exists in the batch, it will be discovered by a binary search. The RSU assigns two positive and negative bloom filters, respectively, to store the hash value for valid and invalid messages. Then, the positive and negative filters will be distributed by the RSU at a particular frequency to neighboring vehicles. Therefore, vehicles just need to investigate the two filters for the authentication of messages. Authentication is considerably reduced by this scheme and the overall efficiency of the system has improved. However, with the presence of a large number of vehicle nodes, the RSU's computing performance will suffer, producing significant delays.

To address this problem, [19] stated the possibility of sharing the computational load on the RSU with adjacent vehicles. The system in this work chooses proxy vehicles based on calculation power. The selected vehicles will share the verification of the messages performed by the RSU and then the verification results will be sent to the RSU. Next, the accuracy of the results will be evaluated by RSU. Although the RSU's verification performance has increased dramatically by the suggested scheme, however, the scheme's performance is insufficient because the basic operations require map-to-point and bilinear pairing, both of which have significant overhead.

An Identity Batch Verification scheme (IBV) is designed by Zhang et al. in [20] for VANETs. The overall confirmation delay of batch message signatures is reduced with this scheme. Besides, in comparison to PKI-based systems, it is also faster. However, this scheme with a huge overhead would lead to performance issues as it is based on bilinear pairing. This is a common problem among all proposed authentication schemes based on bilinear pairing [21, 22].

In order to reduce the computation overhead created by the bilinear pairing method and map-to-point hash function, He et al. [23] proposed a scheme based on ECC. In this scheme, the process of signature generation has been simplified which improves efficiency. This scheme can achieve lower computational overhead compared with traditional pairing-based conditional privacy-preserving authentication schemes for VANETs. However, this scheme is unsuitable for delay-sensitive applications because of the high transmission overhead.

An ECC-based anonymous privacy-preserving authentication scheme is proposed for VANET in [24]. In this scheme, each message transmitted by a vehicle needs the verification of RSUs. However, the aggregate signature verification has a leak by which a malicious user can construct bogus signatures and muddle throughput the aggregate verification. Also, to meet privacy, each vehicle has a group of pseudo-IDs which increases the memory usage. They also proposed an authentication scheme for the Internet of vehicles in [25]. This scheme is certificate-less scheme that satisfy privacy. In this scheme, each traffic message needs to be verified by RSUs. However, because of the big data generated in vehicular network, it increases RSU overhead communications and in result reduce operational efficiency.

In [26], a scheme based on ECC is proposed to message authentication. In this work, for improving message authentication efficiency, a few vehicles are selected as edge nodes to support the RSUs with the message's authentication. It is supposed that RSUs act as the cloud of the vehicles. However, given the very dynamic topology of the network that is related to the high velocity of vehicles, considering vehicle as the edge node cannot be suitable. Also, vehicles are more threatened by destructive nodes, and the selection of reliable vehicles, as the edge nodes, is an important issue. In contrast, RSUs have a high ability in computation than vehicles. Also, since it is difficult for RSU to be threatened by destructive nodes, hence they are more trustable and reliable than vehicle nodes.

Based on available knowledge, there is a lack of a proper security and privacy scheme with the lowest computation overhead, communication overhead, and latency in VANET wherein the number of vehicle nodes and data generated are huge and vehicles also moving fast. In this network, efficient security and privacy scheme are required that not only needs to ensure the legitimacy of vehicle nodes, the integrity of the message, and meet privacy-preserving but also deal with concerns related to big data.

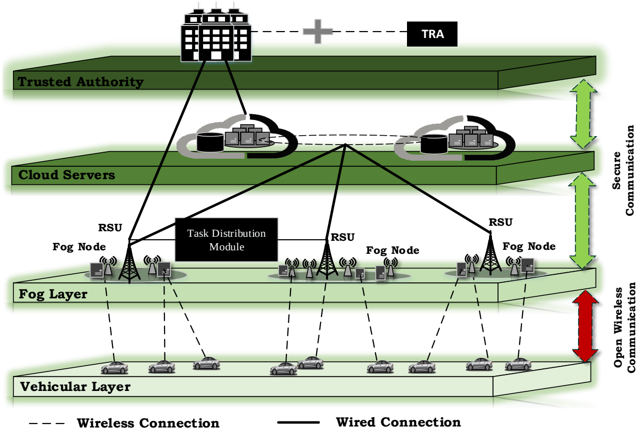


Figure 1: Fog-based VANET architecture.

4. System Architecture and Security Requirements

In this section, we define the main entities involved in the proposed scheme as well as the security requirements.

4.1. Network Model

As shown in Figure 1, the architecture of the proposed fog-enabled VANET architecture includes two layers: upper and lower. The upper layer comprises Cloud Servers (CS) and root Trusted Authority (TA), whereas fog nodes, RSUs, and vehicle nodes are located at the lower layer.

Upper Layer: In this layer, cloud servers are used to provide high computing power as well as permanent and reliable data storage, whereas TA generates the master secret keys, global system parameters, and credentials for the vehicle nodes and fog nodes. TA is also responsible for recovering the vehicles' real identities that sign and disseminating bogus messages. As part of TA, there is a TRace Authority (TRA) that creates pseudonyms for vehicle nodes and tracks the real identity by using the vehicle's pseudonyms.

Lower Layer: Fog layer and vehicular layer are included in the lower layer. In the fog layer, fog nodes and RSUs are deployed along the roadside. RSUs are supposed to host the fog nodes and connect them to cloud through a secure wired communication such as Ethernet. Also, RSUs are equipped with persistent links to a service provider hosted on the cloud and communicate with the vehicles with via range communication capabilities. RSUs continuously monitor various parameters and transmit the required aggregated data to the FNs. RSUs are also able to generate a notification for vehicle nodes whenever required. FNs are equipped with communication capabilities, processing power, and storage space. They interacts with vehicle nodes which are within its communication area via open wireless technologies such as 4G/LTE/5G. It's worth noting that FNs, which have much higher processing capacity comparing to RSUs, experience lower latency and posses higher throughput.

In the vehicular layer, to improve traffic security and regional traffic operational efficiency, The traffic-related data is broadcast by the vehicle nodes to the nearby and local region on a regular basis by using IEEE 802.11p protocol. Vehicles with internal sensors may detect events that occur within the transmission range. In order to keep parameters and keys received from TA secure, each vehicle is equipped with a realistic Tamper-Proof Device (TPD). The medium utilized for communications between vehicles and fog nodes is 5.9-GHz DSRC recognized as IEEE 802.11p.

Figure 2 illustrates a simplified representation of how RSU and FN are employed in the fog layer to aid vehicles in moving from one geographic point to another. RSUs are thought to cover the entire network, while the communication range of FN covers a sector of the vehicular environment and can include several intersections [27]. When a vehicle node is physically placed inside the fog nodes' communication range, it has the ability for sending and receiving data from them. For example, when a vehicle is located inside a fog node-covered area, it will report its speed, current location, and road conditions to the fog node on a regular basis until it exits the area. Hence, a vehicle will be continually supported by fog nodes. When a vehicle node is covered by numerous access fog nodes or RSUs, it must choose the most appropriate RSU/FN for sending and receiving data. To this purpose, the vehicle node analyzes and calculates the link quality between itself and any neighboring FNs, if any are present. Otherwise, it computes quality of link with the existing RSUs. As mentioned in [28], some parameters such as bandwidth, Bit Error Rate (BER), and Signal the Noise Ratio (SNR) can be used to measure the link quality. However, it is out of the scope of this paper.

Additionally, given the massive tasks, generated by vehicles, to be processed by FNs, computing power, memory, and CPU availability, as well as tasks loaded need to be monitored. To address this and in order to collect information on distributed fog nodes, a module is developed on the RSUs to compute the tasks locally and offload them to the fog nodes for processing. The task distribution mechanism greatly reduces the delay for the latency-sensitive applications and enhances the overall system scalability.

4.2. Security Requirements

As mentioned in [26], a well-designed privacy preserving authentication scheme should meet the following security requirements:

1) **Message Verification and Integrity:** An FN verifies the signed message has not been forged or

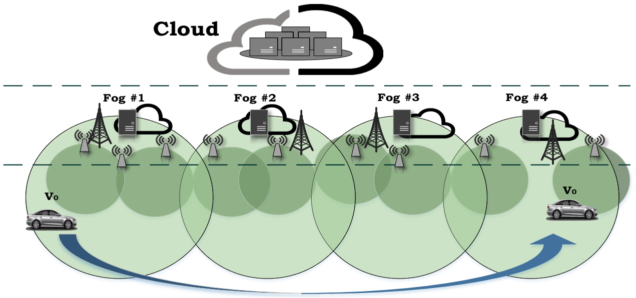


Figure 2: Vehicle node mobility in fog computing.

modified by malicious nodes once receives the message from the authorized vehicles.

2) **Resistance to Unauthorized Nodes:** An unauthorized unregistered node is unable to join the network or communicate with other nodes.

3) **Preserving Privacy:** The vehicle nodes should remain anonymous and no third party extracts the real identity and private information from the vehicle's pseudo-identity.

4) **Resistance to Replay Attack:** A malicious node is unable to store the gathered signed messages and disseminate it when the validity of the message is expired.

5) **Traceability:** Only TRA can trace a vehicle's real-identity by analyzing the pseudo-identity.

5. Proposed Scheme

Security and privacy are real significant in VANET since it is an open-access environment [6]. Building on this, we designed a secure and efficient message and node authentication scheme with privacy-preserving. In the proposed scheme, node authentication is based on QF whereas message authentication is established on ECC. In order to meet the privacy-preserving, mapping each vehicle is also performed to a different pseudo-identity. The following phases related to our scheme are described in this section: (i) initialization, (ii) registration, and (iii) authentication.

5.1. Initialization Phase

In this phase, TA produces the required parameters of system. These parameters will be preloaded into the TPD of vehicle nodes and memory of fog nodes. Let two primes p, q; group G of order q; and consider two distinct generators $P, Q \in G$. TA randomly chooses an at least 160 bits number $s \in Z_q^*$ as the master private key. It also computes the corresponding public key $P_{pub} = s.P$ by using the master private key. Then, the TA selects a secure SHA-256 hash function $h : \{0,1\}^* \rightarrow Z_q$. This is mainly because reconstructing the initial data from the hash value generated by SHA-256 is tricky. Also, it is impossible that SHA-256 creates the same hash value for different messages.

Table 1
Definition of Notations in the Proposed Scheme

Model	Method
$\oplus$	XOR operation
$\|\|$	Concatenation operation
TA	Trusted authority
TPD	Tamper-proof device
TRA	Trace authority
CS	Cloud server
RSU	Roadside unit
FN	Fog node
V	Vehicle node
h	Hash function
PID	Pseudo-identity
RID	Real identity
P_{pub}	System public key
G	Cycle additive group
s	System private key
$params$	Public parameters of the system
P, Q	Distinct generators of G
$\tau, \tau^{'}$	Signature generated by vehicle and RSU/FN, resp.
t	Timestamp of message
VP	Timestamp of pseudo-identity

Then $params = \left\{p, q, a, b, G, P, P_{pub}, h\right\}$ as system's public parameters will be set and published by TA to the cloud servers, RSUs, fog nodes, and vehicles where a and b are parameters of elliptic curve function $E_P\left(a, b\right) : y^2 = x^3 + ax + b \left(mod \quad p\right)$. Table 1 represents the notations utilized in this study.

5.2. Registration Phase

In this phase, TA accomplishes the registration of vehicles, RSUs, fog nodes, and cloud servers as follows:

1) **Fog Node**: Let $\mathfrak{F}_{FN} = \{FN_1, FN_2, \cdots, FN_M\}$ be a list of registered FNs in the network. Each $FN_k \in \mathfrak{F}_{FN}$ has a conventional private key s_{fn}, public key PUB_{fn}, and a unique identity RID_{FN_k}. The public key and real identity are known by the TA while private key is kept secret by fog node.

