Article

Lightweight Secure Authentication and Key Distribution Scheme for Vehicular Cloud Computing

Hadjer Goumidi ¹, Saad Harous ²*, Zibouda Aliouat ¹ and Abdelhak Mourad Gueroui ³

¹ LRSD Laboratory, Computer Science Department, Ferhat Abbas University of Setif1, 19000 Setif, Algeria; hadjer.goumidi@univ-setif.dz (H.G.); zaliouat@univ-setif.dz (Z.A.)
² College of Information Technology, United Arab Emirates University, 15551 Al-Ain, United Arab Emirates
³ LI-PaRAD Laboratory, Université Paris Saclay, University of Versailles Saint-Quentin-en-Yvelines, 78000 Versailles, France; mourad.gueroui@uvsq.fr
* Correspondence: harous@uaeu.ac.ae

Abstract: A vehicular ad-hoc network (VANET) is the basic block in building an intelligent transportation system that improves the traffic flow and makes needed information conveniently accessible. VANET depends on a dense exchange of sensed data between vehicles and Road Side Units (RSUs). A large amount of sensed data requires a huge computation and storage capabilities, which is provided by the vehicular cloud computing (VCC). However, the security problems of data confidentiality, access control, vehicles’ authentication, and conductors’ privacy in VCC are issues that need to be solved. In this paper, we propose an efficient algorithm to ensure VCC security and privacy. We use Pseudo-ID instead of vehicles’ real ID to provide conductors’ privacy, Identifier-Based Signature mechanism is used to guarantee vehicles’ authentication, and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) algorithm is used for key distribution. Our lightweight secure authentication and key distribution scheme for vehicular cloud computing (GUARANTY) ensures a secure keys distribution to minimize the encryption and decryption computation cost. Vehicles use a symmetrical cryptography in their communication. We analyze the security of our algorithm using AVISPA tool. We use this tool to simulate insiders and outsiders attacks. We evaluate our algorithm’s performance in terms of computation delay and reception rate.

Keywords: VANET; VCC; security; privacy; CP-ABE; IBS

1. Introduction

The technological development has affected all aspects of our daily life, particularly the communication area, which is evolving very rapidly due to the advances in wireless technology. This lead to the introduction of vehicular ad-hoc network (VANET), which has resulted in reducing the problem of road accidents. VANET is a network of vehicles that are able to self-organized. These vehicles can communicate directly with each other or via road sides units with the aim to achieve a safe and efficient transportation traffic system. Recently, VANET is considered the main building block for the future intelligent transportation system. Generally, VANET consists of On-Board Units (OBU) installed in the vehicles (enabling Vehicle to Vehicle (V2V) communication) and Road Side Units (RSU) deployed on the roadside (enabling Vehicle to RSU (V2R) communication). These days, vehicles are equipped with a powerful on-board processing unit, a device capable of storing large amount of data, GPS device, cameras, radio transceiver, and enough sensors that give the vehicles the ability to collect data traffic environment and exchange it among themselves. This huge sensed data requires large space capacity to be stored and fast computing capability to be manipulated [1], which led to the introduction of vehicular cloud computing (VCC). This new environment enables vehicles to request storage, computing, network, cooperation, and sensing services [2,3]. It provides new
applications, such as autonomous driving, urban surveillance, traffic management, and many others [4].

The combination of cloud computing and VANETs creates new challenges, such as resource heterogeneity, loss of communication, and, especially, the privacy and security. The interaction among VCC members raises many security issues and attacks problems, such as Man in the Middle Attack (MiM), which is based on the identity usurpation of a legitimate node. To deal with an MiM attack, it is necessary to ensure a confidential solution by using powerful cryptography algorithms with secure authentication and data integrity verification.

Denial of Service (DOS) attack aims to consume system resources and makes them unavailable to authentic users by sending dummy messages to jam the channel. In a Brute force Attack, the attacker tries to get sensitive information, such as user password or personal identification number. In a Sybil Attack, the attacker generates multiple fake vehicles with the same identity on the road to make an illusion that the network is heavily congested [5].

VCC environment is usually used by vehicles moving at high speeds. In this kind of highly dynamic environment, traditional security solutions have to deal with challenging issues, which are due to the sophisticated communication system used by the vehicles, the changing users' clusters, real-time constraints, etc. The VCC communication mechanism must must protect the transmitted messages’ privacy and confidentiality. To stop attackers from altering, injecting, and/or replaying messages, vehicles have to be authenticated.

Traditional asymmetrical message encryption could be a solution; however, it is costly to encrypt and decrypt the messages using public/private keys, which leads to an inefficient resource access. Another approach is to use the vehicle’s public key of the receiver to encrypt each message, sign the message, and send it. However, in this method a huge number of ciphertext messages will be sent since a message is usually sent to many vehicles. Which makes it very difficult to satisfy the data dissemination real-time constraints required by vehicular applications. It is a complex issue because the safety in these kind of applications require strong timing constraints. Since there is no centralized access control to disseminate encrypted messages in an environment that is very dynamic, the access control problem is not solved in a satisfying way yet.

We present a liGht-weight secURe AutheNticaTion and keY distribution scheme for vehicular cloud computing (GUARANTY) to ensure VCC security and privacy. We used Pseudo-ID instead of the vehicles’ real ID to ensure conductors’ privacy, Identifier-Based Signature (IBS) mechanism to ensure vehicles’ authentication, and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) algorithm [6] for key distribution. Our proposed solution is based on the secure distribution of keys because, once we guarantee that keys are distributed securely, we can just open a session between two entities using symmetrical cryptography. This way, we can ensure fast data encryption. The contributions of this work are:

1. We propose a key distribution model that is efficient and secure. We use CP-ABE algorithm for keys encryption instead of using this algorithm for messages encryption. This way, we just need to encrypt keys during the registration step. Vehicles can then communicate with RSU using symmetrical encryption. The model securely ensures fast transmission of messages.
2. GUARANTY protocol ensures vehicles’ privacy by using Pseudo-ID instead of the vehicle’s original ID.
3. Our scheme ensures vehicles authentication by using IBS signature scheme to authenticate legitimate vehicles.
4. V2V communication is secured in our approach by using authentication key ‘AuK’ transmitted by RSU to all authenticated vehicles.

The remaining sections of this work are organized as: Section 2 presents previous works in the aspects of VANET and VCC security using CP-ABE algorithm and IBS scheme. Section 3 discusses different algorithms related to our proposed solution. Section 4 presents
GUARANTY proposed solution. We discuss security analysis and evaluate our proposed solution performance in Section 5. Finally, we conclude the paper in Section 6.

2. Related Work

In this section, we highlight the proposed solutions that preserve VANET and VCC security and privacy. In Table 1, we compare these existing solution to our proposed GUARANTY.

| Ref  | Advantages                                                                 | Techniques Used                                                                 | Limitations                                                                 |
|------|---------------------------------------------------------------------------|--------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| [7]  | Preserves vehicles’ privacy and authentication, short signature verification time | IBS signature; batch verification                                              | Vulnerable because it does not support a proper encryption scheme          |
| [8]  | Ensures messages confidentiality and integrity; preserves vehicles’ privacy | Lagrange interpolation; algebraic signature                                    | Vulnerable against Sybil and impersonate attacks                           |
| [9]  | Provides vehicles’ privacy and trustworthiness                             | Public keys as identity; reputation management                                 | Certificate update; reputation score update; high computation time and storage capacity |
| [10] | Preserves privacy, authentication, confidentiality using Blockchain technology | Pseudo-ID; key of group; blockchain for authentication                        | High computation time; high storage capacity                               |
| [11] | Evaluating privacy protection                                              | Analyzing users’ historical behaviour                                          | Vulnerable against eavesdropping and spoofing attacks                      |
| [12] | Secure communication with low delay; avoids information tracking           | WAVE protocol; Blowfish protocol                                              | Vulnerable against impersonate and Sybil attacks                           |
| [13] | Provides the confidentiality of conductor’s location which preserves vehicles’ privacy | Location-based encryption                                                     | Messages can be decrypted in the case where malicious vehicle exists at the region Z and at the confident time |
| [14] | Provides V2R authentication and V2V authentication with and without RSU; preserves vehicles’ privacy | IBS signature; IBOOS signature                                                | Vulnerable against eavesdropping, spamming and malware attacks             |
| [15] | Provides access control and confidentiality                                | CP-ABE                                                                        | Significant decryption time                                               |
| [16] | Preserves access control, confidentiality, and low decryption time         | CP-ABE where the decryption task is carried out by RSU                        | Lacks of vehicles’ privacy and authentication                              |
| [17] | Ensures vehicles’ privacy, authentication, access control, and confidentiality | CP-ABE to encrypt messages; IBS signature                                       | Lacks of V2V authentication; privacy can be affected when transmitting ID in the registration; high computation time |
| [18] | Provides an efficient and an effective miner node selection strategy in the application of vehicular blockchain | Blockchain technology; service score-based protocol; PBFT consensus algorithm | Lacks of authentication, confidentiality, and access control; high storage capacity |
| [19] | Provides privacy in LPR systems using blockchain                          | Ethereum private blockchain; Smart contract                                    | Lacks of authentication and confidentiality                                 |
| [20] | Ensures vehicles’ safety using a trust model                              | Mathematical statistical methods; computation of vehicles’ trust value         | Lacks of authentication, confidentiality and integrity                     |
|      | GUAR-ANTY                                                                 | CP-ABE for key exchange; symmetric cryptography for messages exchange; IBS signature and key group for authentication; Pseudo-ID for privacy | Lacks of messages integrity                                               |

Table 1. Existing algorithms classification.
The attribute-based encryption (ABE) [6] is used in diverse access control applications in VANET. Huang, in Reference [15], introduces the first ciphertext-policy attribute-based encryption (CP-ABE) algorithm in VANET, where each vehicle has its access rights and capabilities based on its attributes. Vehicles that have certain attributes satisfy the access policy can access the broadcasted message and decrypt it with its OBU module. Vehicles’ OBU processors have a slow computation power. It represents a significant challenge to process the message. The decryption time required on high performance mobile terminal would be about 30 s for an ABE ciphertext that contains 100 attributes, which leads to a significant battery energy consumption [21]. OBUs have limited processing power to make VANETs economically viable.

