Design a Cloud Security Model in VANET Communication: Design and Architecture

Hatem M. Hamad
Department of Computer Engineering
Islamic University of Gaza
Palestine, Gaza Strip

Alaaeddin B. AlQazzaz
Department of Computer Engineering
Islamic University of Gaza
Palestine, Gaza Strip

ABSTRACT
During the last few years, Intelligent Transportation System (ITS) has been progressed at a rapid rate, which aimed to improve the transportation activities in the terms of the safety and efficiency. According to many issues with the traditional Vehicular Ad-Hoc Networks (VANET), some efforts are made to merge the VANET with the cloud technology. This work proposes the VANET based on the cloud (V2Cloud), and designs a security model framework that is hosted on the cloud to manage the security services, and provide a secure VANET communication between the different entities e.g. vehicles, authorities and etc. This security model framework is called VANET Security as a Service (VSaaS).

Our works will presented in a set of two papers. In this first one, it presents VSaaS design and architecture in order to show that the VSaaS fulfills the VANET's security requirements, and protects the VANET against the different types of attacks. The second paper will present the progress towards the implementation and the security analysis of the proposed architecture, along with the results of the performance of the security overhead for the secure Vehicle Information Messages (VIMs), which are sent by vehicles to the cloud as a coarse-grained information.

Keywords
VANET, Cloud, VSaaS, Certified Authority, Cryptography, Vehicle Information Messages, Traffic Information Messages, Authentication, Privacy, Security Overhead

1. INTRODUCTION
During the last few years, Intelligent Transportation System (ITS) has been progressed at a rapid rate, which aimed to improve the transportation activities in terms of safety and efficiency. In traditional way, Vehicle-to-Vehicle (V2V) communications and Vehicle-to-Infrastructure (V2I) communications are the main components of the ITS architecture. Communication between vehicles is often referred to Vehicular Ad-Hoc Networks (VANET) and it has many advantages as reducing the cars' accidents, minimizing the traffic jam, reducing the fuel consumption and emissions and etc [1].

In spite of all the efforts which had been made in recent years in the field of VANET and its security, researchers still depend directly on the existence of the Road Side Units (RSUs), and the local Certified Authorities (CAs) to provide VANET services including the security services. According to our survey on the previous related works, this dependency is facing many problems as,

- The availability issues: the RSUs which rarely exist in the highway roads and rural areas causes the services such as security services, which are available only in limited regions. Also, making these services available in different areas needs additional costs and efforts to deploy new RSUs, and connect both of the RSUs and the authorities into a single IP network.
- As to business viewpoint, RSUs deployment is costly.
- There is a need to a central management architecture to ensure the availability of the services when a vehicle moves from a region to another. Also, there is a need to mechanisms to make a trust between RSUs.
- These RSUs are not efficient when the number of nodes (vehicles) is very large. It consumes channel bandwidth and effects network performance.
- Highly computations are needed in the vehicles to analyze the collected data and produce fine human readable information.
- Traditional VANET is not suitable for the cheaper vehicles, which lack of proper hardware and sensors to participate into VANET and make its benefits.

A number of authors pointed out in their works that the allocated bandwidth exceeds far more than the requirements for the VANET safety applications [2]. Thus, the surplus bandwidth opens the doors for new opportunities along with the normal functionality of the VANET [3]. This gave a motivation to Professor Olariu and his colleagues to envision a paradigm shift from the traditional VANET to the Vehicular Cloud Computing (VCC) by merging the VANET with the Cloud Computing [4].

Cloud Computing has changed the computation and communication mindset by decoupling the computational assets from the physical infrastructure, and thereby enabling virtualization [5]. The main motivation of the Cloud Computing is to "exactly what you need and when you need" [6].

But till (February/2015), there was no solid architecture or general model for VCC as mentioned in [7]. Modern vehicles are equipped with permanent Internet by the 3G/4G cellular networks which make the cloud request available even when the RSUs do not exist. In addition to featuring like on-board computational, storage and sensing capabilities, which can be thought as a huge farm of computers, remain idle while the vehicles stay on the road. Moreover, as every vehicle which has Internet connections, it can automatically send the messages and the measurements to the cloud, which is controlled by authorities like police. It is worth to mention that the notation of Cloud Computing idea started from the fact of benefiting from it, by using it as an alternative of investing in infrastructure, business may find it useful to rent the infrastructure, and sometimes the needed software to run their applications. It decreases the number of RSUs as well as

giving the vehicles an access to Internet where there is no coverage signal of the RSUs [8].

In the 12th Annual IEEE Consumer Communication and Networking Communication Conference at 2015, Authors in [9] mentioned that the secure VANET cloud was a challenge task, and there was a need to a compatible security model which worked well with the different manufacturers in different countries to satisfy the security requirements in the VANET communication. The security and privacy challenges, which are faced by the standalone VANET and Cloud Computing, will remain unchanged even when the two technologies are merged to form VANET clouds [6].