2) **Vehicle Node**: Considering a list of authorized vehicle nodes that have been registered in the network $\mathfrak{B}_v = \{V_1, V_2, \cdots, V_N\}$. Each vehicle $V_l \in \mathfrak{B}_v$ has a conventional private key VSK_l and public key VPK_l that the VPK_l is known by TA. For each vehicle node, TA chooses a unique identity RID_{V_l} and password PWD_{V_l}. It sends $Y_l = \left\{RID_{V_l}, PWD_{V_l}, s\right\}$ to the vehicle node, securely. To this end, TA firstly signs Y_l by using its private key SK_{TA} and sends the encrypted message $Z = Enc_{VPK_l}\left\{Y_l, SIG_{SK_{TA}}(Y_l)\right\}$

to the vehicle node through RSU. When Z is received, the vehicle node needs to decrypt Z in order to obtain $\{RID_{V_l}, PWD_{V_l}, s\}$ and verifies the signature using PK_{TA}. If it holds, vehicle node preloads Y_l to the TPD. In this work, each vehicle uses $PID_V = \{PID_{V,1}, PID_{V,2}\}$ as pseudo-identity generated by TPD and TRA to meet the privacy requirements that we explain more next.

3) **Roadside Unit**: Let a list of registered RSUs in the network $\mathfrak{R}_{rsu} = \{RSU_1, RSU_2, \cdots, RSU_L\}$. In this network, each $RSU_j \in \mathfrak{R}_{rsu}$ has a unique real identity RID_{rsu_j}, private secret key s_{rsu} and public key PUB_{rsu}. It is assumed that RSUs are trusted and hard to be compromised.

4) **Cloud Server**: Let $\mathfrak{C}_{CS} = \{CS_1, CS_2, \cdots, CS_P\}$ be a list of authorized and registered cloud servers in the network, wherein $CS_i \in \mathfrak{C}_{CS}$ has a unique identity RID_{CS_i}, master private key s_{cs}, and public key PUB_{cs}. Cloud servers also are fully-trusted and hard to be compromised.

In this work, TA-RSUs, TA-CSs, and CSs-RSUs communicate using a secure transmission protocol, such as the wired Transport Layer Security (TLS) protocol [29]. It is also worth noting that, since privacy is not a concern or a requirement for fog nodes and cloud servers, they sign messages with their real identities.

5.3. Authentication Phase

In this section, we explain both node and message authentication and verification procedures as follows:

5.3.1. QF-based Node Authentication Scheme

In the vehicular network, data exchanged among the nodes is the basis of the network. To ensure security, the receiver of data must verify the legitimacy of the sender before initiating any data sharing. Due to the big data generated in the network and a large number of vehicle nodes, we developed a QF-based node authentication scheme. As described above, QF is a probabilistic data structure to query massive datasets that decreases processing overhead and improve security [30].

In this scheme, each vehicle V is equipped with a quotient filter QF_V to maintain the information related to the legitimate and illegitimate vehicle nodes. Depending on the authorized and or unauthorized of vehicle nodes belonging to the RSU_k, they will be recorded in the relevant and appropriate quotient filter of the vehicle using the Equation 1:

$$(QF_V) \leftarrow h\,(fingerprint\,(PID_W) \oplus PUB_{rsu}) \parallel A/U \tag{1}$$

where PID_W refers to the fingerprint of pseudo vehicle identity, PUB_{rsu} is a public key provided by the related RSU, and A and U represent respectively authorized and unauthorized.

Algorithm 1: Pseudo-code for V-to-V Node Authentication

```
$V_i$: Sender
$V_j$: Receiver
$FN$: Relevant fog node
Performs $Query(QF_{V_j})$ by $V_j$
if $(True, A) \leftarrow Query(QF_{V_j})$ then
    Establish a link with $V_i$ and perform
      message authentication
if $(True, U) \leftarrow Query(QF_{V_j})$ then
    Ignore the request from $V_i$
if $(False) \leftarrow Query(QF_{V_j})$ then
    Performs $Query(QF_{FN})$ by $FN$
    if $(True, A) \leftarrow Query(QF_{FN})$ then
        Establish a link with $V_i$ and perform
          message authentication
        Update $QF_{V_j}$
    if $(True, U) \leftarrow Query(QF_{FN})$ then
        Ignore the request from $V_i$
        Update $QF_{V_j}$
    if $(False) \leftarrow Query(QF_{FN})$ then
        Ignore the request from $V_i$
End
```

RSU updates the QF_V of the legitimate vehicle nodes who are under its transmission range immediately after a change in the list of authorized and unauthorized vehicle nodes. It will be performed by broadcasting the new list to the nearby vehicle nodes.

As the same way, each FN maintains own quotient filter QF_{FN} of all genuine and fake vehicle nodes. Like the QF_V, all QF_{FN} continuously upgraded by the relevant RSU.

In a vehicle-to-vehicle communication, prior to data sharing, the receiver V_j executes a $Query(V_i)$ on its QF_{V_j}. If the query returns $TRUE$ with A, it means the V_i is a legal and genuine node, and if it returns $TRUE$ with U, it means the V_i is an unauthorized node otherwise, if the query returns $FALSE$, it means the V_i is not a member of the QF_{V_j}, hence V_j immediately sends a request to the FN_k. Upon receiving a request from the V_j, FN_k checks the legitimacy of vehicle node V_i by executing a query on QF_{FN}. If the query on QF_{FN_k} returns $TRUE$ with A, V_i and V_j start data sharing and then V_j updates QF_{V_j}, but if the query returns $TRUE$ with U, V_j stop any communication with V_i and updates QF_{V_j}. Otherwise, if the query on QF_{FS_k} returns $FALSE$, it means that V_i has not been registered in the network and so it is an illegal vehicle node that has entered the network. In this regard, V_j has to wait for a

Algorithm 2: Pseudo-code for V-to-FN Node Authentication

```
$V_i$: Sender
$FN$: Receiver
Performs $Query(QF_{FN})$ by $FN$
if $(True, A) \leftarrow Query(QF_{FN})$ then
    Establish a link with $V_i$ and perform
      message authentication
if $(True, U)$ or $(False) \leftarrow Query(QF_{FN})$ then
    Ignore the request from $V_i$
End
```

certain time to receive a reply from the FN_k. If V_j did not receive a reply, it rejects the request to communicate with V_i. Algorithm 1 shows the pseudo-code of node authentication in vehicle to vehicle communication.

In a vehicle-to-fog node communication, FN_k performs the query on its QF_{FN_k}. If the query returns $TRUE$ with A, the link between vehicle and FN will be established because the vehicle node is authorized. Otherwise, if the query returns $TRUE$ with U or it returns $FALSE$, the link request will be rejected by FN (see Algorithm 2).

5.3.2. *Message Authentication Scheme*

In fog computing-based VANET, raw data can be gathered by sensors installed on the vehicle node, and stored in on-board storage. Because of the redundancy of the raw data, the processing of data is conducted to extract valuable information. Then, for further processing in terms of integrity and reliability of data, the vehicle node signs the extracted information and sends it to the relevant FN/RSU. After verifying the vehicle's signature and checking the data reliability, FN/RSU also signs the message and broadcasts it in the vehicular network. Once a vehicle node received a signed message from FN/RSU, it checks the signature first and then signs the message for broadcasting to the neighbour vehicle nodes and nearby FNs/RSUs.

Based on the designed network architecture (see Figure 1), Vehicle and Vehicle (V-V), Vehicle and FN (V-FN), Vehicle and RSU (V-RSU), FN and RSU (FN-RSU), and RSU and CS (RSU-CS) are conceivable. It supposes that both FN-RSU and RSU-CS communication are via a secure manner. Therefore, we focus on other communications and explain the message authentication in the following.

A) V-FN COMMUNICATION

[Message signing by Vehicle]: Each message should be signed by the vehicle before being sent to neighbor nodes to verify authentication and message integrity. Besides, in order to maintain anonymity, each vehicle node must utilize its own pseudo-identity.

TPD chooses a number $r_i \in Z_q^*$ randomly and calculates $PID_{i,1} = r_i.P$ to generate pseudo-identity. When a vehicle node joins the VANET, TPD sends a secure message $\{RID_i, PWD_i, PID_{i,1}\}$ to TRA for verifying $\{RID_i, PWD_i\}$. To this end, vehicle firstly signs $SIG_{VSK_i}(RID_i, PWD_i, PID_{i,1})$ using its private key VSK_i and then encrypt it using the TRA public key $Enc_{PK_{TRA}}(SIG_{VSK_i}(RID_i, PWD_i, PID_{i,1}))$. After decrypting the received message and verifying the signature using the vehicle's public key VPK_i, TRA calculates the pseudo-identity $PID_i = \{PID_{i,1}, PID_{i,2}, VP_i\}$ by choosing a random number $r_i \in Z_q^*$, where $PID_{i,2} = RID_i \oplus h\left(r_i.P_{pub}, VP_i\right)$, and VP_i defines the valid period of the PID_i. The generated pseudo-identities are valid within VP_i. This frequent change is mainly because when a vehicle constantly uses a pseudo-identity within the communication, an adversary can trace the vehicle movement trajectory.

Then, vehicle V_i has to sign the $M_i = PID_{i,2} \parallel m_i \parallel t_i$ where M_i is combining of $PID_{i,2}$ as a part of pseudo-identity, m_i as message and t_i as the timestamp guarantees the signed message's freshness against a replay attack. TPD selects $r_i \in Z_q^*$ to sign the M_i. Next, it computes the corresponding signature $\tau_i = r_i + s.h(M_i)$ on M_i for PID_i. Then, the vehicle sends $\{PID_i, M_i, \tau_i\}$ to the relevant RSU/FN.

[Message verification by FN]: When a FN gets a signed message from a vehicle, it must not only validate the vehicle node's authentication, but also verifies the message's signature. It ensures the vehicle is not attempting to impersonate legitimate vehicles or spread false message. If the vehicle node is genuine (see Section 5.3.1), it verifies the signed message as follows:

- Single Message Verification: Once a fog node $FN_j \in \mathfrak{F}_{FN}$ receives a signed message $\{PID_i, M_i, \tau_i\}$, after checking the freshness of $t_i - t_c \leq \Delta t$ and VP_i, if the message and pseudo-identity have not expired, it calculates $h(PID_{i,2} \parallel m_i \parallel t_i)$ and verifies whether

$$\tau_i.P = PID_{i,1} + P_{pub}.h(PID_{i,2} \parallel m_i \parallel t_i) \tag{2}$$

Equation 2 holds or not. If so, the message will be verified; otherwise, FN_j discards the message and recommend the vehicle with PID_i as an illegal vehicle node to the relevant RSU.