To reduce processing load used by the decryption, the authors in Reference [16] propose “a Secure and Efficient Message Dissemination with policy enforcement in VANET”. This protocol uses a ciphertext-policy attribute-based encryption (CP-ABE) to to allow different access control services. Most of the computations needed for decryption are carried out by RSU. This algorithm consists of three phases: registration phase, dissemination phase, and decryption phase. During the registration phase, vehicles register at the nearest RSU by sending their attributes information, such as speed, direction, and position. The RSU then transfers this information to the top trusted center for key generation. The center generates two keys, Attribute key ‘AK’ and Security key ‘SK’. The Attribute key will be transmitted to the RSU and the vehicles have to keep the security key private. In the message dissemination phase, when the center receives an event rapport, it generates an alert message to the related vehicles, and it encrypts the alert message with access policy. Only vehicles which have the set of attributes satisfying the access policy can decrypt the message. In the decryption phase, when the RSU receives the encrypted emergency message, it must perform a fast decryption test using the Attribute Key ‘AK’ of registered vehicles. The RSU will transform a ciphertext from vehicles that their attribute set satisfied the access policy in the new ciphertext. When the responding vehicles receive the new ciphertext, they only need to do a little decryption computation using their secret key. However, this algorithm ensures only the access control of messages; the privacy and authentication of vehicles are not ensured, and the attributes of vehicles are transmitted in unsecured way. In addition, the transmission of ‘SK’ from the center to the vehicle can be altered, and each message must be encrypted using the ABE algorithm, which takes calculation time.

In Reference [15], authors integrate identity-based encryption scheme (IBES) and identity-based signature algorithm to encrypt messages and ensure the confidentiality and vehicles’ authentication. The identity-based signature algorithm is based on the use of batch and rapid verification in VANET communication [22].

The work in Reference [17] proposes a cloud-based security and privacy-aware information dissemination over ubiquitous VANET algorithm. The authors combine the identity-based signature (IBS) with pseudonym to ensure the privacy and the authentication of vehicles. Moreover, they use CP-ABE method to ensure access control for both VANET and Cloud. Vehicles register themselves on the cloud via the nearest RSU, and the cloud then generates vehicles’ private key and the Pseudo-ID, which encrypts the vehicles’ ID using the system master key. The cloud encrypts the message using CP-ABE method to ensure access control and signs the message using IBS signature to ensure authentications of vehicles. Vehicles that satisfied the access policy decrypt the message. This algorithm ensures the authentication of vehicles and the confidentiality of messages, but the decryption step takes time because vehicles have limited computing capacity. This algorithm ensures only the authentication in V2R communication; V2V authentication is not satisfied. The privacy of vehicles can be affected, and the transmission of vehicle’s ID to the cloud can be detected, by a malicious vehicle before Pseudo-ID generation.

The authors in Reference [14] propose a Secure Vehicular Communication using ID-Based signature algorithm, which allows establishing a secure communication between vehicles. It uses Pseudo-ID to ensure the privacy of vehicles and IBS or IBOOS signature to
provide authentication. This algorithm provides V2R, V2V with RSU, and V2V without RSU authentication. It can ensure security against authentication-based attacks, such as Sybil attack and Impersonate attack, and isolate the malicious nodes from the network. In V2R communication, RSU affects an offline signature to authenticate vehicles. V2V authentication is carried out by using the offline signature generated by RSU to calculate an online signature used to verify the authentication of vehicle. Vehicles under the control of different RSUs can communicate with each other by sending an authentication request to the nearby RSU about the offline signature of the vehicle. In the case of the absence of RSU, vehicles can communicate securely without the control of RSU by using RTA algorithm.

Hussain et al. [13] present a Geolock-based encryption algorithm in which the receiver has to be in the location indicated by the sender. The location-based encryption uses the location information to generate the encryption and decryption keys. This algorithm can provide the location confidentiality against the outsider, by using KZ (key is given to all vehicles in the region Z) and KRSU (key distributed by RSU to all vehicles within its range) to encrypt the message, and it can keep the insiders from manipulating the context of the message by changing these two keys periodically. Vehicles insiders must be exactly in the region Z at the confident time to decrypt the message.

To solve the problems of privacy breach in cloud-based V2X communication, the authors in Reference [11] design a Privacy Assessment method with Uncertainty consideration (PAU). In this scheme, the authors focus on evaluating the privacy protection of vehicles by analyzing the users’ historical behavior under cloud-based V2X scenarios. To measure the record in the users’ historical behavior, authors expand a subjective logic of uncertainty. PAU calculates the vehicle’s real-time privacy capability by observing this vehicle’s real-time communication. It uses the privacy aggregation algorithm to improve the accuracy of privacy assessment.

Authors in Reference [12] propose a Sensor-enabled Secure Vehicular Communication for emergency messages dissemination using cloud services (SSVC) algorithm. SSVC improves communication efficiency and reduces its delay using WAVE protocol. It uses Blowfish protocol [23] to prevent other vehicles from tracing the information from the cloud storage.

In Reference [10], authors propose Blockchain security technique for vehicles communication to ensure secure vehicles’ communication and authentication. In this scheme, authors use pseudo-identity instead of real identity to provide vehicle privacy. These pseudo-identities are stored in a public ledger as transactions, where only RSUs can read the content of the ledger to check vehicles’ authentication. Authenticated vehicles have a group key for secure data transfer.

Authors in Reference [9] propose a Blockchain-based Anonymous Reputation System (BARS), based on blockchain technology, to build a trust model for VANET that preserves the privacy. They use the public key as pseudonym, instead of using real identity, to ensure the vehicle’s privacy. They propose a reputation evaluation method, which does not allow forged messages’ distribution. They apply direct historical interactions and indirect opinion about vehicles to assess vehicles’ reputation score.

In Reference [18], authors propose an efficient and scalable proof-variant that can be implemented in various VANET applications based on blockchain, in combination with Practical Byzantine Fault Tolerance (PBFT) consensus. They introduce a new proof-variant, named Proof of Driving (PoD), to randomize the selection of honest miners based on earning coins. Additionally, this work introduces a filtering technique based on Service Standard Score of the vehicular miner nodes to detect and eliminate the malicious nodes.

The work presented in Reference [19] proposes a blockchain storage architecture focused on license plate recognition (LPR) systems for smart cities focusing on privacy, performance, and security. The proposed architecture relies on the Ethereum platform and smart contracts. Additionally, this work is the first one that proposes a solution to provide privacy in LPR systems using blockchain.
Identity-Based Data Transmission protocol (EIBDT) is a protocol used to protect transmitted data between vehicles and RSUs. It is based on the identity-based encryption method with a low computation task. The authors in Reference [8] propose to use the Lagrange interpolation method instead of using the bilinear mapping scheme for integrity and confidentiality protection with a low computation task. They use an algebraic signature to encrypt the vehicle’s real identity and use the identity-based signature algorithm to guarantee vehicle’s privacy and avoid the pseudonym management complexity.

The work in Reference [20] presents a trust model based on the mathematical statistics method, which combines entity trust with data trust to enhance the safety of VANET networks. The proposed model relies on statistical principles, such as significance test, hypothesis test, and confidence interval. It can help vehicles to judge whether an event message can be trusted or not, and it can help the reputation management center to calculate trust values of all vehicles. The authors in Reference [7] present a secure vehicle traffic data dissemination and analysis protocol in VCC, where they adopt the pseudonym technique to preserve vehicles privacy. An anonymous credential approach is used to generate vehicles’ authorization. Identifier-based signature algorithm is used to ensure the authentication of vehicles. Authors also use batch verification method to reduce the signature verification time, and the pseudo revocation list to discard the messages generated by revoked vehicles. However, this technique is susceptible because it does not support an appropriate encryption scheme. The previously presented algorithms are based on message encryption. However, keys dissemination is the most challenging steps in security because these keys will be used in messages encryption. The efficiency of security algorithm depends directly on the efficiency of keys dissemination.

In this work, we propose a protocol based on secure and efficient keys dissemination, with vehicles authentication, conductors’ privacy, and messages confidentiality.

3. Background
3.1. Network Architecture

Figure 1 illustrates our proposed network’s architecture. It is mainly consists of the following components:

1. Vehicle: In our network model, a vehicle has sensors to sense traffic congestion and status, Global Positioning System (GPS), On-Board Unit (OBU), and Event Data Recorder (EDR). The vehicle is the main component of the proposed network architecture.
2. On-Board Unit (OBU): Each vehicle is equipped with an OBU to collect its information, such as the position, speed, and acceleration/deceleration, and transmits it to vehicles or RSUs within its transmission range through the wireless medium. As stated in Reference [24], “it uses the Wireless Access in Vehicular Environment (WAVE) standard, which is based on the emerging IEEE 802.11p specification”.
3. Road Side Units (RSU): It is a fixed infrastructure deployed on the roadside to monitor the traffic or in the road intersections to manage the traffic lights. It acts as a gateway.
for the vehicles to access the Internet. The main use of RSUs in the proposed approach is to provide secret key and authentication key transmission in a secure manner.

4. Vehicle-to-Vehicle (V2V) communication: Vehicles communicate with other vehicles within their transmission range through a direct wireless transmission of data without need of using fixed infrastructures. In GUARANTY algorithm, vehicles must be authenticated to establish a secure communication between them. The registration, Pseudo-ID formation, and authentication phases are used at the start of vehicles’ trip or when the vehicle’s secret key is threatened. In this case, the vehicle can re-register itself at the nearby RSU to have a new secret key.

5. Vehicle-to-RSU (V2R) communication: This is the communication between vehicles and RSU, and it is usually done via wireless transmission. In our model, vehicles send a registration request for key generation and a signature request for its authentication.

6. Vehicle-to-Internet (V2I) communication: It is a communication, between vehicles and the Internet, which can be established directly or indirectly using the RSU.

7. Cloud Center: A virtual server, in a cloud computing platform, which offers many services, such as computing capabilities, storage, and communication resources on demand. The cloud generates the keys. All vehicles’ information is stored in the database and then used in vehicles’ private key generation.

3.2. Network Security Model

In this subsection, we describe our network security model, by presenting our algorithm’s assumptions and the list of attackers that can be carried out in the network.