- Batch Message Verification: Once the fog node FN_j receives multiple signed messages from vehicles in a time interval, it uses the batch message verification method as follows:

Consider n distinct vehicles $\mathfrak{V}_V = \{V_1, \cdots, V_n\}$ and corresponding message-signature tuples

$$SML = \{\{PID_1, M_1, \tau_1\}, \cdots, \{PID_n, M_n, \tau_n\}\}$$

To sign verification, the fog node FN_j computes $h(PID_{i,2}||m_i||t_i)$ for $i = 1, \cdots, n$ and then checks whether

Equation 3

$$\left(\sum_{i=1}^{n} v_i.\tau_i\right).P = \left(\sum_{i=1}^{n} v_i.PID_{i,1}\right) + \left(\sum_{i=1}^{n} v_i.h\left(PID_{i,2}||m_i||t_i\right)\right).P_{pub} \tag{3}$$

is established or not. If holds, it indicates that the checking was successful and that the signatures should be accepted; otherwise, it indicates that at least one message in the batch is invalid. In the following, due to the $P_{pub} = s.P$, $PID_{i,1} = r_i.P$, $PID_{i,2} = RID_i \oplus h\left(r_i.P_{pub}\right)$, $M_i = PID_{i,2} \parallel m_i \parallel t_i$ and $\tau_i = r_i + s.h\left(M_i\right)$, we prove the validation of the batch message verification.

$$\begin{aligned}
\left(\sum_{i=1}^{n} v_i.\tau_i\right).P &= \left(\sum_{i=1}^{n} v_i.\left(r_i + s.h\left(M_i\right)\right)\right).P \\
&= \left(\sum_{i=1}^{n} v_i.r_i\right).P + \left(\sum_{i=1}^{n} v_i.s.h\left(M_i\right)\right).P \\
&= \left(\sum_{i=1}^{n} v_i.r_i.P\right) + \left(\sum_{i=1}^{n} v_i.s.P.h\left(M_i\right)\right) \\
&= \left(\sum_{i=1}^{n} v_i.PID_{i,1}\right) + \left(\sum_{i=1}^{n} v_i.h\left(M_i\right)\right).P_{pub} \\
&= \left(\sum_{i=1}^{n} v_i.PID_{i,1}\right) + \left(\sum_{i=1}^{n} v_i.h\left(PID_{i,2} \parallel m_i \parallel t_i\right)\right).P_{pub}
\end{aligned}$$

After the fog node FN_j has completed the batch message verification, to discover the invalid and incorrect messages in the batch, a recursive method based on the binary search is used (see Algorithm 3).

In this algorithm, we have considered a batch segmentation and both single and batch message verification. The desired batch contained signed event messages from $Lindex$ till $Hindex$ will be divided into two separate batches by this algorithm. The first batch is from $Lindex$ till $Mindex = (Lindex + Hindex)/2$ and the later one is from $Mindex + 1$ till $Hindex$. After each segmentation, the batch message verification will be used to verify the new batches. If each new batch holds Equation (3), existing messages in the batch will be inserted to the *vML* and algorithm immediately will be stopped for this batch. Otherwise, segmentation will be continued until finding invalid message(s). When there is two or one message in the batch, the single message verification by Equation (2) will be used to check validity of the message. If Equation (2) is established, this message will be inserted into the *vML*, otherwise, it goes to *ivML*.

The output of the algorithm is two lists namely *vML* and *ivML*. Finally, the fog node FN_j signs the $List = \{vML, ivML\}$ and sends to the related RSU. When the RSU receives the list from a fog node, it verifies the signature using the fog node public key PUB_{fn}, and then upgrade its quotient filter QF_{RSU} and nearby fog nodes QF_{FN} .

Algorithm 3: Pseudo-code for valid/invalid message detection in the batch using binary search

```
vML: valid messages
ivML: invalid messages
BMV: batch message verification
SMV: single message verification
SiML: A list of signed messages

if $BMV(SiML, Lindex, Hindex) == true$ then
    $vML$.insert($SiML[Lindex, \cdots, Hindex]$)
    return 1
else
    if $Lindex == Hindex$ then
        if $SMV(SiML[Lindex]) == true$ then
            $vML$.insert($SiML[Lindex]$)
            return 1
        else
            $ivML$.insert($SiML[Lindex]$)
            return 1
    else
        $Mindex = (Lindex + Hindex)/2$
        $BMV(SiML, Lindex, Mindex)$
        $BMV(SiML, Mindex + 1, Hindex)$
End
```

B) V-RSU Communication

Apparently, the distributed fog nodes cannot cover all areas in the vehicular environment given the limitation of fog node's communication range and the expansion of the transportation network. As a result, a vehicle may occasionally be beyond the fog node's communication range. In this situation, the vehicle node has to interact with the associated RSU, therefore it sends signed messages to the RSU.

In order to verify the signed message, it is required the RSU checks whether the batch authentication equation is established. If Equation 3 holds, it means the message's batch checking has been successfully passed; otherwise, it indicates that there exists at least one invalid message in the batch. The RSU employs the binary search algorithm to identify invalid and faulty messages in the batch.

C) RSU-V / FN-V Communication

[Message signing by RSU]: To guarantee that communication is secure, each event message should be signed by RSU/FN and then broadcast it to the nearby vehicle nodes. To this end, the $RSU_i \in \mathfrak{R}_{rsu}$ signs $M_i = RID_{rsu_i}||m_i||t_i$ with private key s_{rsu}. As discussed earlier, privacy is not a concern and requirement for the RSUs. Therefore, the RSU/FN real identity (RID_{rsu_i}) will be used to sign the message. The corresponding signature on M_i is $\tau'_{RSU_i} = s_{rsu}.h\left(M_i\right)$ and the RSU broadcasts $\left\{RID_{RSU_i}, M_i, \tau'_{RSU_i}\right\}$ to vehicles and the relevant FNs.

[Message verification by Vehicle]: Whenever a vehicle node receives the signed message from the RSU, it has to verify the signature of the message to ensure that the RSU is not attempting to impersonate any other legitimate RSUs or disseminate false messages. To this end, when a vehicle $\left(V_j\right)$ receives a signed message $\left\{RID_{RSU_i}, M_i, \tau'_{RSU_i}\right\}$, after checking the freshness of t_i, it verifies whether

$$\tau'_{RSU_i}.P = PUB_{rsu_i}.h\left(RID_{RSU_i}||m_i||t_i\right) \tag{4}$$

Equation 4 holds or not. If it is established, the message will be accepted by vehicle; otherwise, it ignores the message and marked the RSU as an intruder and broadcast an alert to authorized RSU in its communication range.

D) V-V Communication

When the vehicle node V_l receives a signed event message from vehicle V_k, it must first verify V_k authenticity as explained in Section 5.3.1). If V_k is valid, it checks the integrity of the message $M_k = PID_k||m_k||t_k$. In a V-V communication, V_l makes the request $Req = \langle Req_{id}, PID_l, M_k, PID_k\rangle$ to the relevant fog node and waits for a response to check the message's integrity. If no fog nodes are within its communication range, it forwards the request to the associated RSU, as previously stated.

Upon receiving the vehicle node's request by fog node FN_j, by performing a query, it checks whether $\langle M_k, PID_k\rangle$ exists in *vML*. If so, to verify the message M_k, FN_j sends a reply $Rep(verified)$ to the V_l. Otherwise, FN_j performs a query on *ivML*. If the query returns true, it means the message exists in *ivML* and then FN_j replies $Rep(ignored)$ to V_l. If both queries return false, it means that the message does not exist in either *vML* and *ivML*. It indicates that FN_j has not received the message M_k from V_k and or the fog node FN_j has received M_k after the V_l's request. In this situation, fog node FN_j needs to wait for a certain time. If FN_j received the message within this time, it needs to evaluate the message's authentication by using the single message verification and replies the output to the V_l. Otherwise, it sends $Rep(ignored)$ to V_l.

Once receiving the reply from FN_j, V_l should verify/ignore the message depending on the reply; otherwise, if V_l has not received a response within a specific amount of time, the message will be discarded. In this V-V communication, Algorithm 4 shows the pseudo-code for message verification.

Algorithm 4: Pseudo-code for V-to-V Message Authentication

```
$V_k$: Sender
$V_l$: Receiver
$M_k = PID_k||m_k||t_k$ message send by $V_k$
Sending $Req = \langle Req_{id}, PID_l, M_k, PID_k\rangle$ to $FN$ by $V_k$
if $\langle M_k, PID_k\rangle$ is in vML then
    Send $Rep(verified)$ to $V_l$
if $\langle M_k, PID_k\rangle$ is in ivML then
    Send $Rep(ignored)$ to $V_l$
if $\langle M_k, PID_k\rangle$ is not in vML and ivML then
    $FN$ waits for a certain time i.e. $CT$
    if $FN$ receive message from $V_k$ within $CT$ then
        Check authentication of message using single message verification
        Send result to $V_l$
    else
        Send $Rep(ignored)$ to $V_l$
End
```

The pseudo-identity generation process, signature generation and signature verification between Vehicle and FN (V-FN) and between RSU and Vehicle/FN (RSU-V/FN) can also be found in Figure 3.

6. Security Analysis and Verification

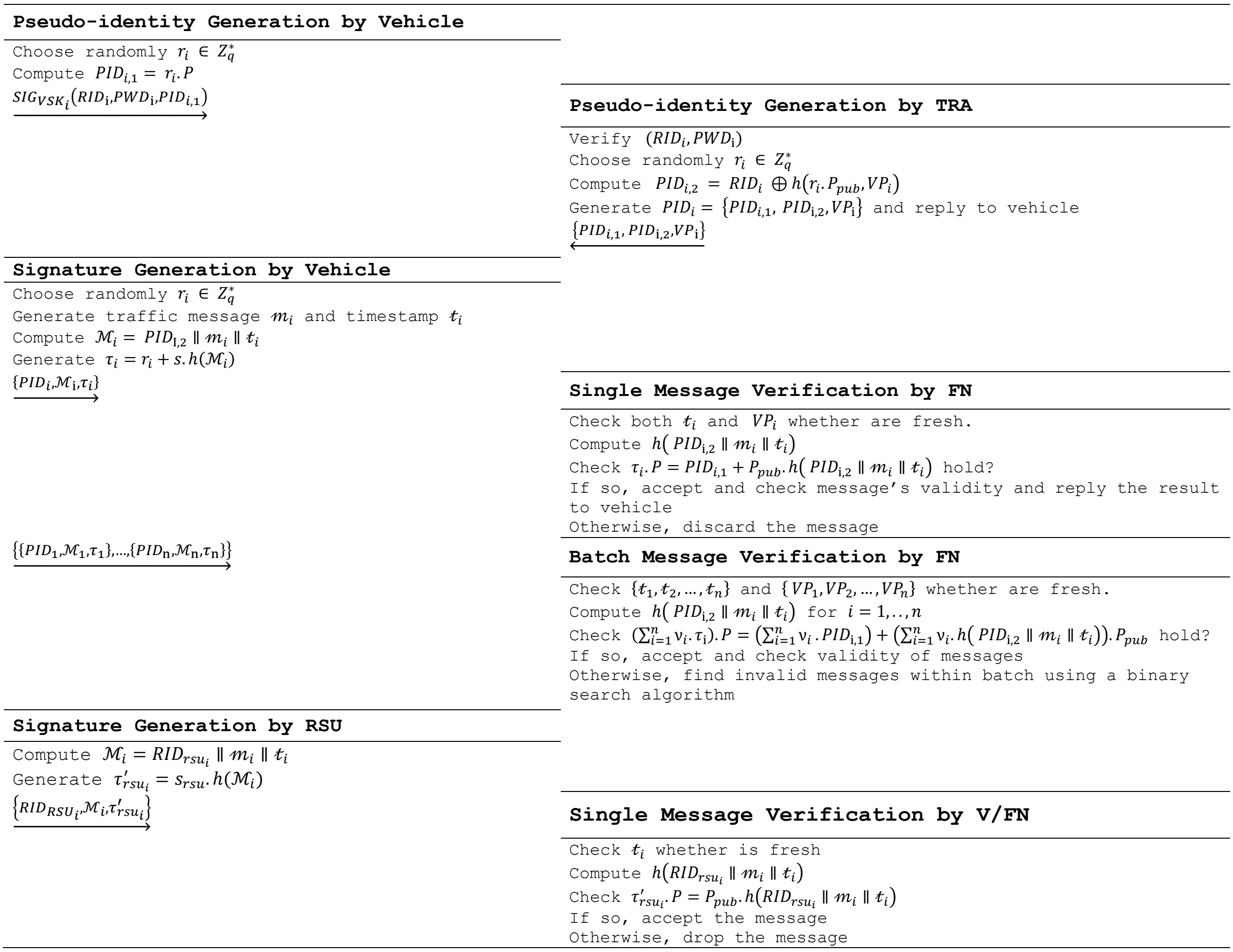
In this section, we prove that our scheme meets the security requirements mentioned in subsection 4.2 and resists attacks. We also test the safety of our scheme against a passive/active adversary, such as replay and man-in-the-middle attacks. To this end, Automated Validation of Internet Security Protocols and Applications (AVISPA) is a popular tool.