Assumptions

• The RSUs are connected via a wired connection, and the communication between them is carried out via a secure channel.
• Vehicles are in an area covered completely by RSUs.
• The cloud is trusted and secure.
• The communication between the cloud and the RSU is carried out via a secure channel.
• The vehicles contain OBUs that are equipped with processing and storage capabilities to carry out the encryption/decryption.
• Cloud-based Revocation Authorities (CRAs) are responsible for the revocation process.

Network Attackers: the wireless medium used in vehicles’ communications has several disadvantages that makes it vulnerable to different kind of attacks. In this paper, we aim to solve the following security challenges: different types of attacks. The aim of our paper is to tackle the following security issues:

• Attacks on confidentiality: The aim of these kind of attacks is to access confidential data illegally. Eavesdropping and Man in the Middle (MiM) are examples of these attacks.
• Attacks on authentication: The objective of this attack is to claim that it is an authorized vehicle, such as impersonate and Sybil attacks.
• Attacks on vehicles’ privacy: The objective of this attack is to expose the identity of vehicles, such as tracking attacks.

3.3. Access Structures

Let \( \{\text{Att}_1, \text{Att}_2, \ldots, \text{Att}_n\} \) be a set of parties and an Access Structure (AS). We say that a collection \( \text{AS} \subseteq 2^{\text{Att}_1, \text{Att}_2, \ldots, \text{Att}_n} \) is monotone \( \forall B, C : \text{if } B \in \text{AS} \text{ and } B \subseteq C, \text{ then } C \in \text{AS}. \)

We say that a collection \( \text{AS} \subseteq 2^{\text{Att}_1, \text{Att}_2, \ldots, \text{Att}_n} \) is non-monotone \( \forall B, C : \text{if } B \in \text{AS} \text{ and } B \subseteq \complement C, \text{ then } C \notin \text{AS}. \)

Monotone AS is a monotone collection of non-empty subsets of \( \{\text{Att}_1, \text{Att}_2, \ldots, \text{Att}_n\} \), i.e., \( \text{AS} \subseteq 2^{\text{Att}_1, \text{Att}_2, \ldots, \text{Att}_n} / \text{Att}_i \neq \emptyset \). The sets in AS are called the authorized sets, and the sets not in AS are called the unauthorized sets \([25]\).

In our context, the AS contains the authorized parties, which are represented by the attributes. These attributes in our proposed solution are the vehicles’ information, such as
vehicle identifiers, locations, speeds, direction values, etc. We concentrate on monotone AS. However, authors in Reference [26] proposed to use the non-monotone AS. They add the negation in space of the attributes along with each part. Hence, the number of possible attributes is doubled as compared to the classical scheme and the resulting ciphertext may contain many negations attributes that have are of no use in decrypting the data. It leads to a strong increase in the size of the ciphertext.

3.4. Access Tree

The access tree $\tau$ describes and implements AS. Each leaf node in $\tau$ represents an attribute. Internal node represents a threshold gate defined by its children and threshold value as:

For a node $x$, we denote:
- $N_x$: number of children of $x$.
- $th_x$: threshold value, where:
  - If $th_x = 1$, the threshold gate is an ‘OR’ gate.
  - If $th_x = N_x$, the threshold gate is an ‘AND’ gate.
- $index_x$: represents node’s index.
- $att_x$: is the attribute value associated with the leaf node $x$.
- $parent_x$: is the parent of $x$ in the tree $\tau$.

The evaluation and satisfaction of $\tau$ on a set of attributes is described iterative from the leaves toward the root $R$ of $\tau$.

3.5. Linear Secret Sharing Schemes (LSSSs)

Let $F$ be a finite field, and $\Pi$ be a secret sharing scheme. $\Pi$ is said to be a LSSS over $F$ if:
1. The piece of each party is a vector over, in which there exists a constant $d_k$ for each $k$, and the piece of $F_k$ is taken from $F$.
2. We define a share generating matrix (M) for $\Pi$, with $l$ rows and $n$ columns in a similar way to Reference [25]. “For all $i = 1, ..., l$, where ‘$i$’ is the row of M. The function $\rho$ defines the party labeling row ‘$i$’ as $\rho(i)$. The column vector $v = (s, r_2, \ldots, r_n)$, where $s \in F$ is the secret to be shared, and $r_2, \ldots, r_n \in F$ are randomly chosen. So, $M_v$ is the vector that shares the secret $s$ according to $\Pi$. The share $(Mv)_i$ belongs to party $\rho(i)$” [25].

To show that every LSSS according to the above definition also enjoys the linear reconstruction property, authors in Reference [25,27] suppose that $\Pi$ is an LSSS for the access structure AS. We put $S \in AS$ as any authorized set, and $H \subset \{1, 2, \ldots, l\}$ be defined as $H = \{i : \rho(i) \in S\}$. There exists constants $\{\omega_i \in (Z/p)\}$, where $(i \in H)$, in which, if $[\lambda_j]$ are valid, shares of any secret ‘s’ according to $\Pi$, then $\sum_{i \in H} \omega_i \lambda_j = s$.

The LSSS matrix can be defined by an access tree where the tree’s non-leaf nodes are ‘AND’ and ‘OR’ gates, and leaf nodes are attributes. The matrix’s number of rows is equal to the number of leaf nodes of the access tree.

3.6. Bilinear Maps

Let $G_1$ and $G_2$ are two (multiplicative) cyclic groups of prime order $p$ and $q$. Let $g_1, g_2$ are generators of $G_1$ and $G_2$, respectively, and “$e$” is a bilinear map $e: G_1 \times G_1 \rightarrow G_2$ with the properties:
1. Bilinearity: for all $u, v \in G_1$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g_1, g_1) \neq 1$.

We say that $G_1$ is a bilinear group, in the case of the group action in $G_1$ and the bilinear map $e: G_1 \times G_1 \rightarrow G_2$ can be computed efficiently [27].
3.7. Ciphertext-Policy Attribute-Based Encryption (CP-ABE)

It is a confidential and secure algorithm based on preserving access control of encrypted data. CP-ABE allows users to encrypt data by defining an access policy over attributes, where only users that satisfy the policy can decrypt the message and access to the data. This algorithm has four principal steps [6]:

- **Setup**: The algorithm takes the security parameter and the attribute of the user as input. It generates the public key (PK) and a master key (MK) as output.
- **Encryption**: The algorithm has as input: the PK, a message, and an access structure. It will encrypt the message and generate a ciphertext, where only the users that possess the set of attributes that satisfies the access structure will be able to decrypt the message.
- **Key Generation**: The values of the master key and a set of attributes (describing the key) are the input to the algorithm, which generates a private key SK.
- **Decryption**: The algorithm gets the public key, the ciphertext, and the private key input. The algorithm checks if the set of attributes satisfies the access structure AS (defined in the ciphertext), and then it will decrypt the ciphertext and return the original message.

3.8. Identity-Based Signature (IBS)

It is an algorithm used to ensure the authentication of entities, where the user’s identity is used to compute its signature. The information about the the identity of the signer and the master key is required to verify the signature. The signature is generated [28] as follows:

- **Setup**: The master key is generated by a certified authority and broadcasted to all vehicles via RSUs.
- **Extraction**: Vehicle generates its private key based on its identifier, the master key, and timestamp.
- **Signing**: Vehicles encrypts the message (Msg) using its private key to generate the signature (SIG).
- **Verification**: The validity of the signature is verified by the receiving vehicle.

4. Our Proposed GUARANTY Scheme

Now, we detailed the construction of our algorithm. The different notations used in our scheme are presented at the end of the paper in “Abbreviations”. GUARANTY algorithm has 4 main steps.

4.1. Registration

Each vehicle must register itself at the nearest RSU, and the RSU then transmits vehicles’ information to the cloud for keys generation and for updating vehicles’ information as shown in Algorithm 1. The registration mechanism has three steps, and Figure 2 describes the sequence diagram of the registration process.
Figure 2. Sequence diagram of registration step.

1. Step 1: The cloud assigns an identifier $ID_{RSU}$, a verification number $VN_{RSU}$, and an authentication key ‘AuK’ to each RSU. It also registers all vehicles’ ID with their chassis numbers (CN).

2. Step 2: Each vehicle must register itself at the nearby RSU, when it enters or leaves the particular region, by broadcasting a Registration Request (RR) message encrypted with the public key of RSU ($PK_{RSU}$). The registration request contains the vehicle’s attribute values (vehicle identifier, location, current speed, and the direction value, etc.). The RSU verifies location-related attributes using location proof methodologies [29]. At the end, the vehicle’s attribute values are transmitted to the cloud with the $VN_{RSU}$, where $VN_{RSU}$ ensures the legitimacy of the RSU. In the case where the RSU is malicious, it will be detected by the cloud by checking the $VN_{RSU}$, and then it will be revoked.

$$RR = (T, ID_z, Sp_z, D_z, L_z, ID_{RSU})PK_{RSU}.$$  

$T$ represents the current time, $ID_z$ represents vehicle’s identifier, $Sp_z$ represents the speed of vehicle $Z$, $D_z$ represents the direction of vehicle $Z$, $L_z$ represents the position of vehicle $Z$, and $ID_{RSU}$ represents the identifier of the nearest RSU.

3. Step 3: The cloud generates two keys, after receiving the vehicle’s attribute values, and sends them to the RSU: the attribute key ‘AK’ and the secret key ‘SK’. When RSU receives the two keys, it must encrypt the secret key ‘SK’ using attribute key ‘AK’ of the corresponding vehicle.

For the transmission of the secret key ‘SK’, we apply the CP-ABE algorithm, which has the following five parts:

- Setup: When the cloud receives the attribute values of the vehicle, it runs the setup algorithm. It takes attributes set $S$ as input in which the set of attributes
contains the vehicle’s attributes transmitted by the vehicle and the corresponding chassis number (CN) registered in the cloud. The cloud then generates the public and the master key as output. For this, it chooses two groups $G_1$ and $G_2$ of prime order $P$, and two generators $g_1$ and $g_2$. It chooses also two random exponents $b, \alpha \in \mathbb{Z}_p$. The public key is generated as:

$$PK_c = g_1, g_2, e(g_1, g_2)^a, h_1^b, h_2^b.$$  \hfill (1)

The master key is set as:

$$MK = g^\alpha.$$  \hfill (2)

b Key-generation: To generate the attribute key (AK), the cloud gets the master-key (MK) and a set of attributes $S$ as input. It generates the attribute key AK corresponding to the set of attributes of the vehicle. The algorithm first generates a random number $t \in \mathbb{Z}_p$ and then creates the attribute key as:

$$AK = (K = g_2^s b^t, L = g_2^t, \forall x \in S, K_x = h_1^x),$$  \hfill (3)

where $h_x$ is the attribute string of attribute $x$ hashed to an element of $G_1$. To generate the secret key, the cloud chooses a random number $s \in \mathbb{Z}_p$ and computes the secret key:

$$SK = e(g_1, g_2)^{as}.$$  \hfill (4)

c Encryption: The algorithm defines the access matrix $(M, k)$ based on a given LSSS. The matrix $M$ is defined as $(l \times n)$, where $l$ is the number of rows, $n$ is the number of columns, and $k$ refers to the function that maps rows of $M$ to the attributes. The algorithm selects a random vector $v = (s, y_2, ..., y_n)^T \in \mathbb{Z}_p$, where the exponent $s \in \mathbb{Z}_p$ is randomly chosen as the secret to be shared in secret key generation. The other values are used to share the encryption exponent $s$. It calculates $\lambda_i = M_i v$, where $i$ takes value from 1 to $l$. The algorithm also chooses random $r_1, ..., r_n \in \mathbb{Z}_p$. The ciphertext is published as:

$$CT = (C = g^s, C_i = g^{b s_i h_i - r_i}, D_i = g^{r_i}).$$  \hfill (5)

So, the cloud generates keys, sends them to the RSU, and then it encrypts the secret key using the attribute key and sends the ciphertext to the corresponding vehicle.

d Decryption: In this step, the vehicle takes as input a ciphertext (CT) and the attribute key for a set of attributes $S$. The Lagrange basis polynomials [30] evaluated at zero, $p_i(0)$, are computed for the indices involved in access policy by:

$$p_i(0) = \prod_{j \in S_i, j \neq i} \frac{X_j}{X_j - X_i}.$$  \hfill (6)

The decryption algorithm computes

$$e(C, K) \prod_{i \in A} e(C_i, L)e(D_i, K_i)^{p_i(0)},$$  \hfill (7)

where

$$\prod_{i \in A} e(g_1^{b s_i h_i - r_i}, g_2^{r_i}) = e(g_1^s, g_2^s).$$  \hfill (8)

We use the properties of bilinearity, which allows exponents to be interchanged as

$$e(g_1, g_2) = e(g_1^s, g_2).$$  \hfill (9)
We can write:
\[
e^{g_1 g_2 \alpha s} e^{g_1 g_2^b s t} \prod_{i \in A} e^{g_{bsi} (h'_{ri} - ri, g't) p_i(0)}.
\]
(9)

We use the property: \( e(r, t) e(s, t) = e(rs, t) \).

We write:
\[
e^{g_1 g_2 \alpha s} e^{g_1 g_2^b s t} \prod_{i \in A} e^{g_{bsi} (h'_{ri} - ri, g't) p_i(0)}.
\]
(10)

Since the only \( i \)-dependence is present in the exponent of the denominator, it can be expressed as:
\[
e^{g_1 g_2 \alpha s} e^{g_1 g_2^b s t} \prod_{i \in A} e^{g_{bsi} (h'_{ri} - ri, g't) p_i(0)}.
\]
(11)

Recall the property of Lagrange basis polynomial [30] \( s, \Sigma_{i \in p_i(0)} = p(0) \); thus, it recreates the coefficient at zero which is the secret \( s \). Therefore, it will be:
\[
e^{g_1 g_2 \alpha s} e^{g_1 g_2^b s t} = e^{g_1 g_2^{as}}.
\]
(12)

This is equal to the vehicle’s SK. The vehicle can decrypt the ciphertext and recover its SK, which must be kept private.

Each vehicle, at the start of its trip, registers at the nearest RSU. When a vehicle leaves a region of RSU, the RSU transmits the vehicle’s information to the next RSU located on the vehicle’s direction on a secure channel. If the vehicle changes its direction, it will register with the next nearest RSU again. Table 2 lists the symbols used in the mathematical equations of registration step.

Table 2. Meaning of different notations.

| Symbol | Meaning |
|--------|---------|
| \( g_1, g_2 \) | Generators of a group of prime order. |
| \( \alpha \) | Master secret key component. |
| \( e() \) | Exponential function. |
| \( h_x \) | The attribute string of attribute \( x \) hashed to an element of \( G_1 \). |
| \( M \) | The access matrix. |
| \( s \) | The secret to be shared. |
| \( k \) | Function that maps rows of \( M \) to the attributes. |
| \( \lambda_i \) | Valid shares of the secrets. |
| \( p_i(0) \) | The Lagrange basis polynomials function. |

Algorithm 1 Registration Algorithm.

The cloud registers all vehicles ID with their chassis number and RSUs ID with their VN_{RSU}. RSU broadcasts \( <ID_{RSU}, T, PK_{RSU}> \) in the network. \( V_z \) sends \( <T, Sp_z, D_z, L_z, ID_{RSU}> \) encrypted by \( PK_{RSU} \).

RSU verifies:
\[ \text{if } V_z \text{ is legitimate} \]
- RSU transmits \( Sp_z, D_z, L_z \) to the cloud.
- Cloud runs the Setup algorithm.
- Cloud runs the key generation algorithm.
- Cloud transmit ‘SK’ and ‘AK’ keys to the RSU.
- RSU runs encryption algorithm.
- \( V_z \) runs decryption algorithm.
\[ \text{else} \]
- RSU runs System Revocation.
4.2. Pseudo-ID Formation

After the registration step, each vehicle generates its Pseudo-ID, to ensure privacy preservation in the proposed system as shown in Algorithm 2. The vehicle uses the Pseudo-ID instead of the real ID to ensure that its original ID is not exposed to other vehicles in the network. So, the privacy of vehicles is protected while communicating. Pseudo-ID is generated using the following steps.

1. Step 1: RSU will encrypt the secret key ‘SK’ using the attribute key ‘AK’ and broadcasts it on the network. Vehicles within the range of particular RSU will acquire the encrypted secret key. The vehicle that satisfies the attribute values will decrypt the message and recover the secret key for Pseudo-ID generations.
2. Step 2: Pseudo-ID of the vehicle (z) is generated by encrypting the original ID of the vehicle (z) by its secret key and sends it to the RSU as:
   \[ \text{PID}_z = < T, (ID_z)_SK, ID_{RSU} >, \]
   where \( \text{PID}_z \) is the generated Pseudo-ID of vehicle \( Z \), \( T \) is the current time, \( (ID_z)_SK \) is the encrypted value of the vehicle’s ID using the secret key, and \( ID_{RSU} \) is the ID of the RSU.

Algorithm 2 Pseudo-ID Formation Algorithm.

- \( V_z \) encrypts \( ID_z \) by \( SK_z \).
- \( V_z \) transmits < \( T, (ID_z)_SK, ID_{RSU} > \) to RSU.
- RSU saves \( \text{PID}_z \) and the real \( ID_z \) of vehicle \( z \).

4.3. Authentication

A- V2I and I2V Authentication: Identifier-Based Signature (IBS) algorithm is used to ensure the authentication among vehicles and Infrastructures as illustrated in Algorithm 3. The V2I and I2V authentication are explained next:

1. Step 1: Each RSU broadcasts its information periodically to all vehicles within its range. This information is in this form:
   \[ < ID_{RSU}, T, PK_{RSU} >, \]
   where \( ID_{RSU} \) is the ID of RSU, \( T \) is the current time, and \( PK_{RSU} \) is the public key of RSU.
2. Step 2: Vehicles acquire the information from RSU. Vehicles that need to generate their signature will send a signature request message to the RSU. The signature request has this form:
   \[ < ID_{RSU}, PID_z, T, SIG_z(ID_z||T) >, \]
   where \( ID_{RSU} \) is the ID of RSU, \( PID_z \) is the Pseudo-ID of the vehicle \( Z \), \( T \) is the current time, and \( SIG_z(ID_z||T) \) is the IBS generated by vehicle using its ID and the current time. The \( ID_z \) and the current time values are concatenated, and the resulting value is encrypted using the secret key (\( SK_z \)), which results in \( SIG_z(ID_z||T) \).
3. Step 3: After receiving the signature request from the vehicle, the RSU verifies whether the vehicle is legitimate(by decrypting the signature using the secret key of vehicle \( Z \) and verifying the identifier and the time). If it is a legitimate vehicle, the RSU encrypts the authentication key(\( AuK \)) using the secret key of vehicle \( Z \) and broadcasts it with an authentication message to indicate that \( V_z \) is legitimate. The authentication signature message has this form:
   \[ < ID_{RSU}, T, PID_z, SIG_z(PID_z||T), AuM, (AuK)_{SK_z} >, \]
where \( ID_{RSU} \) is the ID of RSU, \( T \) is the current time, \( PID_z \) is Pseudo-ID of vehicle \( z \), \( \Sigma G_z(ID_z||T) \) is the signature of vehicle \( Z \), \( AuM \) is an authentication message to indicate that the vehicle which has the \( PID_z \) is legitimate, and \( (AuK)_{SK_z} \) is an authentication key encrypted by the secret key of vehicle \( Z \). Only the vehicle authenticated (vehicle \( Z \)) can access to this key. All vehicles within RSU’s transmission range save the Pseudo-ID of vehicle \( Z \) in their authentication table for V2V authentication.

In the case where a vehicle repeatedly moves into the same RSU scope, it does not need to authenticate itself at the RSU each time it wants to establish a communication with it. The first authentication is sufficient because vehicles’ information is saved at the RSU.