6.1. Security Proof

Firstly, we give a proof that our scheme is secure with the random oracle model. This is because proof in the random oracle model ensures the security of the overall design of a signature scheme [31]. To this purpose, Theorem 1 gives a formal proof of the proposed signature scheme against an alternatively chosen message attack using a game between challenger and an adversary as follows:

Theorem 1: *In the random oracle model, our scheme is secure, since it is existentially unforgeable against alternatively and adaptively chosen message attacks under the Discrete Logarithm Problem (DLP) assumption.*

Proof: Let in our system, the security model is established by a challenger $\mathcal{CH}$ and an adversary $\mathcal{ADV}$, in which $\mathcal{ADV}$ can forge $\{PID_i, M_i, \tau_i\}$. Consider a



Pseudo-identity Generation by Vehicle

Choose randomly $r_i \in Z_q^*$
Compute $PID_{i,1} = r_i.P$
$\xrightarrow{SIG_{VSK_i}(RID_\mathrm{i}, PWD_\mathrm{i}, PID_{i,1})}$

Pseudo-identity Generation by TRA

Verify (RID_i, PWD_i)
Choose randomly $r_i \in Z_q^*$
Compute $PID_{i,2} = RID_i \oplus h(r_i.P_{pub}, VP_i)$
Generate $PID_i = \{PID_{i,1}, PID_{\mathrm{i},2}, VP_\mathrm{i}\}$ and reply to vehicle
$\xleftarrow{\{PID_{i,1}, PID_{\mathrm{i},2}, VP_\mathrm{i}\}}$

Signature Generation by Vehicle

Choose randomly $r_i \in Z_q^*$
Generate traffic message m_i and timestamp t_i
Compute $\mathcal{M}_i = PID_{\mathrm{I},2} \parallel m_i \parallel t_i$
Generate $\tau_i = r_i + s.h(\mathcal{M}_i)$
$\xrightarrow{\{PID_i, \mathcal{M}_\mathrm{i}, \tau_i\}}$

Single Message Verification by FN

Check both t_i and VP_i whether are fresh.
Compute $h(PID_{\mathrm{i},2} \parallel m_i \parallel t_i)$
Check $\tau_i.P = PID_{i,1} + P_{pub}.h(PID_{\mathrm{i},2} \parallel m_i \parallel t_i)$ hold?
If so, accept and check message's validity and reply the result to vehicle
Otherwise, discard the message

$\xrightarrow{\{\{PID_1, \mathcal{M}_1, \tau_1\}, \ldots, \{PID_\mathrm{n}, \mathcal{M}_\mathrm{n}, \tau_\mathrm{n}\}\}}$

Batch Message Verification by FN

Check $\{t_1, t_2, \ldots, t_n\}$ and $\{VP_1, VP_2, \ldots, VP_n\}$ whether are fresh.
Compute $h(PID_{\mathrm{i},2} \parallel m_i \parallel t_i)$ for $i = 1, .., n$
Check $(\sum_{i=1}^{n} \nu_i.\tau_\mathrm{i}).P = (\sum_{i=1}^{n} \nu_i.PID_{\mathrm{i},1}) + (\sum_{i=1}^{n} \nu_i.h(PID_{\mathrm{i},2} \parallel m_i \parallel t_i)).P_{pub}$ hold?
If so, accept and check validity of messages
Otherwise, find invalid messages within batch using a binary search algorithm

Signature Generation by RSU

Compute $\mathcal{M}_i = RID_{rsu_i} \parallel m_i \parallel t_i$
Generate $\tau'_{rsu_i} = s_{rsu}.h(\mathcal{M}_i)$
$\xrightarrow{\{RID_{RSU_i}, \mathcal{M}_i, \tau'_{rsu_i}\}}$

Single Message Verification by V/FN

Check t_i whether is fresh
Compute $h(RID_{rsu_i} \parallel m_i \parallel t_i)$
Check $\tau'_{rsu_i}.P = P_{pub}.h(RID_{rsu_i} \parallel m_i \parallel t_i)$
If so, accept the message
Otherwise, drop the message

Figure 3: Pseudo-identity generation and authentication processes of our scheme.

game between $\mathcal{CH}$ and $\mathcal{ADV}$, which can solve the DLP by running $\mathcal{ADV}$ with a non-negligible probability. To that end, it is supposed that $\mathcal{CH}$ maintains three hash lists $List_{H_1}$, $List_{H_2}$ and $List_{H_3}$ which are initialized to empty.

Definition 1: Let G be an additive elliptic curve group of order q and $P, Q \in G$ as two random numbers on E where $Q = x.P$. Based on the DLP, it is not easy to compute x from Q.

Setup: $\mathcal{CH}$ chooses randomly the number s as the private key and compute the public key using $P_{pub} = s.P$. Then, $\mathcal{CH}$ sends the generated system parameters $params = \left\{p, q, P, Q, P_{pub}, H_1, H_2, H_3\right\}$ to $\mathcal{ADV}$.

H_1**-Oracle**: $\mathcal{CH}$ keeps a list $\left(List_{H_1}\right)$ with the form of $\langle m, \tau\rangle$. When $\mathcal{ADV}$ creates a H_1 query with message m, $\mathcal{CH}$ checks whether the tuple $\langle m, \tau\rangle$ is already in the $List_{H_1}$ or not. If so, $\mathcal{CH}$ sends $\tau = H_1(m)$ to $\mathcal{ADV}$; if not, $\mathcal{CH}$ selects a random $\tau \in Z_q^*$ and adds $\langle m, \tau\rangle$ into the $List_{H_1}$. Finally, $\mathcal{CH}$ sends $\tau = H_1(m)$ to $\mathcal{ADV}$.

H_2**-Oracle**: $\mathcal{CH}$ keeps a list $\left(List_{H_2}\right)$ with the form of $\langle PID_i, m, \tau\rangle$ When $\mathcal{ADV}$ creates a H_2 query with the message $\langle PID_i, m, \tau\rangle$, $\mathcal{CH}$ exams whether the tuple $\langle PID_i, m, \tau\rangle$ is already in the $List_{H_2}$ or not. If so, $\mathcal{CH}$ sends $\tau = H_2(PID_i||m)$ to $\mathcal{ADV}$. Otherwise, $\mathcal{CH}$ selects a random $\tau \in Z_q^*$ and then adds $\{PID_i, m, \tau\}$ into the $List_{H_2}$. In the end, $\mathcal{CH}$ sends $\tau = H_2(PID_i||m)$ to $\mathcal{ADV}$.

H_3**-Oracle**: $\mathcal{CH}$ keeps a list $\left(List_{H_3}\right)$ with the form of $\langle PID_i, M_i, \tau\rangle$ in which $M_i = m_i||t$. Once $\mathcal{CH}$ receives a query of $\mathcal{ADV}$ creates with the message $\langle PID_i, M_i, \tau\rangle$, it checks whether the tuple $\langle PID_i, M_i, \tau\rangle$ is already in the $List_{H_3}$ or not. If so, $\mathcal{CH}$ sends $\tau = H_3(PID_i||M_i)$ to $\mathcal{ADV}$. Otherwise, $\mathcal{CH}$ selects a random $\tau \in Z_q^*$ and then adds $\{PID_i, M_i, \tau\}$ into the $List_{H_3}$. In the end, $\mathcal{CH}$ sends $\tau = H_3(PID_i||M_i)$ to $\mathcal{ADV}$.

Sign-Oracle: Upon receive a query of $\mathcal{ADV}$ with the message m, $\mathcal{CH}$ generates three random numbers $\alpha_i, \beta_i, \tau_i \in Z_q^*$ and chooses a random point $PID_{i,2}$ and computes $PID_{i,1} = \tau_i.P - P_{pub}.h(PID_{i,2}||m_i||t_i)$. Then, $\mathcal{CH}$ adds $\langle PID_i, m_i, \alpha_i\rangle$ and $\langle PID_i, M_i, \beta_i\rangle$, respectively, into the $List_{H_2}$ and $List_{H_3}$ in which $PID_i = \{PID_{i,1}, PID_{i,2}\}$. Next, $\mathcal{CH}$ sends $\langle PID_i, M_i, \tau_i\rangle$ to

$\mathcal{ADV}$. It is easy to verify the equation $\tau_i.P = PID_{i,1} + P_{pub}.h\left(PID_{i,2}||m_i||t_i\right)$ holds. Therefore, all signatures generated by $\mathcal{CH}$ are indistinguishable from those generated by legal vehicles. Finally, $\mathcal{ADV}$ outputs a message $\langle PID_i, M_i, \tau_i\rangle$ and $\mathcal{CH}$ checks whether $\tau_i.P = PID_{i,1} + P_{pub}.h\left(PID_{i,2}||m_i||t_i\right)$ is established or not. If no, $\mathcal{CH}$ aborts the process.

By using Forking Lemma [31], $\mathcal{ADV}$ produces another valid message $\left\langle PID_i, M_i, \tau_i^{'}\right\rangle$. In the valid messages $\langle PID_i, M_i, \tau_i\rangle$ and $\left\langle PID_i, M_i, \tau_i^{'}\right\rangle$, the signatures $\tau_i = r_i + s.h\left(M_i\right)$ and $\tau_i^{'} = r_i + s.h^{'}\left(M_i\right)$ where $h \neq h^{'}$ are produced by $\mathcal{CH}$ within polynomial running time. Given these signatures, $\mathcal{CH}$ achieves the value of $x = \left(\tau_i - \tau_i^{'}/h - h^{'}\right) \quad mod \quad q$ as the answer of the DLP. To prove this, with substituting $r_i = \tau_i - s.h$ in the $r_i = \tau_i^{'} - s.h^{'}$, it gives the following result:

$$\tau_i - s.h = \tau_i^{'} - s.h^{'} \Rightarrow \tau_i - \tau_i^{'} = s\left(h - h^{'}\right) \Rightarrow$$
$$s = \left(\tau_i - \tau_i^{'}/h - h^{'}\right) mod \;\; q$$

This contradicts the hardness of the DLP. As explained in [32], a scheme is secure if the DL problem is hard. In addition, a scheme is secure if an existential forgery is computationally impossible, even under an adaptively chosen-message attack [33]. Therefore, our scheme is secure against forgery under adaptive chosen message attacks in the random oracle model and provides message authentication for VANETs.

Theorem 2: *(Message Verification and Integrity) the message's integrity is ensured by the signature of the message.*

Proof: Proof in the random oracle model ensures the security of the signature scheme. As discussed in Theorem 1, our proposed signature is secure against an alternatively chosen message attack under the random oracle model, and as a result, a malicious attacker cannot forge valid signatures.

Theorem 3: *(Resistance to Unauthorized Nodes) it guarantees an unauthorized and fake node cannot enter the network and initiating data sharing with authorized nodes.*

Proof: It is supposed that the pseudo-identity of legitimate and illegitimate nodes is stored in a filter QF_V on each vehicle. When a vehicle node received a communication request from another vehicle node, it assesses the validity and legitimacy of the sender by executing a query on its own filter before beginning any communication. If the query returns $FALSE$, the vehicle node forwards the request instantly to the corresponding FN. If the query on the filter QF_{FN} returns $FALSE$, it indicates that the vehicle node is an unregistered entity and hence the vehicle node is marked as an unauthorized node. In contrast, when the query returns $TRUE$, it means the vehicle has been detected as an unauthorized node by FN. Consequently, an unauthorized vehicle will be unable to connect to the network and begin communication with other vehicles or fog nodes.

Theorem 4: *(Identity Privacy-Preserving) it is impossible for an adversary to extract the real identity of the authorized vehicle node from its pseudonym.*

Proof: The vehicle V_i transmits message $\{PID_i, M_i, \tau_i\}$ to other nodes, where $PID_i = \{PID_{i,1}, PID_{i,2}, VP_i\}$, $PID_{i,1} = r_i.P$, and $PID_{i,2} = RID_i \oplus h\left(r_i.P_{pub}, VP_i\right)$. The real identity RID_i of the vehicle is perfectly concealed since PID_i is an unknown identity with a random number r_i. Based on the DLP, it is hard to compute the private key r_i of the vehicle through $PID_{i,1}$ and P. Hence, the adversary is unable to extract RID_i and as a result, the proposed scheme satisfies privacy-preserving. Furthermore, in our scheme, a vehicle node changes pseudo-identity PID_i after a valid period of time VP_i. This frequent change is mainly because when a vehicle uses a pseudo-identity constantly within the vehicular communication, the vehicle movement trajectory can be traced by an adversary [17]. We prove that the relation between the pseudo-identities can be revealed only by TRA. To this end, consider two pseudo-IDs $PID_{i,2}$ and $PID_{i+1,2}$ related to the vehicle node RID_i where $PID_{i,2} = RID_i \oplus h\left(r_i.P_{pub}, VP_i\right)$ and $PID_{i+1,2} = RID_i \oplus h\left(r_{i+1}.P_{pub}, VP_{i+1}\right)$. Assuming that attacker knows P_{pub}, VP_i, and VP_{i+1}. To verify relation of $PID_{i,2}$ and $PID_{i+1,2}$ with RID_i, the attacker should computes both $h^{-1}\left(r_i.P_{pub}, VP_i\right)$ and $h^{-1}\left(r_{i+1}.P_{pub}, VP_{i+1}\right)$. These computations are performed until the relation verification is confirmed. As described in [5], for a n-bit one-way hash function, the complexity of solving h^{-1} is $O(2^{n-1})$. Suppose $PID_{i,2}$ an d $PID_{i+1,2}$ belong to RID_i, hence for each $h^{-1}\left(r_i.P_{pub}, VP_i\right)$, 2^{n-1} times of $h^{-1}\left(r_{i+1}.P_{pub}, VP_{i+1}\right)$ operation needs to confirm. So, the total complexity is $O(2^{2n-2})$. Since the hash function used in our scheme is a SHA-256, hence the relationship verification between two pseudo identities is not easy computational problem.

Theorem 5: *(Resistance to Replay Attack) an adversary is unable to broadcast the received signed message if it is expired.*

Proof: The message's signature consists of a timestamp that can withstand and resist replay attacks. In our work, the timestamp t_i is concatenated to the message the message m_i and time synchronization is maintained in all vehicles. For all communicating entities, the current timestamp is employed and the maximum transmission delay in each exchanged message is usually a small amount. Hence, if an adversary replays the intercepted message, It is easily recognizable and detectable. Let $\mathcal{ADV}$ be an adversary that intercepts the message $\{PID_i, M_i, \tau_i\}$ where $M_i = m_i||t_i$ and $\mathcal{ADV}$ launches a replay attack at t_j. Given $t_j - t_i > \Delta t$, message will be rejected by receiver in which Δt is a conjointly agreed to transmission delay. As a result, our scheme prevents a replay attack.

Theorem 6: *(Traceability) it is possible only for TRA to track the vehicle's real-identity from its pseudonym.*

Definition 2: *It is possible to encrypt the string of text by employing the XOR operation ($\oplus$) to every character utilizing a given key. For decryption the output, the cipher will be removed only by reapplying the XOR function with the key as:*

$$If \quad \mathbb{X} \oplus \mathbb{Y} = \mathbb{Z} \quad then \quad \mathbb{X} \oplus \mathbb{Z} = \mathbb{Y}$$

Proof: Consider $PID_i = \{PID_{i,1}, PID_{i,2}, VP_i\}$ as pseudo-identity and $PID_{i,2} = RID_i \oplus h\left(r_i.P_{pub}, VP_i\right)$. TRA can trace the vehicle's real identity using Definition 2:

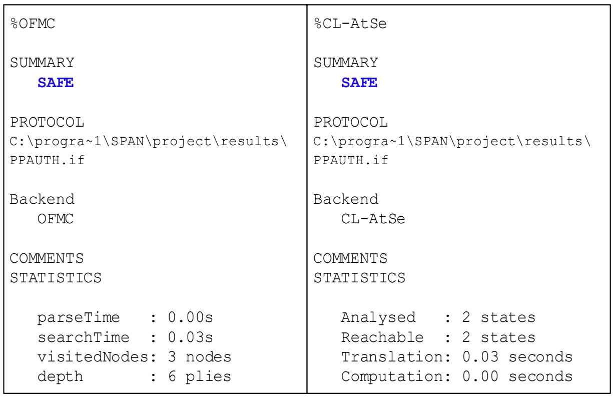
$$RID_i = PID_{i,2} \oplus h\left(r_i.P_{pub}, VP_i\right)$$

It is impossible for other involved entities since only TRA knows about $PID_{i,2}$.

6.2. Security Verification

Recently, AVISPA becomes a popular tool to test whether a security protocol is safe against a passive and or active adversary, such as replay and man-in-the-middle attacks. AVISPA is integrated with a Security Protocol ANimator (SPAN). It is a graphical user interface that uses to verify the security of cryptographic protocols. AVISPA also provides a High-Level Protocol Specification Language (HLPSL) that consists of four sections, role, session, environment, and goal. In order to evaluate a security protocol on the AVISPA, it needs to firstly execute the protocol in HLPSL specification. Then, HLPSL2IF, a built-in translator, converts the HLPSL specification into the Intermediate Format (IF). Finally, in order to check whether the security protocol is safe against the active or passive attack, the IF specification will be evaluated by using the back-ends integrated into AVISPA. The On-the-Fly Model-Checker (OFMC), Constraint Logic-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC), and Tree Automata tool based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) are widely-used back-ends in AVISPA.

Here, we have modelled our scheme using the AVISPA. To this end, we implemented the role specifications of the vehicle, RSU, and FN in HLPSL. Then, we have simulated the proposed scheme using the SPAN. The simulation results under both OFMC and CL-AtSe are reported in Figure 4. The obtained results assure that our scheme is safe against replay and man-in-the-middle attacks.



```
%OFMC                                   %CL-AtSe

SUMMARY                                 SUMMARY
   SAFE                                    SAFE

PROTOCOL                                PROTOCOL
C:\progra~1\SPAN\project\results\       C:\progra~1\SPAN\project\results\
PPAUTH.if                               PPAUTH.if

Backend                                 Backend
   OFMC                                    CL-AtSe

COMMENTS                                COMMENTS
STATISTICS                              STATISTICS

   parseTime   : 0.00s                     Analysed   : 2 states
   searchTime  : 0.03s                     Reachable  : 2 states
   visitedNodes: 3 nodes                   Translation: 0.03 seconds
   depth       : 6 plies                   Computation: 0.00 seconds
```

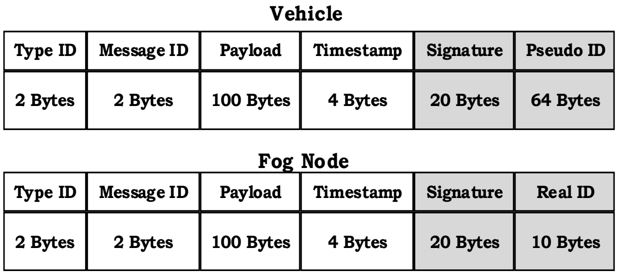
Figure 4: The results of OFMC and CL-AtSe back-ends.

OFMC is an extremely effective, state-of-the-art protocol analysis tool both in terms of coverage and performance. It is able to re-discover known attacks as well as find new attacks. OFMC integrates a number of symbolic techniques and optimisations, which are correct and complete, in the sense that no attacks are lost nor new ones are introduced. The CL-AtSe tool has proved to be extremely efficient on protocol analysis, especially when the associativity of the concatenation is not required. Moreover, CL-AtSe is able to perform verification and validation of security protocols modulo various algebraic properties.

7. Performance Evaluation

The bilinear pairing and ECC are two basic and well-known methods that are usually used for the proposed authentication schemes. In this study, we show a comparison of works that are established on these two methods. To this end, a comparison is made between our scheme and CPAS [21], PPAS [22], AAAS [34], CL-CPPA [25], and EMAS [26] in terms of both communication and computation cost. The first three schemes are based on the bilinear pairing method, whereas CL-CPPA, EMAS, and our scheme are established on ECC.

We have used OMNET++ to assess the performance of the scheme. The simulation area is 5 km × 5 km with a maximum node density of 500 nodes. In this simulation area, 5 RSUs and 15 fog nodes are considered to serve the vehicle nodes. In order to take the benefits and the advantages of the fog computing-based VANET, RSUs and fog nodes are deployed at proper distances to give sufficient coverage where each RUS can handle 500 requests at once. The two-ray ground



Vehicle

Type ID	Message ID	Payload	Timestamp	Signature	Pseudo ID
2 Bytes	2 Bytes	100 Bytes	4 Bytes	20 Bytes	64 Bytes

Fog Node

Type ID	Message ID	Payload	Timestamp	Signature	Real ID
2 Bytes	2 Bytes	100 Bytes	4 Bytes	20 Bytes	10 Bytes

Figure 5: The signed message format of vehicle and fog node.

reflection model is used as the radio propagation model to simulate the wireless channel. IEEE 802.11p is utilized in the MAC-layer. Moreover, the transmission range of the vehicle nodes has been set at 300 meters and the channel bandwidth is 6 Mbps. In each simulation run, the total simulation time is 360 seconds and at the start of the simulation, the setup time is set to 30 seconds to eliminate the influence of transient performance on the findings. Besides, the sending packets will be stopped in the last 30 seconds of the simulation. For the sake of simplicity, we assume that both vehicles and fog nodes have the same equipment and the experiment is executed in a machine equipped with a 3.4GHz i7-2600 CPU.

7.1. Communication Overhead

The cost of communication is an important factor to consider when evaluating the scheme's performance. In the proposed scheme, vehicles and fog nodes must sign messages before transmission. Although this ensures the message integrity and authenticity of the sender, however, it increases the communication cost. As shown in Figure 5, in order to analyze the communication overhead, a generic event message format is specified in [21]. In the adopted format, the signature is taken into account as cryptography and communication overhead. Obviously, the size of the signature must be reduced in order to reduce transmission and communication cost. As described in [21], it is suitable to use a 159-bit subgroup of the MNT curve with an embedding degree of 6 to reduce the signature length.

The total packet size may be reduced by 192 bytes using our scheme, where the signature is 20 bytes and pseudo-identity is 64 bytes.

As mentioned in [35], the element's size in group G such $\{PID \in G\}$, timestamp $\{VP\}$, the hash function's output, such as $\left\{\tau \in Z_q^*\right\}$, and real-identity $\{RID\}$ are respectively 40 bytes, 4 bytes, 20 bytes, and 10 bytes. So, given $\{PID_V, M_V, \tau_V\}$, our scheme's overall signature size excluding message and pseudo-identity is 20 bytes and the size of pseudo-identity $\{PID_{V,1}, PID_{V,2}, VP_V\}$ is 64 bytes. Besides, our system sends messages from RSU/FN to vehicle by using the real-identity rather than a pseudo-identity. As a result of the size of the message, Type-ID, Message-ID, signature, and pseudo-identity, the total packet size in our scheme from vehicle to RSU/FN is 192 bytes, whereas from fog node to vehicle is 138 bytes.