B- V2V Authentication: To establish secure communication between vehicles, V2V authentication is carried out (Figure 3 describes the sequence diagram of the V2V authentication process). Algorithm 4 presents the V2V authentication process. When a vehicle wants to communicate with other vehicles, it will generate an authentication signature \( AuSIG_z \) and send a communication request message:

\[
< PID_z, T, CReq, SIG_z(ID_z||T), AuSIG_z(PID_z||T), ID_{RSU} >
\]

where \( PID_z \) is the Pseudo-ID of vehicle \( Z \), \( T \) is the current time, \( CReq \) is the communication request message, \( SIG_z(ID_z||T) \) is the IBS (generated by vehicle \( Z \) using its ID, and the current time), and \( AuSIG_z(PID_z||T) \) is the authenticated signature of vehicle \( Z \). The Pseudo-ID of vehicle \( Z \) (PIDz) and the current time value are concatenated, and the resulting value is encrypted using the secret key \( (AuK) \), which results in \( AuSIG_z(PID_z||T) \). \( ID_{RSU} \) is the ID of the RSU which has authenticated the vehicle. Each vehicle that receives the communication request message of vehicle \( Z \) must verify \( PID_z \) in its authentication table, and it must verify also the authentication signature (by decrypting the signature using \( AuK \) and verifying the Pseudo-ID and the time).

Algorithm 3 V2I authentication Algorithm.

\[ V_z \text{ calculates } \Sigma G_z(PID_z||T). \]
\[ V_z \text{ sends } < ID_{RSU}, PID_z, T, \Sigma G_z(PID_z||T) > \text{ to RSU.} \]
\[ \text{RSU verifies:} \]
\[ \text{if } V_z \text{ is valid} \]
\[ \text{Broadcast} < ID_{RSU}, T, PID_z, \Sigma G_z(PID_z||T), AuM, (AuK)_{SK_z} > \]
\[ V_i \text{ Save } PID_z \text{ and } \Sigma G_z(PID_z||T) \text{ in authentication table.} \]
\[ \text{else} \]
\[ \text{RSU runs System Revocation.} \]

In the case when vehicle \( Z \) wants to communicate with a vehicle that exists in the range of another RSU (described in Figure 4), the vehicle must send a verification message (VER) to the nearest RSU which contains:

\[
< PID_z, T, ID_{RSU}, \text{VER}, \Sigma G_z(PID_z||T), AuSIG_z(PID_z||T) >
\]

where \( PID_z \) is the Pseudo-ID of vehicle \( Z \), \( T \) is the current time, \( \text{VER} \) is the verification message, and \( \Sigma G_z(PID_z||T) \) is the IBS generated by vehicle \( Z \) (using its ID, the current time and its secret key). \( ID_{RSU} \) is the ID of the RSU that has authenticated the vehicle \( Z \). \( AuSIG_z(PID_z||T) \) is the authenticated signature of vehicle \( Z \) (using \( PID_z \), the current time, and the authentication key).

When a vehicle repeatedly moves into the same RSU scope, it does not need to authenticate itself with a vehicle that has already communicated with it. It needs only to authenticate itself with a new vehicle because it does not know if it is a legitimate vehicle or not.
The RSU must retransmit the verification message to the other RSU (the RSU which has the vehicle Z in its transmission range). The other RSU verifies in its authentication table and sends the result to the RSU; then, it will retransmit the result to the vehicle to decide to accept the communication or not.

**Algorithm 4 V2V authentication Algorithm**

\[ V_2 \text{ sends } <PID_2, T, CReq, SIG_z(ID_z||T), AuSIG_z(PID_z||T), ID_{RSU} > \text{ to } V_x \]

\( if \ V_2 \text{ and } V_x \text{ registered at the same RSU} \)

\( V_x \text{ verifies in its authentication table:} \)

\( if \ \text{Valid} \)

\( V_x \text{ verifies the authentication signature:} \)

\( if \ \text{correct} \)

\( \text{Reply} = \text{accept} \)

\( else \)

\( \text{Reply} = \text{Refuse} \)

\( \text{Runs System revocation algorithm} \)

\( else \)

\( \text{Reply} = \text{Refuse} \)

\( \text{Runs System revocation algorithm} \)

\( else \)

\( V_x \text{ sends } <PID_z, T, ID_{RSU}, \text{VER}, SIG_z(PID_z||T) > \text{ to RSU2.} \)

\( \text{RSU2 transmits verification request to RSU.} \)

\( \text{RSU sends Reply and Auk to RSU2.} \)

\( \text{RSU2 sends Reply and Auk encrypted by ‘SK_x’ to } V_x. \)

\( V_x \text{ sends reply to } V_2. \)
Figure 4. V2V authentication description (case2).

4.4. Cryptography

A - V2I Communication: After the distribution of SK in a secure manner, we use symmetric cryptography for the transmission of messages. Vehicles must encrypt the message using their secret key (SK) to ensure confidentiality and the security of information. The RSU then decrypts the message and transmits it to the cloud via a secure channel. When the cloud sends the answer, the RSU must also encrypt the message using the secret key SK of the vehicle. In this way, we ensure secure communication between cloud and vehicles. V2I communication process is described in Algorithm 5.

B - V2V Communication: Each vehicle in an RSU’s transmission range must authenticate itself on the RSU, and, when the RSU authenticates the vehicle, it will send an authentication message with authentication key (AuK). All vehicles authenticated on the same RSU will obtain the same authentication key (AuK). So, we can provide a secure communication session with vehicles authenticated by the same RSU. Vehicles authenticated by different RSU (RSU2) can obtain the (AuK) from its nearby RSU (RSU2) and provide a secure communication session with other vehicles. Algorithm 6 shows the V2V communication process.

Vehicles that want to communicate with other vehicles must encrypt the message using the authentication key (AuK), and only authenticated vehicles can decrypt the message.
Algorithm 5 V2I Communication Algorithm.

1. $V_z$ sends $[message]_{SK_z}$ to RSU.
2. RSU decrypts the message.
3. RSU transmits the message to the cloud.
4. Cloud sends a reply to RSU.
5. RSU send $[reply]_{SK_z}$ to $V_z$.

Algorithm 6 V2V Communication Algorithm.

1. $V_z$ sends $[message]_{AuK}$ to $V_x$.
2. If $V_z$ authenticated by the same RSU
   - $V_z$ decrypts the message.
   - $V_z$ sends $[reply]_{AuK}$ to $V_x$.
3. Else
   - $V_x$ obtains the AuK from the nearby RSU.
   - $V_x$ decrypts the message.
   - $V_x$ sends $[reply]_{AuK}$ to $V_z$.

We use CRAs (Cloud-based Revocation Authorities) to revoke misbehaving vehicles.

To summarize the proposed scheme, we give an application scenario of our model as shown in Figure A1 of the Appendix A. When a vehicle “Z” ($V_z$) starts its trip, it must authenticate itself at the nearest RSU by sending an RR message. The RSU then forwards this vehicle’s information to the cloud for key generation. The cloud uses Equation (1) to generate the Public Key (PK), Equation (2) to generate the Master Key (MK), Equation (3) to generate the Attribute Key ($AK_z$), and Equation (4) to generate the Secret Key ($SK_z$). After that, it will transmit the $AK_z$ and the $SK_z$ to the RSU. The RSU then uses the encrypt function to create the ciphertext using Equation (5) and broadcasting it. Only $V_z$ can decrypt the message because it has the set of attributes $S$. $V_z$ uses the Equations (7)–(12) to decrypt the ciphertext. After receiving the $SK_z$, $V_z$ can calculate and send $PID_z$ and $SIG_z$ to the RSU to be authenticated. $V_z$ can establish a secure communication with RSU using a symmetric encryption by encrypting messages with $SK_z$. If it wants to communicate with other vehicles in the vicinity, $V_z$, for example, $V_z$ needs to authenticate itself at $V_x$ by generating $AuSIG_z$. They can establish a secure communication using symmetric encryption by encrypting messages with AuK.

5. Security Analysis and Performance Evaluation

We provide an analysis of GUARANTY algorithm in terms of security analysis, computation delay, and transmission overhead.

5.1. Security Analysis

We analyze and discuss the security of the proposed protocol based on the security aims. Table 3 shows the comparison of GUARANTY scheme and other protocols in terms of security features.
Table 3. Comparison of security features.

| Features                | [7] | [8] | [9] | [10] | [11] | [12] | [13] | [14] | [16] | [17] | [18] | [19] | [20] | GUARANTY |
|-------------------------|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|---------|
| Authentication          | √   | ×   | √   | ×    | ×    | ×    | √    | ×    | ×    | ×    | ×    | ×    | ×    | √       |
| Confidentiality        | ×   | √   | ×   |    | ×    | √    | ×    |    | ×    | ×    | ×    | ×    | ×    | √       |
| Privacy                | √   | √   | √   |    | √    | √    |    | √    | ×    | √    | ×    | √    | √    | √       |
| Secure key distribution| ×   | ×   | ×   |    | ×    | ×    | ×    |    | ×    | ×    | ×    | ×    | √    | √       |
| Access control         | ×   | ×   | ×   |    | ×    | ×    | ×    |    | √    | ×    | ×    | ×    | √    | √       |
| Impersonation attack resistance | √   | ×   | √   | √    | ×    |    |    | √    |    | ×    | √    | √    | √    | √       |
| MIM attack resistance  |    | ×   | ×   |    | ×    |    |    | √    |    | √    | ×    | √    | √    | √       |
| Sybil attack resistance|    |    | ×   |    |    |    |    |    |    |    |    |    |    | √       |
| Eavesdropping attack resistance | √   |    |    |    |    |    |    |    |    |    |    |    |    | √       |
| Tracking attack resistance |    |    |    |    |    |    |    |    |    |    |    |    |    | √       |

Lemma 1. Our proposed scheme provides identity privacy-preserving for vehicles and prevents tracking attacks.

Proof. An attacker cannot obtain the real identity $ID_i$ of a vehicle ‘$i$’ throughout GUARANTY algorithm. It encrypts its real identifier using its secret key to generate its Pseudo-ID ($PID_i = (ID_i)_{SK_i}$) which will be used instead of its real $ID_i$ in their communication. Any other vehicle can access the real $ID_i$ because the $SK_i$ is distributed in a secure manner. In the registration step, vehicle sends its real $ID_i$ to the RSU encrypted by the public key of the RSU ($ID_i|PK_{RSU}$). Only the RSU can access to the real $ID_i$. □

Lemma 2. Our proposed algorithm ensures the secure distribution of keys between vehicles and RSU.