Due to the size of each element, the signature size for CPAS is 20+20+20 = 60 bytes. And, it is 20+20 = 40 bytes for PPAS, 40+40 = 80 bytes for AAAS, 40+20 = 60 bytes for CL-CPPA, whereas the signature's size of EMAS is 20 bytes. The pseudo-ID size of CPAS, PPAS, and our scheme is 40+20+4 = 64 bytes. It is 40+40 = 80 bytes for CL-CPPA, whereas, size of pseudo-ID of EMAS is 40+20 = 60 bytes and it is 20+20+4 =44 bytes for AAAS.

The communication cost of the proposed scheme and other work are presented in Table 2 in which EMAS has the lowest cost of communication. The reason is other schemes use timestamp as an element in pseudo-ID generation which increases the communication cost by 4 bytes. In EMAS and CL-CPPA, each vehicle uses only one pseudo-ID when communicating with other entities during movement; whereas the proposed scheme, CPAS, AAAS, and PPAS change the pseudo-ID of the vehicle nodes over a period of time. According to [17], an adversary can trace the vehicle movement trajectory, if the vehicle uses one pseudo-identity during all communication. As a result, EMAS and CL-CPPA cannot meet privacy-preserving requirements.

7.2. Computation Overhead

Here, we compare our scheme, CPAS, PPAS, AAAS, CL-CPPA, and EMAS in terms of computation overhead. To this end, by inspiring the computation evaluation method for VANET in [29], the ECC and bilinear pairing on the security level of 80 bits are created as follows: The ECC is constructed using an additive group G generated by a point P on a non-singular elliptic curve $E : y^2 = x^3 + ax + b$ $(mod \;\; p)$, and its order is q, where $a, b \in Z_p^*$, and p,q are two 160-bit prime numbers; whereas the bilinear pairing is created using $\bar{e} : G_1 \times G_1 \to G_T$, where G_1 is an additive group which is generated by a point $\bar{P}$ with the order $\bar{q}$ on the super singular elliptic curve $\bar{E} : y^2 = x^3 + ax + b \; mod \; \bar{p}$ with embedding degree 2, especially $\bar{p}$ consisting of a 512-bit prime number, $\bar{q}$ consisting of a 160-bit Solinas prime number.

In this work, we compute the execution time of cryptographic operations by executing the benchmark on Intel Core i5-6300U processor with CPU speed 2.4 GHz and 8G Memory as well as on Linux host using Intel Core i7-2600 processor with CPU speed 3.4 GHz and 8G Memory as testbeds. To this end, we use the Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) [36]. Table 3 represents the computation time of cryptographic operations utilized in our scheme and other comparable schemes.

Here, we compute the time it takes to generate a pseudo-identity, computation time of single message

Table 2
Comparison of Communication Cost

Model	Type ID	Msg. ID	Payload	Timestamp	Signature	Pseudo-ID	Total
CPAS	2	2	100	4	60	64	232 Bytes
PPAS	2	2	100	4	40	64	212 Bytes
AAAS	2	2	100	4	80	44	232 Bytes
CL-CPPA	2	2	100	4	60	80	248 Bytes
EMAS	2	2	100	4	20	60	188 Bytes
Our Scheme	2	2	100	4	20	64	192 Bytes

Table 3
Computation Time of Cryptographic Operations

Hardware Specification		Intel Core i7-2600	Intel Core i5-6300U
Symbol	Operation	Execution time (ms)	
T_{bp}	Bilinear pairing	4.2110	5.1840
$T_{bp.m}$	Scalar multiplication associated with BP	1.7090	1.9285
$T_{bp.sm}$	Small scalar multiplication associated with BP	0.0535	0.0820
$T_{bp.a}$	Point addition associated with BP	0.0071	0.0120
T_{mtp}	Map-To-Point hash associated with BP	4.4060	5.0270
$T_{e.m}$	Scale multiplication associated with ECC	0.4420	0.4835
$T_{e.sm}$	Small scalar multiplication associated with ECC	0.0138	0.0155
$T_{e.a}$	Point addition	0.0018	0.0056
T_h	One-way hash function	0.0001	0.0001

verification, message signing, and batch message authentication for our scheme and related works, separately.

[To pseudo-identity generation]: to this end, two scalar multiplication procedures and one one-way hash function operation make up our scheme. As a result, the total computing time is $2T_{e.m} + T_h \cong 2 * 0.4420 + 0.0001 = 0.8841(ms)$. For PPAS, it includes two scalar multiplication and one one-way hash function. Therefore, the overall computation time is $2T_{e.m} + T_h \cong 2 * 0.4420 + 0.0001 = 0.8841(ms)$. For CPAS, this includes three scalar multiplication processes and one one-way hash function. As a result, the total computation time for all process is $3T_{e.m} + T_h \cong 3 * 0.4420 + 0.0001 = 1.3261(ms)$. For AAAS, the pseudo-identity generation consists one scalar multiplication and one one-way hash function. So, the overall computation time is $T_{e.m} + T_h \cong 0.4420 + 0.0001 = 0.4421(ms)$. CL-CPPA consists one scalar multiplication and two point addition operations. So, the overall computation time is $(T_{e.m} + 2T_{e.a}) \times z \cong 0.4420 + 2 * 0.0018 = 0.4456(ms)$. And, for EMAS, pseudo-identity generation includes one one-way hash function and two scalar multiplication. Therefore, the computation time for whole procedure is $2T_{e.m} + T_h \cong 2 * 0.4420 + 0.0001 = 0.8841(ms)$.

[To message signing]: our scheme consists of one one-way hash function, and two scalar multiplication. Hence, the total computation time is $2T_{e.m} + T_h \cong 2 * 0.4420 + 0.0001 = 0.8841(ms)$. Whereas, PPAS includes tree scalar multiplication, one map-to-point hash function, and two one-way hash function. Therefore, the overall computation time for the entire procedure is $3T_{e.m} + T_{mtp} + 2T_h \cong 3 * 0.4420 + 4.4060 + 2 * 0.0001 = 5.7302(ms)$. CPAS signs a message with five scalar multiplication, and two one-way hash function. Consequently, the overall computation time is $5T_{e.m} + 2T_h \cong 5 * 0.4420 + 2 * 0.0001 = 2.2102(ms)$. For AAAS, it consists one map-to-point hash function, two scalar multiplication and one point multiplication. Hence, the overall computation time is $T_{mtp} + 2T_{e.m} + T_{bp.m} \cong 4.4060 + 2 * 0.4420 + 1.7090 = 6.9990(ms)$. For CL-CPPA, it includes three scalar multiplication, two point addition operations and only one one-way hash function. So, the overall computation time for message signing is $3T_{e.m} + 2T_{e.a} + 1T_h \cong 3 * 0.4420 + 2 * 0.0018 + 1 * 0.0001 = 1.3297(ms)$. And, EMAS includes four scalar multiplication and two one-way hash function. So, the overall computation time is $4T_{e.m} + 2T_h \cong 4 * 0.4420 + 2 * 0.0001 = 1.7682(ms)$.

[To single message verification]: our scheme involves one one-way hash function, and three scalar multiplication. Hence, the overall computation time is $3T_{e.m} + T_h \cong 3 * 0.4420 + 0.0001 = 1.3261(ms)$. PPAS comprises two bilinear pairing, three one-way hash function, one map-to-point hash function, and three scalar multiplication. So, the total computation time is $2T_{bp} + 3T_h + T_{mtp} + 3T_{e.m} \cong 2 * 4.2110 + 3 * 0.0001 + 4.4060 + 3 * 0.4420 = 14.1543(ms)$. CPAS includes two bilinear pairing operation, two one-way hash function, three scalar multiplication. Thus, the overall computation time is $2T_{bp} + 2T_h + 3T_{e.m} \cong 2 * 4.2110 + 2 * 0.0001 + 3 * 0.4420 = 9.7482(ms)$. For AAAS, it consists three bilinear pairing, one map-to-point hash function, and two scalar multiplication. So, the overall computation time is $3T_{bp} + T_{mtp} + 2T_{e.m} + T_{bp.m} \cong 3 * 4.2110 + 4.4060 + 2 * 0.4420 = 17.9230(ms)$. CL-CPPA includes tree scalar multiplication, three point addition operations and one one-way hash function. So, the overall computation time is $3T_{e.m} + 3T_{e.a} + T_h \cong 3 * 0.4420 + 3 * 0.0018 + 0.0001 = 1.3315(ms)$. And, EMAS comprises two one-way hash function and five scalar multiplication. Therefore, the total computation time is $5T_{e.m} + 2T_h \cong 5 * 0.4420 + 2 * 0.0001 = 2.2102(ms)$.

[To batch message verification]: our scheme is made up of $(n + 1)$ scalar multiplication, and (n) one-way hash function. Therefore, the total computation time is $(n + 1)T_{e.m} + nT_h \cong (n + 1) * 0.4420 + n * 0.0001 = 0.4421n + 0.4420(ms)$. PPAS contains comprises two bilinear pairing, $(n + 1)$ scalar multiplication, $(2n)$ map-to-point hash function, and $(2n + 1)$ one-way hash function. Accordingly, the overall computation time is $2T_{bp} + (2n + 1)T_h + 2nT_{mtp} + (n + 1)T_{e.m} \cong 2 * 4.2110 + (2n + 1) * 0.0001 + 2n * 4.4060 + (n + 1) * 0.4420 = 9.2542n + 8.8641(ms)$. CPAS includes two bilinear pairing, $(n + 1)$ scalar multiplication, and $(3n)$ one-way hash function. Therefore, the overall computation time is $2T_{bp} + 3nT_h + (n + 1)T_{e.m} \cong 2 * 4.2110 + 3n * 0.0001 + (n + 1) * 0.4420 = 9.2541n + 8.8640(ms)$. CL-CPPA comprises $(3n + 2)$ scalar multiplication, $(3n)$ point addition operations and (n) one-way hash function. So, the total computation time is $(3n + 2)T_{e.m} + (3n)T_{e.a} + (n)T_h + \cong (3n + 2) * 0.4420 + (3n) * 0.0018 + (n) * 0.0001 = 1.3315n + 0.8840(ms)$. EMAS contains $(4n + 1)$ scalar multiplication and $(2n)$ one-way hash function. So, the overall computation time is $(2n)T_h + (4n + 1)T_{e.m} \cong (2n) * 0.0001 + (4n + 1) * 0.4protocol420 = 1.7682n + 0.4420(ms)$.

Table 4 represents the computational cost of our scheme and other comparable schemes in pseudo-identity generation, message signing, single message verification, and batch message verification. As we can see in this table, AAAS only support single message verification, whereas other schemes support both single and batch message verification. Due to the big data generated in the vehicular environment, this issue can be a drawback of AAAS. In terms of batch message verification, the computational cost of our scheme, CPAS, PPAS, CL-CPPA and EMAS for 100 messages is 44.6520, 48.8830, 934.2841, 134.0340 and 177.2620 (ms), respectively. It indicates that batch verification in our scheme has an improvement higher than CPAS, PPAS, CL-CPPA, and EMAS. In this phase, the percentage improvement of the total operation time of the proposed scheme is $\frac{48.8830-44.6520}{48.8830} \times 100 \cong 8.65\%$, $\frac{934.2841-44.6520}{934.2841} \times 100 \cong 95.22\%$, $\frac{134.0340-44.6520}{134.0340} \times 100 \cong 66.68\%$, and $\frac{177.2620-44.6520}{177.2620} \times 100 \cong 74.81\%$, approximately.