Proof. GUARANTY protocol ensures that keys are distributed in a secure manner between vehicles and the RSU, and any other vehicle can access the keys. During the registration step, the cloud generates two keys for each registered vehicle $AK_i$ and $SK_i$. The $AK_i$ is an attribute key calculated using the attributes transmitted by the vehicle ‘$i$’ and the corresponding chassis number ($CN_i$) registered in the cloud which can be used by only the vehicle that satisfies the set of attributes. $SK_i$ is a secret key of vehicle ‘$i$’ which must be kept secret by the vehicle $V_i$. In our proposed scheme, the RSU encrypts the $SK_i$ by the $AK_i$, and only the $V_i$ which satisfies the set of attributes can decrypt it. Among the attributes transmitted by the vehicles, each vehicle sends its identifier with its direction, position … So, only one vehicle which has the corresponding identifier ($ID_i$) and the corresponding chassis number ($CN_i$) can access to the $SK_i$. □

Lemma 3. Vehicles using our proposed protocol can determine whether the vehicle trying to establish a communication is authenticated or not.

Proof. GUARANTY protocol provides the authentication of vehicles using IBS signature. Each vehicle ‘$i$’ must authenticate itself (by an RSU) by sending $< SIG_i(PID_i||T) >$. The concerned RSU broadcasts an authentication message $< PID_i, AuM_i, (AuK)_{SK_i} >$ if $V_i$ is authenticated. Vehicles in the vicinity will receive the authentication message of vehicle $i$.

When $V_i$ sends a communication request to vehicles in the vicinity $< PID_i, AuSIG_i(PID_i||T) >$, they verify in their authentication table the $PID_i$ and the $AuSIG_i$
to decide to accept communicating or not. The communication between vehicles can be established only with vehicles authenticated by the RSU.

In the case where the vehicle is not authenticated by the RSU, it is defined as a malicious vehicle and it is revoked.

**Lemma 4.** GUARANTY algorithm can detect whether the RSU is authenticated and, in fact, what it declared itself to be.

**Proof.** Our algorithm can detect malicious entities that use the RSU’s information to take vehicles’ data. A malicious vehicle can:

A. Take the RSU’s information and declare itself that it is the RSU: In this case, the malicious vehicle cannot decrypt the message (vehicles’ attributes) because it does not have the private key of RSU ($PrK_{RSU}$). The vehicle waits for its $SK_i$ for $\Delta$ time; then, it will define the RSU as a malicious node after this time, and it will run a System Revocation.

B. Take the RSU’s identifier and generate its private and public keys: In this case, the malicious vehicle can access to the vehicles’ attributes but:

- It cannot access to the cloud for keys generation because it does not have the $VN_{RSU}$ of RSU. If it tries to access the cloud, it will be detected as a malicious node, and then its access will be revoked.

- When it generates false keys ($AK_i, SK_i$), $V_i$ can detect that $AK_i$ is wrong because the $AK_i$ does not include the real $CN_i$ due to cloud access failure of the malicious node to recover the $CN_i$ (RSU does not have a verification number). So, the malicious RSU cannot generate $AK_i$ and cannot use the vehicles’ attributes to access the $SK_i$ (RSU does not have the $CN_i$). When vehicles receive a wrong $AK_i$ from a node, they will define it as a malicious node and run the System Revocation.

**Lemma 5.** The messages transmitted between vehicles and the ones between vehicles and RSUs are confidential. Vehicles that can access these messages are authenticated by the RSUs.

**Proof.** GUARANTY algorithm ensures the confidentiality of messages and the access control of vehicles. The use of the CP-ABE algorithm for key distribution can define legitimate vehicles that satisfy the access policy to take the secret key. In this way, the access control to the keys is ensured.

V2I communication: Since $SK_i$ is distributed in secure manner between RSU and $V_i$, messages are encrypted using the $SK_i$, so only entities that know the $SK_i$ can decrypt the messages.

V2V communication: Vehicles authenticated by the RSU during the authentication step have an authentication key ($AuK$) which is transmitted in secure manner ($AuK$). This key is used to encrypt messages between vehicles.

**Lemma 6.** GUARANTY protocol resolves different type of attacks, such as attacks based on authentication, privacy, confidentiality, and non-repudiation.

**Proof.** GUARANTY protocol resolves the following type of attacks:

- Impersonate attack: is an attack that harms the authentication of vehicles. In this kind of attack, an attacker impersonates a good vehicle to receive its messages, and get privileges not granted to it. It can also do malicious things and declare that a legitimate vehicle did them. In our proposed scheme, malicious vehicles can use the Pseudo-ID ($PID_i$) and the signature $SIG_i(PID_i||T)$ of the legitimate vehicle to communicate with the other vehicles. So, it will send a communication request: $<PID_i, T, CReq, SIG_i(PID_i||T), SIGAu_i(PID_i||T), ID_{RSU}>$ to vehicle $z$. Vehicle $z$ must verify the
AuSIG_i(PID_i||T) of vehicle ‘i’ to detect that its signature is not satisfied (it does not have the real AuK) and declares it as a malicious vehicle.

- **Man in the Middle (MiM) attack**: It is an attack that harms the confidentiality and the integrity of vehicles. In this case, a malicious vehicle listens to the communication between two vehicles. GUARANTY algorithm ensures the confidentiality of:
  - Messages transmitted between vehicles, by the encryption of messages using the authentication key (AuK).
  - Messages transmitted between vehicles and RSU, by the encryption of messages using the vehicle’s secret key (SK).
  - Attributes transmitted during the registration step, by the encryption of attributes using the public key of RSU (PK_RSU).

- **Sybil attack**: this attack harms the authentication of vehicles. In this case, an attacker generates multiple fake vehicles on the road with different identities and tries to authenticate them at the same time. Our approach can detect this attack:
  - If the malicious vehicle communicates with RSU to register and authenticate itself. The RSU detects that it is a malicious vehicle by verifying the location-related attributes using location proof methodologies [29].
  - If the malicious vehicle ‘mal’ generates its Pseudo-ID (PID_mal) and its signature SIG_mal and establishes a communication directly with other vehicles by sending a communication request <PID_mal, T, CReq, SIG_mal (PID_mal||T), AuSIG_mal(PID_mal||T), ID_RSU>). They can detect that it is a malicious vehicle by verifying the PID_mal and SIG_mal (PID_mal||T) in their authentication table.

- **Eavesdropping attack**: The proposed algorithm authenticates the vehicle properly, which does not allow the intruder to access the information circulating in the network during the transmission process. Furthermore, the symmetric session keys (Sk, AuK) is used to encrypt the collected data. This approach keeps the data confidential and stops the attacker from accessing the message.

Table 3 presents a comparison of the security features of our proposed scheme and other schemes available in the literature [7–14,16–20]. We can conclude that our proposed scheme is robust against the considered security issues. GARANTY resists impersonate attack, Sybil attack, MiM attack, and eavesdropping attack. It provides vehicles authentication, confidentiality, access control, and privacy.

### 5.2. Specification of GUARANTY Scheme Using AVISPA Tool

AVISPA (a popular tool) checks whether the security protocol is safe against active and passive attacks. This simulator supports High Level Protocol Specification Language (HLPSL) [31].

Next, we describe the role of each participant (vehicle Z, the RSU, the session, the goal, and the environment) of the proposed scheme; Figure A2 describes the role of vehicle Z in HLPSL. During the execution of the registration phase, vehicle Z sends the registration request SND (T.S.D.L.ID_PK) to RSU through a secure channel using the RSU’s public key and SND () operation. The type declaration channel (dy) dictates that the channel follows Dolev and Yao threat model [32].

In transition 2, vehicle Z receives information RCV (PID’,{SK’}_AK) from RSU securely using the RCV () operation and AK symmetric key. After that, vehicle Z creates timestamp T’ using new () operation, PID using its ID encrypted by SK. Then, it sends (PID’,ID’) which is the signature SIG_z to RSU through secure network with its Pseudo-ID via a public channel. The statement witness (R, Z, z_rsu_nb, SK) is used by the vehicle Z to ask the RSU to authenticate it using the SK.

In transition 3, vehicle Z receives RCV (PID’,{AuK}_SK) its Pseudo-ID through a public channel and the authentication key through a secure channel using SK symmetric_key. Then, it sends (PID’,T’)_AuK which is its authentication signature (AuSIG) to
vehicle V. The statement witness \((V, Z, z_{\text{rsu_nv}}, \text{PID}')\) is used by the vehicle Z to ask the vehicle V to authenticate it using the PID’. After key exchange and vehicles authentication, vehicle Z sends message M1 to RSU encrypted with SK and message M2 to vehicle V encrypted with AuK. The declaration secret\((M1 , n1, \{Z, R\})\) dictates that the message M1 is only known to the vehicle Z and RSU, and the declaration secret\((M2 , n2, \{Z, V\})\) dictates that the message M2 is only known to the vehicle Z and vehicle V.

In Figure A3, we present RSU’s role in HLPSL, where RSU first receives RCV (T.S.D.L.ID) from vehicle Z. Then, RSU sends the SK encrypted with AK. In transition 2, RSU receives vehicle Z’s Pseudo-ID \(\{T'.ID\}_SK\), and the declaration request \((Z, R, z_{\text{rsu_nb}}, SK)\) is used by the RSU to check whether the vehicle Z uses the same key “SK” that it sent at transition 1. During the last transition, it sends the authentication key \(\{AuK\}_SK\) via a secure channel using SK.

Figure A4 represents the role of vehicle V, with which vehicle Z wants to establish a communication. Vehicle V receives the authentication signature of vehicle Z, verifies its authentication using the statement request, and establishes a communication with it using the authentication key to ensure secure communication.

In Figure 5, the role of session, goal and environment are described in HLPSL. The top-level role contains global constants and a composition of sessions, where the intruder (i) participates in the execution as a legitimate node.

```
role session(Z, R, V: agent,
    SK, AuK, SSK: symmetric_key,
    PK: public_key) def=

    local SZ, RZ, SR, RR, SV, RV: channel (dy)

    composition

    \[\text{vehicleZ}(Z,R,V,SK,AuK,PK,SZ,RZ)\]
    \[\land \text{rsu}(Z,R,SK,AuK,SSK,PK,SR,RR)\]
    \[\land \text{vehicleV}(Z,R,SSK,AuK,SV,RV)\]

    end role

role environment() def=

    const z, r, v : agent,
    sk, ki, auk, ssk : symmetric_key;
    pk: public_key;
    nb, na, n1, n2, z_{\text{rsu_nb}}, z_{\text{rsu_nv}}: protocol_id

    intruder_knowledge = \{(z, r, v, ki, pk, inv(ki))\}

    composition

    session(z,r,v,sk,auk,ssk,pk)
    \land session(z,i,v,ki,ssk,pk)
    \land session(z,i,v,ki,auk,ssk,pk)
    \land session(i,r,v,ki,auk,ssk,PK)

    end role

goal

security_of_nb
authentication_on_z_{\text{rsu_nb}}
authentication_on_z_{\text{rsu_nv}}
security_of_na
security_of_n1
security_of_n2

end goal

environment()
```

Figure 5. Proposed protocol session specification in High Level Protocol Specification Language (HLPSL).
The proposed protocol uses the current version (2006-02-2013) of OFMC model, which supports secrecy goals and authentication objectives. However, the proposed protocol uses five secrecy goals and two authentications objectives in operation:

1. Secrecy_of_nb indicates that SK is kept secret to vehicle Z and RSU.
2. Secrecy_of_na indicates that AuK is only known by RSU, vehicle Z, and vehicle V.
3. Secrecy_of_n1 signifies that the message M1 is kept secret between vehicle Z and RSU.
4. Secrecy_of_n2 signifies that the message M2 is only known by vehicle Z and vehicle V.
5. Authentication_on_z_rsu_nb signifies that RSU creates a secure key SK, and vehicle Z obtains it securely via message; if RSU receives the same value of SK from vehicle Z, RSU then authenticates vehicle Z.
6. Authentication_on_z_rsu_nv signifies that RSU creates an authentication key AuK and sends it securely to an authenticated vehicle V. Vehicle Z receives the authentication key securely via message; if vehicle V receives the same value of AuK from vehicle V, vehicle V then authenticates vehicle Z.

Figure 6 represents the result obtained after executing the HLPSL code in AVISPA. As a result, we conclude that the protocol is “SAFE” under OFMC.

![Figure 6](image.png)

5.3. Computational Complexity Analysis

We evaluate the theoretical efficiency of GUARANTY protocol in terms of computational complexity. In our algorithm, the pairing computations and exponentiation are the most expensive operations. To calculate GUARANTY scheme’s complexity, we need to calculate the number of pairing and exponentiation operations performed in the algorithm. Table 4 summarizes our algorithm’s computation cost, where $C_e$ represents the computation cost of exponential operation, and $C_p$ represents the computation cost of pairing operation.

| Operation            | Complexity          |
|----------------------|---------------------|
| Setup                | $O(1)$              |
| Key Generation       | $O((|S| + 3)C_e)$   |
| Encryption           | $O(2C_p)$           |
| Decryption           | $O((2 + l)C_p)$     |

Table 4. GUARANTY Computational complexity.
When the system is setup, the cloud selects a bilinear group and some random numbers to calculate PK and MK.

Key-Generation: The cloud generates in this phase the SK and AK using MK and the set of attributes. The computation cost of this phase depends on the number of the attribute in the set \(| S |\) for the vehicle.

Encryption: This operation requires two exponentiations to calculate \(C_i\) and \(D_i\) for each row in the ciphertext policy. The computation cost of this phase grows linearly with the size of access structure’s matrix \(M (l*n)\).

Decryption: During the decryption phase, the number of pairing computations grows linearly with the number of attributes required for decryption.

5.4. Performance Evaluation

To study the performance of our algorithm, we simulated GUARANTY scheme with parameters illustrated in Table 5. We generate these scenarios using VanetMobiSim vehicles’ mobility generator.

We compare the computation delay of our algorithm to two other works [16,17] that use the CP-ABE algorithm to ensure data confidentiality and vehicles’ access control. Table 6 summarizes the obtained results.

Table 5. Parameters.

| Topology (m\(^2\))         | 5000 \(\times\) 5000 |
|---------------------------|-----------------------|
| Number of nodes           | 100                   |
| Number of RSUs            | 4                     |
| Communication range of nodes (m) | 300               |
| Speed (m/s)               | 30                    |
| MAC layer                 | MAC 802_11p           |
| Time(s)                   | 300 s                 |
| Mobility model            | Random Following IDM_ LC |

Table 6. Computation delay comparison.

| Algorithm | Key-Management | Authentication | Communication (1 Message) | Communication (N Messages) |
|-----------|----------------|----------------|---------------------------|---------------------------|
| GUARANTY  | 2.003035 (s)   | 1.205304 (s)  | 0.0912 (s)                | N \(\times\) 0.0912 (s)  |
| [17]      | -              | 1.0026 (s)    | 1.890035 (s)              | N \(\times\) 1.890035 (s) |
| [16]      | -              | -             | 2.300527 (s)              | N \(\times\) 2.300527 (s) |

The work in Reference [16] does not ensure the key-Management and the authentication phases, and all the calculations are also not delegated to the cloud; for this, it requires more computation delay to encrypt and decrypt the messages in the communication phase than GUARANTY and Reference [17]’s algorithms.

The work in Reference [17] ensures only the V2R authentication; thus, it requires less computation time than the GUARANTY algorithm, which ensures both (V2R and V2V) authentication. Concerning the Communication delay, GUARANTY scheme carries out the Key-Management step at the start of the communication; then, it uses a symmetry encryption for the N messages exchanged between vehicles or vehicles and RSU because the secret key is distributed in a secure manner. However, authors in Reference [17] did not provide a Key-Management phase, and it needs to carry out all the computations for each message, which requires a high computation delay by comparison to GUARANTY algorithm.

We provide a performance evaluation of GUARANTY algorithm in terms of reception rate and computation delay. We consider various scenarios by changing the number of vehicles in the network. Specifically, we test for 100, 200, 300, and 400 nodes.

Figure 7 illustrates the computation delay of each step of our algorithm based on the number of vehicles.
We notice that the delay increases when the number of vehicles increases. This is due to the number of vehicles that calculate their authentication and their key distribution. The key-management step contributes almost 59% of the computation delay to generate vehicles’ keys and distribute them in a secure manner (using CP-ABE algorithm). The computation delay in authentication step requires almost 40% to authenticate vehicles and to distribute authentication key.

GUARANTY scheme uses these steps only at the start of vehicles’ trip to generate keys and authenticate vehicles. However, the communication phase is used for the whole trip. This phase depends only on messages encryption and decryption cost. The computation delay of messages transmission and routing is not discussed in this phase. This step needs almost 1% of the computation delay to encrypt/decrypt messages with symmetric keys.

The Packet Delivery Ratio (PDR) in a network is the ration of the total number of received messages to that of the total number of transmitted messages. PDR determines the number of messages received at the destination. In our proposed scheme, we calculate the PDR (%) using the Formula (13):

$$PDR = \left( \frac{\sum Msg_r}{\sum Msg_s} \right) \times 100.$$  

(13)

Figure 8 shows the comparison of PDR according to the number of vehicles. The results show that the PDR is inversely proportional to the number of vehicles. This is due to the number of vehicles that need key generation, authentication, messages encryption, and decryption, which require a significant computation time. This latter and the high vehicle’s mobility lead to the loss of messages that represents 9% of the transmitted messages. The reception rate for 100 nodes is 95%, 200 nodes is 89%, 300 nodes is 85%, and 400 nodes is approximately 80%.

Figure 8. Reception rate according to the change of vehicles’ number.
6. Conclusions

In this paper, we proposed a lightweight secure Authentication and key (GUARANTY) distribution scheme for vehicular cloud computing. We ensured vehicles’ authentication using the IBS mechanism, conductors’ privacy using Pseudo-ID instead of using vehicles’ real ID, secure keys’ distribution and access control using the CP-ABE algorithm. GUARANTY algorithm secures the vehicles’ communication with a minimum computation cost. To ensure our scheme’s security against insiders’ and outsiders’ attacks, we simulated it using the AVISPA tool. We compared the computation delay of our proposed scheme to two other approaches [16,17] that use CP-ABE algorithm. The simulation results show that GUARANTY algorithm uses a minimum computation delay during the communication step, which is 1% of the computation time, compared to other algorithms that need to recalculate the ciphertext in each communication. The calculation of the ciphertext using the CP-ABE algorithm requires a high computation time (in our case, the CP-ABE algorithm calculation requires 59% of the computation time). In order to evaluate our algorithm’s performance in terms of computation delay and reception rate, we tested several scenarios generated by ‘VANETMOBISIM’ mobility generator. The simulation results show that our algorithm achieves convincing results. In the future, we plan to integrate blockchain technology in our proposed solution to guarantee the integrity of messages.

Author Contributions: H.G. conceived the idea, performed the analysis and wrote the first draft of the paper; S.H. reviewed, edited, and finalized the writeup of the paper; Z.A. discussed the idea, assisted with the experiment design and revised the writeup of the paper; A.M.G. discussed the idea. All authors have read and agreed to the submitted version of the manuscript.

Funding: This work received no funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available upon request.

Acknowledgments: This research work is supported in part by PHC-Tassili (grant 18MDU114).