As mentioned above, the elliptic curve point operations have much less computational cost than the bilinear pairing operations [21, 22]. Table 4 shows that our scheme outperform other works because of using bilinear pairing operation and elliptic curve. To prove this claim, we compared the works using the Monte-Carlo simulation. We computed the time consumption of the message signing and single message verification for each scheme, separately. To calculate the time consumption of a large-scale network, we run 1000 Monte-Carlo simulations. As shown in Figure 6, the average time consumption in our scheme is about 3.23 ms. Therefore, our scheme obtains more efficiency in terms of authentication by eliminating the bilinear pairing operations.

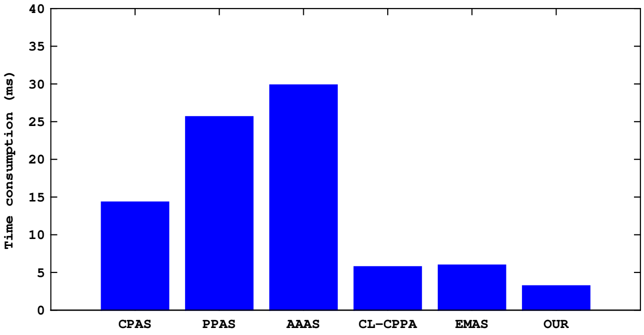


Figure 6: Computational cost for message signing and single message verification in our scheme and other related works.

Moreover, we demonstrated the impact of fog node in the designed framework. Since delay is an important issue in VANET, we compared the network delay of the proposed scheme with other works in four different scenarios: (i) our scheme with only RSU; (ii) our scheme with only Cloud; (ii) our scheme with cloud-edge; (iv) our scheme with cloud-RSU-fog. Figure 7 shows the average network delay is high when we use only RSU or cloud. In contrast, the average network delay while using edge or fog is low. For the last two scenarios, the network delay is almost the same until the number of vehicles in the proposed scheme is below 400, but when the density is increased, the cloud-edge is utilized and more network delay is experienced. This is mainly because of the lower processing and storage capabilities of edge nodes compared to the fog nodes. Hence, the edge node needs to send the data to cloud for more processing.

7.3. Transmission Delay

Here, we used the transmission delay to measure the communication overhead in order to indicate the efficiency of our scheme. The transmission latency of our scheme is compared with other related schemes at different speeds (40 km/h, 70 km/h, 100 km/h and 150 km/h) under different density of vehicle nodes (50, 100, 200, 300, 400, and 500 nodes) when 20% of participated vehicles in the network are malicious and generate invalid signatures (see Figure 8).

We acknowledge that the average transmission delay increases with increasing the number of vehicle node. Also, velocity influences the transmission de-

Table 4
Comparison of Computation Cost

Model	Pseudo-id Generation	Message Signing	Single Verification	Batch Verification
CPAS	1.3261	2.2102	9.74820	$0.4423n + 4.6530$
PPAS	0.8841	5.7302	14.1543	$9.2542n + 8.8641$
AAAS	0.4421	6.9990	17.9230	–
CL-CPPA	0.4456	1.3297	1.33150	$1.3315n + 0.8840$
EMAS	0.8841	1.7682	2.21020	$1.7682n + 0.4420$
Our Scheme	0.8841	0.8841	1.32610	$0.4421n + 0.4420$

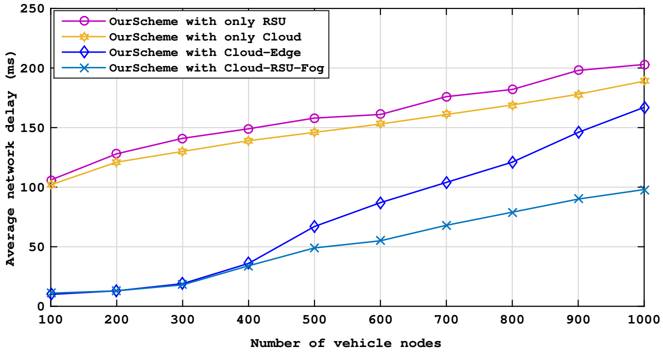


Figure 7: Comparison of network delay of proposed scheme in four different scenarios.

lay. In fact, the transmission delay increases with more number of malicious nodes. To prove this, we measured transmission delay when 50% of vehicle nodes in the network are malicious. We observed that when the speed of vehicle nodes is 100 (km/h), with the number of malicious nodes increasing from 20% to 50% in the network, the transmission delay of our scheme, CPAS, PPAS, AAAS, CL-CPPA and EMAS respectively increase nearly 25%, 42%, 37%, 39%, 36% and 38%.

7.4. Batch Message Verification Analysis

As mentioned earlier, in the batch message verification, when there is at least one invalid message in the batch, it needs to find the invalid message(s). To this end, we proposed a recursive algorithm based on the binary search. In this algorithm, the desired batch will be broken into two separate batches, first. This segmentation will be continued until finding all invalid message(s). Figure 9 shows the segmentation by this algorithm when the desired batch contained 10 event messages.

As shown in this figure, the batch message verification will be performed on the initial batch contained 10 messages, first. If Equation (3) is not established, there exist at least one message in the batch that is invalid. Therefore, the initial batch divides into two separate batches contained 5 messages. The batch message verification performs on these two batches, separately. If each batch holds Equation (3), it means all messages exist in the batch are valid and hence the algorithm will be stopped for this batch. Otherwise, the batch segmentation will be continued until each batch contained two or one messages. In this situation, the proposed single message verification will be performed to check the validity of the message.

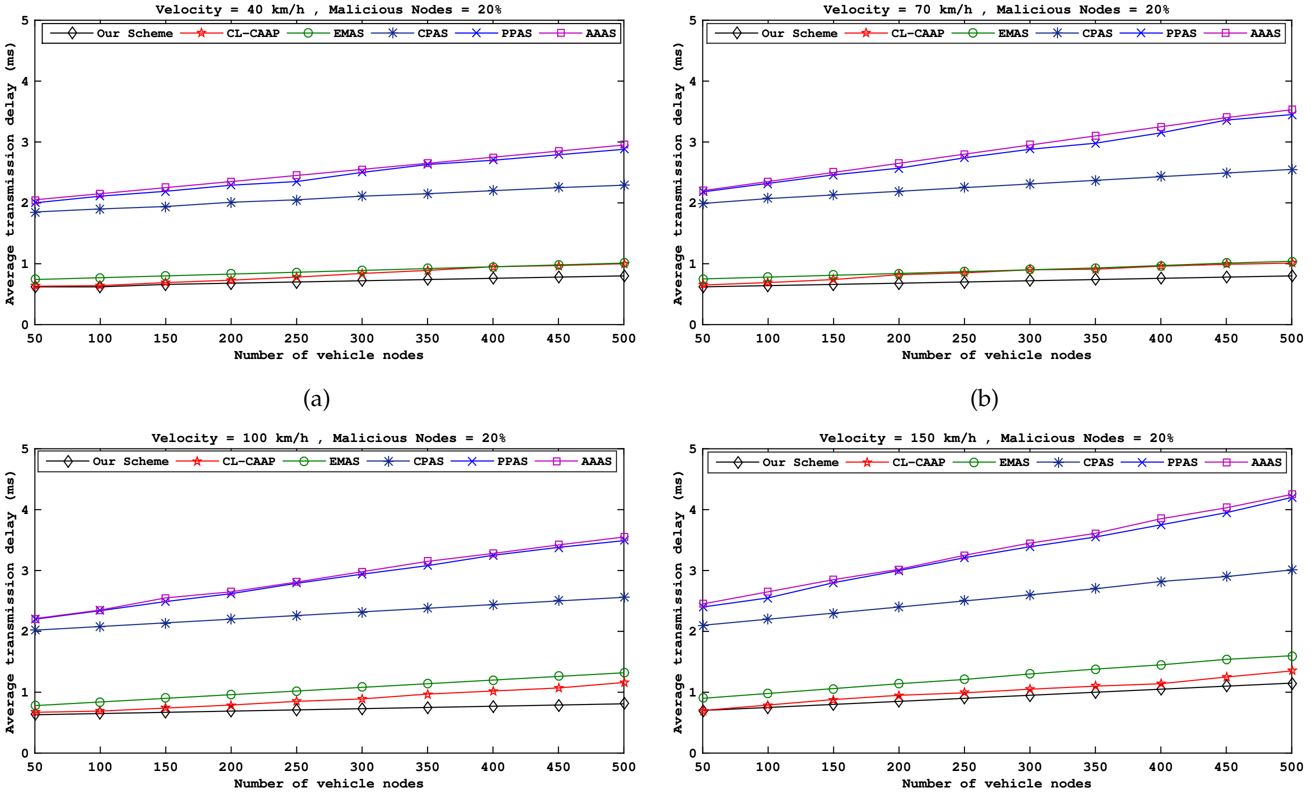
For this example, in the worst case when all existing messages in the batch are invalid, we use one BMV(10), two BMV(5), two BMV(3), and ten SMV where BMV(x) is a batch message verification on a batch contained x messages and SMV is a single message verification. As illustrated in Table 4, the batch message verification's computation cost and in addition finding invalid messages is $BMV(10) + 2 * BMV(5) + 2 * BMV(3) + 10 * SMV = (0.4421 * 10 + 0.4420) + 2 * (0.4421 * 5 + 0.4420) + 2 * (0.4421 * 3 + 0.4420) + 10 * 1.32610 = 26.9656(ms)$ and the total overhead cost for only finding invalid messages is $26.9656 - BMV(10) = 26.9656 - (0.4421 * 10 + 0.4420) = 22.1026(ms)$. Whereas, the total cost of computation for 10 messages using the single message verification is $10 * 1.32610 = 13.2610(ms)$.

The mathematical proof shows that it is better to use the single message verification instead of batch message verification in the proposed scheme, but we experimentally found that the proposed scheme with both batch and single message verification is much better than the scheme with only single verification. Hence, we have separately simulated the proposed scheme with only SMV and with SMV & BMV under the different density when 20% of participated vehicles in the network are malicious nodes. The comparison of obtained computation cost shows that the performance of proposed scheme with SMV & BMV about 49% is better than the scheme with only SMV (see Figure 10).

7.5. Quotient Filter Analysis

The probabilistic data structure is extremely useful for big data generated in VANET [37]. It usually uses to enhance lookup performance and consuming less memory. In this section, we evaluate the quotient filter utilized in our authentication scheme. To evaluate efficiency of the proposed QF-based scheme, it has been compared to an approach based on bloom filter (BF), counting bloom filter (CBF), hash table (HT), and B+ tree (B+).

The obtained results in the proposed fog-enabled VANET are shown in Figure 11. In this figure, the delay has been reflected in comparison among the above approaches. It is clear that the proposed scheme has



(a)

(b)

(c)

(d)

Figure 8: Average transmission delay on different number of vehicles (a) velocity = 40 km/h, malicious node = 20% (b) velocity = 70 km/h, malicious node = 20% (c) velocity = 100 km/h, malicious node = 20% (d) velocity = 150 km/h, malicious node = 20%.

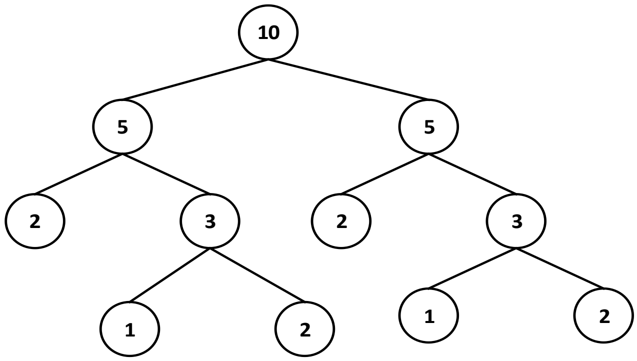


Figure 9: Segmentation of the batch contained 10 messages by the proposed algorithm.

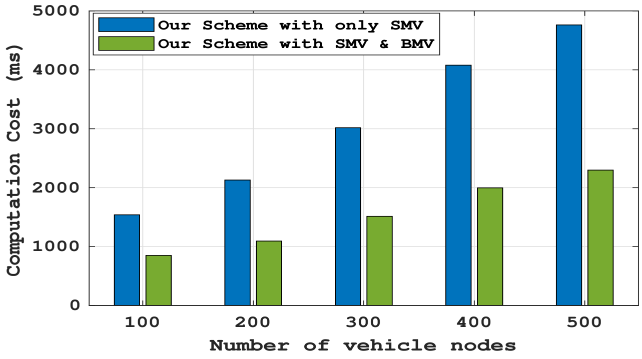


Figure 10: Comparison of computation cost of proposed scheme with SMV and with SMV&BMV.

comparatively less delay relative to other schemes. An average of overall improvement of 32.51, 34.70, 39.40, and 44.18 percent has been observed in this figure.

As explained in [38], QF needs 0.3 sec to extract 10000 packets from a standard database and load into memory where the size of each packet is 1166 bytes; whereas 0.6 sec is needed for BF. Building on this, QF has a throughput of around 310 Mbits/sec, whereas BF has a throughput of 155 Mbits/sec. It indicates that QF performs better than BF in terms of execution time and throughput.

8. Conclusion

In this work, we have presented a security and privacy scheme based on node and message authentication. In the proposed scheme, fog nodes are deployed to the edge of the vehicular network to minimize latency and improve security, while the RSUs host the

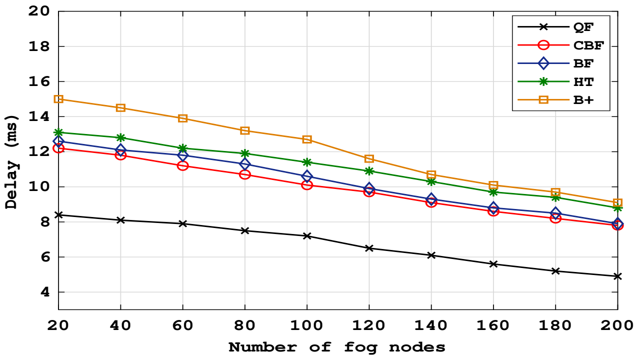


Figure 11: Comparative delay evaluation of QF with BF, CBF, HT, and B+.

fog nodes. Due to the large number of vehicle nodes as well as the big data generated in VANET, the quotient filter is utilized to keep the information of authorized and unauthorized vehicles. The proposed QF-based node authentication scheme ensures the legitimacy of the nodes entering the network. In fact, the authenticity of the vehicle node is checked before initiating data sharing. Additionally, the ECC-based authentication scheme ensures the message's integrity by signing the messages and verifying the signatures. We utilized a pseudonym for vehicle nodes to meet privacy-preserving and maintain anonymity. As shown in the security analysis, our scheme meets the security requirement of VANET appropriately and is suitable to be in real-life scenarios. Furthermore, the performance analysis represents that our scheme outperforms the existing related works.

Acknowledgement

This work is supported by Iran University of Science and Technology. Muhammad Khurram Khan is supported by Researchers Supporting Project number (RSP-2021/12), King Saud University, Riyadh, Saudi Arabia.

References

[1] I. F. Akyildiz, M. Pierobon, S. Balasubramaniam, Y. Koucheryavy, The internet of bio-nano things, IEEE Communications Magazine 53 (3) (2015) 32–40.

[2] L. Ye, L. Kong, K. Z. Ghafoor, G. Chen, S. Mumtaz, LAB: Lightweight adaptive broadcast control in DSRC vehicular networks, Wireless Communications and Mobile Computing 2018 (2018).

[3] S. Yi, C. Li, Q. Li, A survey of fog computing: concepts, applications and issues, in: Proceedings of the 2015 workshop on mobile big data, 2015, pp. 37–42.

[4] S. A. Soleymani, S. Goudarzi, M. H. Anisi, M. Zareei, A. H. Abdullah, N. Kama, A security and privacy scheme based on node and message authentication and trust in fog-enabled VANET, Vehicular Communications 29 (2021) 100335.

[5] S. Jiang, X. Zhu, L. Wang, An efficient anonymous batch authentication scheme based on HMAC for VANETs, IEEE Transactions on Intelligent Transportation Systems 17 (8) (2016) 2193–2204.

[6] R. G. Engoulou, M. Bellaïche, S. Pierre, A. Quintero, VANET security surveys, Computer Communications 44 (2014) 1–13.

[7] L. Wei, J. Cui, Y. Xu, J. Cheng, H. Zhong, Secure and lightweight conditional privacy-preserving authentication for securing traffic emergency messages in VANETs, IEEE Transactions on Information Forensics and Security 16 (2020) 1681–1695.

[8] Y. Wang, H. Zhong, Y. Xu, J. Cui, G. Wu, Enhanced security identity-based privacy-preserving authentication scheme supporting revocation for vanets, IEEE Systems Journal 14 (4) (2020) 5373–5383.

[9] S. Soleymani, M. Anisi, A. H. Abdullah, M. A. Ngadi, S. Goudarzi, M. K. Khan, M. N. Kama, An authentication and plausibility model for big data analytic under LOS and NLOS conditions in 5g-vanet, Science China Information Sciences 63 (12) (2020) 1–17.

[10] S. A. Soleymani, A. H. Abdullah, M. Zareei, M. H. Anisi, C. Vargas-Rosales, M. K. Khan, S. Goudarzi, A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing, IEEE Access 5 (2017) 15619–15629.

[11] L. Lyu, J. Jin, S. Rajasegarar, X. He, M. Palaniswami, Fog-empowered anomaly detection in IoT using hyperellipsoidal clustering, IEEE Internet of Things Journal 4 (5) (2017) 1174–1184.

[12] H. A. Khattak, S. U. Islam, I. U. Din, M. Guizani, Integrating fog computing with VANETs: A consumer perspective, IEEE Communications Standards Magazine 3 (1) (2019) 19–25.

[13] M. H. Y. Moghaddam, A. Leon-Garcia, A fog-based internet of energy architecture for transactive energy management systems, IEEE Internet of Things Journal 5 (2) (2018) 1055–1069.

[14] D. E. Knuth, The art of computer programming, Vol. 3, Pearson Education, 1997.

[15] M. A. Bender, M. Farach-Colton, R. Johnson, R. Kraner, B. C. Kuszmaul, D. Medjedovic, P. Montes, P. Shetty, R. P. Spillane, E. Zadok, Don't thrash: How to cache your hash on flash., Proc. VLDB Endow. 5 (11) (2012) 1627–1637.

[16] M. Raya, J.-P. Hubaux, Securing vehicular ad hoc networks, Journal of computer security 15 (1) (2007) 39–68.

[17] C. Zhang, X. Lin, R. Lu, P.-H. Ho, X. Shen, An efficient message authentication scheme for vehicular communications, IEEE transactions on vehicular technology 57 (6) (2008) 3357–3368.

[18] T. W. Chim, S.-M. Yiu, L. C. Hui, V. O. Li, SPECS: Secure and privacy enhancing communications schemes for VANETs, Ad Hoc Networks 9 (2) (2011) 189–203.

[19] Y. Liu, L. Wang, H.-H. Chen, Message authentication using proxy vehicles in vehicular ad hoc networks, IEEE Transactions on vehicular technology 64 (8) (2014) 3697–3710.

[20] C. Zhang, R. Lu, X. Lin, P.-H. Ho, X. Shen, An efficient identity-based batch verification scheme for vehicular sensor networks, in: IEEE INFOCOM 2008-The 27th Conference on Computer Communications, IEEE, 2008, pp. 246–250.

[21] K.-A. Shim, CPAS: an efficient conditional privacy-preserving authentication scheme for vehicular sensor networks, IEEE Transactions on Vehicular Technology 61 (4) (2012) 1874–1883.

[22] H. Zhong, S. Han, J. Cui, J. Zhang, Y. Xu, Privacy-preserving authentication scheme with full aggregation in VANET, Information Sciences 476 (2019) 211–221.

[23] D. He, S. Zeadally, B. Xu, X. Huang, An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks, IEEE Transactions on Information Forensics and Security 10 (12) (2015) 2681–2691.

[24] J. Li, K.-K. R. Choo, W. Zhang, S. Kumari, J. J. Rodrigues, M. K. Khan, D. Hogrefe, EPA-CPPA: An efficient, provably-secure and anonymous conditional privacy-preserving authentication scheme for vehicular ad hoc networks, Vehicular Communications 13 (2018) 104–113.

[25] J. Li, Y. Ji, K.-K. R. Choo, D. Hogrefe, CL-CPPA: certificate-less conditional privacy-preserving authentication protocol for the internet of vehicles, IEEE Internet of Things Journal 6 (6) (2019) 10332–10343.

[26] J. Cui, L. Wei, J. Zhang, Y. Xu, H. Zhong, An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks, IEEE Transactions on Intelligent Transportation Systems 20 (5) (2018) 1621–1632.

[27] C. Huang, R. Lu, K.-K. R. Choo, Vehicular fog computing: architecture, use case, and security and forensic challenges, IEEE Communications Magazine 55 (11) (2017) 105–111.

[28] S. Goudarzi, M. H. Anisi, A. H. Abdullah, J. Lloret, S. A. Soleymani, W. H. Hassan, A hybrid intelligent model for network selection in the industrial Internet of Things, Applied Soft Computing 74 (2019) 529–546.

[29] J. Cui, J. Zhang, H. Zhong, Y. Xu, SPACF: A secure privacy-preserving authentication scheme for VANET with cuckoo filter, IEEE Transactions on Vehicular Technology 66 (11) (2017) 10283–10295.

[30] S. Garg, A. Singh, K. Kaur, G. S. Aujla, S. Batra, N. Kumar, M. S. Obaidat, Edge computing-based security framework for big data analytics in VANETs, IEEE Network 33 (2) (2019) 72–81.

[31] D. Pointcheval, J. Stern, Security arguments for digital signatures and blind signatures, Journal of cryptology 13 (3) (2000) 361–396.

[32] X. Yi, An identity-based signature scheme from the weil pairing, IEEE communications letters 7 (2) (2003) 76–78.

[33] D. Pointcheval, J. Stern, Security arguments for digital signatures and blind signatures, Journal of cryptology 13 (3) (2000) 361–396.

[34] Y. Jiang, S. Ge, X. Shen, AAAS: An anonymous authentication scheme based on group signature in VANETs, IEEE Access 8 (2020) 98986–98998.

[35] Y. Xie, L. Wu, J. Shen, A. Alelaiwi, EIAS-CP: new efficient identity-based authentication scheme with conditional privacy-preserving for VANETs, Telecommunication Systems 65 (2) (2017) 229–240.

[36] MIRACL core library, https://miracl.com/blog/evolving-the-miracl-core-library-4-min-read/.

[37] S. A. Soleymani, S. Goudarzi, M. H. Anisi, N. Kama, S. Adli Ismail, A. Azmi, M. Zareei, A. Hanan Abdullah, A trust model using edge nodes and a cuckoo filter for securing VANET under the NLoS condition, Symmetry 12 (4) (2020) 609.

[38] M. Al-Hisnawi, M. Ahmadi, Deep packet inspection using quotient filter, IEEE Communications Letters 20 (11) (2016) 2217–2220.