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

Abbreviations
The following abbreviations are used in this paper:

- $Sp_z$: Speed of vehicle Z
- $D_z$: Direction of vehicle Z
- $L_z$: Localization of vehicle Z
- $ID_{RSU}$: Identifier of RSU
- $VN_{RSU}$: Verification Number of RSU
- $SK_z$: Secret key of vehicle Z
- $AK_z$: Attribute key of vehicle Z
- $ID_z$: Identifier of vehicle Z
- $PID_z$: Pseudo-ID of vehicle Z
- $SIG_z$: Signature of vehicle Z
- $AuSIG_z$: Authentication Signature of vehicle Z
- $CReq$: Communication request
- $AuK$: Authentication Key
- $Msg$: Message
- $AuM$: Authentication Message
- $VER$: Verification Message
- $RR$: Registration Request
- $PK_{RSU}$: Public Key of RSU
- $MK$: Master Key
- $CT$: Ciphertext
- $CN_z$: Chassis Number of vehicle Z
Appendix A

Figure A1. Summary of GUARANTY algorithm.
role vehicleZ (Z, R, V: agent,
   SK, AuK: symmetric_key,
   PK, public_key,
   SND, RCV: channel (dy))
played_by Z def=
local State : nat,
   S,D,L,T: text,
   ID, PID: text,
   AK, message,
   M1, M2: message
init State := 0
transition
0. State := 0 \land RCV(start) \Rightarrow
   State' := 2 \land SND((T.S.D.L.ID) \_PK)
1. State := 2 \land RCV({\{SK'\}}_AK) \Rightarrow
   State' := 4 \land T' := newO
   \land PID := \{ID\}_SK
   \land SND(PID', (T' ID) _SK)
   \land witness(R.Z.z_rsu.nb, SK)
2. State := 4 \land RCV(PID'.{\{AuK\}}_SK) \Rightarrow
   State' := 6 \land T' := newO
   \land SND((PID'.T')_AuK)
   \land witness(Z.V.z_rsu.nb,AuK)
   \land SND((M1) _AuK)
   \land secret(M1.n1,(Z,R))
   \land SND((M2) _AuK)
   \land secret(M2.n2,(Z,V))
end role

Figure A2. HLPSL specification of vehicle Z of the proposed protocol.

role rsu (Z, R, V: agent,
   SK, AuK, SSK: symmetric_key,
   PK, public_key,
   SND, RCV: channel (dy))
played_by R def=
local State : nat,
   S,D,L,T: text,
   ID, PID: text,
   AK, M1: message
init State := 1
transition
1. State := 1 \land RCV((T.S.D.L.ID) \_PK) \Rightarrow
   State' := 3 \land SK' := newO \land SND({SK'\}_AK)
   \land secret(SK',nb,(Z,R))
   \land SND({AuK\} _SSK)
2. State := 3 \land RCV(PID'.(T.ID) _SK) \Rightarrow
   State' := 5 \land request(Z.R.z_rsu.nb,SK)
   \land SND(PID'.{AuK} _SK)
   \land secret(AuK.na,(Z,R,V))
   \land SND(PID')
3. State := 5 \land RCV((M1) _AuK) \Rightarrow State' := 7
end role

Figure A3. HLPSL specification of RSU of the proposed protocol.
Figure A4. HLPSL specification of vehicle V of the proposed protocol.

```
role vehicleV(Z, R, V: agent,
    AuK, SSK: symmetric_key,
    SND, RCV: channel (dy))
played_by V def=

local State : nat,
    PID,T: text,
    M2: message

init State := 1

transition

1. State = 1 ∧ RCV([AuK]_SSK) ∧ RCV(PID') ∧ RCV((PID,T')_AuK) =>
   State' := 3 ∧ request(Z,V,z_rsa_m,AuK)

2. State = 3 ∧ RCV((M2}_AuK) => State := 5
```

References

1. De Souza, P.R.R.; Matteussi, K.J.; Veith, A.D.S.; Zanchetta, B.F.; Leithardt, V.R.; Murciego, Â.L.; De Freitas, E.P.; Dos Anjos, J.C.; Geyer, C.F. Boosting Big Data Streaming Applications in Clouds With BurstFlow. IEEE Access 2020, 8, 219124–219136. doi:10.1109/ACCESS.2020.3042739. [CrossRef]
2. Trubia, S.; Severino, A.; Curto, S.; Arena, F.; Pau, G. Smart Roads: An Overview of What Future Mobility Will Look Like. Infrastructures 2020, 5, 107. [CrossRef]
3. Zhu, J.; Xu, W. Real-Time Data Filling and Automatic Retrieval Algorithm of Road Traffic Based on Deep-Learning Method. Symmetry 2021, 13, 1. [CrossRef]
4. Goumidi, H.; Aliouat, Z.; Harous, S. Vehicular cloud computing security: A survey. Arab. J. Sci. Eng. 2020, 45, 2473–2499. [CrossRef]
5. Hasrouny, H.; Sambat, A.E.; Bassil, C.; Laouiti, A. VANet security challenges and solutions: A survey. Veh. Commun. 2017, 7, 7–20. [CrossRef]
6. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 20–23 May 2007; pp. 321–334. [CrossRef]
7. Nkenyereye, L.; Park, Y.; Rhee, K.H. Secure vehicle traffic data dissemination and analysis protocol in vehicular cloud computing. J. Supercomput. 2018, 74, 1024–1044. [CrossRef]
8. Wan, C.; Zhang, J. Efficient identity-based data transmission for VANET. J. Ambient Intell. Humaniz. Comput. 2018, 9, 1861–1871. [CrossRef]
9. Lu, Z.; Liu, W.; Wang, Q.; Qu, G.; Liu, Z. A privacy-preserving trust model based on blockchain for vanets. IEEE Access 2018, 6, 45655–45664. [CrossRef]
10. Arora, A.; Yadav, S.K. Block chain based security mechanism for internet of vehicles (iov). In Proceedings of the 3rd International Conference on Internet of Things and Connected Technologies, (ICIoTCT), Jaipur, India, 26–27 March 2018; pp. 267–272. [CrossRef]
11. Feng, X.; Wang, L. PAU: Privacy Assessment method with Uncertainty consideration for cloud-based vehicular networks. Future Gener. Comput. Syst. 2019, 96, 368–375. [CrossRef]
12. SathyaNarayanan, P. A sensor enabled secure vehicular communication for emergency message dissemination using cloud services. Digit. Signal Process. 2019, 85, 10–16. [CrossRef]
13. Hussain, R.; Rezaeifar, Z.; Lee, Y.H.; Oh, H. Secure and privacy-aware traffic information as a service in VANET-based clouds. Pervasive Mob. Comput. 2015, 24, 194–209. [CrossRef]
14. Jenefa, J.; Anita, E.M. Secure Vehicular Communication Using ID Based Signature Scheme. Wirel. Pers. Commun. 2018, 98, 1383–1411. [CrossRef]
15. Huang, D.; Verma, M. ASPE: Attribute-based secure policy enforcement in vehicular ad hoc networks. *Ad Hoc Netw.* 2009, 7, 1526–1535. [CrossRef]

16. Liu, X.; Xia, Y.; Chen, W.; Xiang, Y.; Hassan, M.M.; Alelaiwi, A. SEMD: Secure and efficient message dissemination with policy enforcement in VANET. *J. Comput. Syst. Sci.* 2016, 82, 1316–1328. [CrossRef]

17. Safi, Q.G.K.; Luo, S.; Wei, C.; Pan, L.; Yan, G. Cloud-based security and privacy-aware information dissemination over ubiquitous VANETs. *Comput. Stand. Interfaces* 2018, 56, 107–115. [CrossRef]

18. Kudva, S.; Badsha, S.; Sengupta, S.; Khalil, I.; Zomaya, A. Towards secure and practical consensus for blockchain based VANET. *Inf. Sci.* 2021, 545, 170–187. [CrossRef]

19. Sestrem Ochôa, I.; Reis Quietinho Leithardt, V.; Calbusch, L.; De Paz Santana, J.F.; Delcio Parreira, W.; Oriel Seman, L.; Albenes Zeferino, C. Performance and Security Evaluation on a Blockchain Architecture for License Plate Recognition Systems. *Appl. Sci.* 2021, 11, 1255. [CrossRef]

20. Yan, X.; Gu, X.; Wang, J.; Wan, J.; Chen, L. A kind of event trust model for VANET based on statistical method. *Wirel. Pers. Commun.* 2021, 1–15. [CrossRef]

21. Green, M.; Hohenberger, S.; Waters, B. Outsourcing the decryption of abe ciphertexts. In Proceedings of the USENIX Security Symposium, Washington, DC, USA, 11–13 August 2011.

22. Parno, B.; Perrig, A. Challenges in securing vehicular networks. In Proceedings of the Workshop on hot topics in networks (HotNets-IV), New York, NY, USA, 14–15 November 2005; pp. 1–6.

23. Schneier, B. The Blowfish Encryption Algorithm. 2008. Available online: https://doi.org/http://www.schneier.journal.com/blowfish.html (accessed on 8 May 2012).

24. Liang, W.; Li, Z.; Zhang, H.; Wang, S.; Bie, R. Vehicular ad hoc networks: Architectures, research issues, methodologies, challenges, and trends. *Int. J. Distrib. Sens. Netw.* 2015, 11, 1–11. [CrossRef]

25. Beimel, A. Secret-sharing schemes: A survey. In *International Conference on Coding and Cryptology*, Oxford, UK, 12–15 December 2011; Springer: Berlin, Germany, 2011; pp. 11–46. [CrossRef]

26. Ostrovsky, R.; Sahai, A.; Waters, B. Attribute-based encryption with non-monotonic access structures. In Proceedings of the 14th ACM conference on Computer and communications security, Alexandria, VA, USA, 28–31 October 2007; pp. 195–203. [CrossRef]

27. Waters, B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *International Workshop on Public Key Cryptography*; Springer: Berlin, Germany, 2011; pp. 53–70. [CrossRef]

28. Shamir, A. Identity-based cryptosystems and signature schemes. In *Workshop on the Theory and Application of Cryptographic Techniques*; Springer: Berlin, Germany, 1984; pp. 47–53. [CrossRef]

29. Saroiu, S.; Wolman, A. Enabling new mobile applications with location proofs. In Proceedings of the 10th workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA, 23–24 February; pp. 1–3. [CrossRef]

30. Corless, R.M. On a generalized companion matrix pencil for matrix polynomials expressed in the Lagrange basis. In *Symbolic-Numeric Computation*; Springer: Berlin, Germany, 2007; pp. 1–15. [CrossRef]

31. Armando, A.; Basin, D.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuéllar, J.; Drielsma, P.H.; Héam, P.C.; Kouchnerenko, O.; Mantovani, J.; et al. The AVISPA tool for the automated validation of internet security protocols and applications. In *International Conference on Computer Aided Verification*; Springer: Berlin, Germany, 2005; pp. 281–285. [CrossRef]

32. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theor.* 1983, 29, 198–208. [CrossRef]