However, messages (especially safety-messages), which are sent by vehicles in VANET, should be authenticated because the false or altered messages may lead to bad situations like accidents. It should be aware of many challenges in the VANET security such as privacy, because the drivers want to protect their identities from the others to prevent unauthorized tracking, but at the same time, it (the privacy) is in a conflict with other security attributes like authentication, which makes the design of VANET security model needs extra efforts. Moreover, the desired security model for the VANET should include authentication, authorization, confidentiality, integrity, non-repudiation, revocation and privacy [10].

2. CONTRIBUTIONS

This paper proposes VANET based on the cloud (V2Cloud), and designs a cloud security model framework, which is called VANET-Security as a Service (VSaaS), to manage the security services and provide a secure VANET communication between the different entities, e.g. vehicles, authorities and etc. Our objectives include:

- Proposing VANET that depends on cellular networks, which act as a gateway to the cloud to get the services which include security services.
- Proposing VANET-Security as a Service (VSaaS) model framework. VSaaS is responsible for:
    - Vehicles and authorities registration.
    - Key Management mechanisms, to generate keys for different entities and renew the keys when they become expired.
    - Authenticating the vehicles and their information messages, and authenticate the authorities that interacting with the VSaaS too.
    - Vehicle identity identification mechanism, to preserve the privacy and enable the traceability done only by the trusted authorities that have a permission to track vehicles.
    - Providing a security access list to manage the permissions among the different entities.
    - Providing a mechanism to revoke the misbehaved vehicle and the compromised authority.
    - Providing modules to process the Vehicle Information Messages (VIMs), which are sent by vehicles as coarse-information messages, and to construct fine-information messages, which called Traffic Information Messages (TIMs), they are disseminated to the vehicles based on their locations

The rest of this paper is organized as: Section 3 presents Related Work, Section 4 presents Security Requirements in VANET, Section 5 presents Proposed VSaaS, and finally, Section 6 concludes the paper.

3. RELATED WORK

3.1 VANET Security

Early papers proposed using pseudonyms to keep the privacy into consideration like [11]. Pseudonyms which are defined as, many short-lifetime certificates (private-public key pairs) installed on each vehicle by an authority, where these pseudonyms are used in one period and not be used again. This method protects the vehicle identity from being tracked by the unauthorized observers, but there is one major problem which is the link-ability of the pseudonyms. The attacker may identify the target vehicle by linking the previous pseudonym with the current one by the temporal or spatial locality. Also this approach has many other problems, for example, a large storage space is needed at each vehicle. Moreover, including the certificate in the safety message leads to larger message size and needs more computations to verify every certificate at the receiver side. In addition, the big number of certificates in the authority of all vehicles, causes a big overhead. For liability, the authority should store all these keys to identify the misbehaving vehicle. Moreover, the authority needs to search in a very huge number of keys and that costs a time.

The proposed approach in [12] was aimed to reduce the large number of pseudonyms which are preloaded on each vehicle. The approach reduced the number to a half, on the average. It depended on using two certificates: the encryption certificate and the signing certificate.

To solve the link-ability problem, some approaches like in [13] proposed a strategy called “hiding in crowd”. In this approach, the pseudonyms are updated regularly according to the spatial or temporal criteria. But, there are some situations that the link-ability is unavoidable in them. One of these situations is driving on a long road without junctions. In this case, the vehicle can be traced or linked to its group in spite of changing its pseudonyms.

Other approaches like in [14] and [15] try to solve the locality problem by using a random silent period among the changing of the pseudonyms. In the silent period, vehicle does not transmit any message. The period duration should be random and short. It is hard to link between vehicles before and after the silent period. In this approach, the vehicles must change their pseudonyms in adjacent times, but it is not practical, because of the need of broadcasting the safety messages regularly.

Another solution in [16] and [17] proposed vehicles belong to regions called mix-zones. Each vehicle in the same mix-zone changes its pseudonym at the same time. This solution decreases the linkage problem, but it depends on the number of vehicles in each mix-zone.

Some researchers employ the Identity-Based Cryptography (IBC), where the certificates are not needed for the authentication. IBC was proposed in 1984 by Shamir [18]. IBC differs from the public key infrastructure (PKI). In 2001, Boneh and Franklin [19] introduced the first functional and efficient identity-based encryption scheme that was based on bilinear pairings property of the elliptic curve.

[20] and [21] proposed an approach based on identity-based cryptography (IBC), which provides the authentication, non-repudiation and the privacy. In this approach, each pseudonym, which is an anonymous identity, is generated by the RSU. The approach enables a single authority to reveal the

identity. However, their approach is very dependent on the RSUs which may not be reachable or very busy in some cases. Other approaches were proposed like in [22], [23], they try to avoid the disadvantages of the previous approach.

Another architecture is to use a group signature approach as in [24] and [25]. In this approach, vehicles are arranged into groups. Each group has a group manager. The manger is responsible of the signing vehicle messages. The identity of the vehicle can be detected only by the group manager. Another group-based approach is described in [26]. In this approach, the group manger signs the vehicles pseudonyms to reduce the certificate authority workload. Each vehicle produces its pseudonyms and signs its messages. But it is difficult to achieve that in a dynamic VANET, because of the size, membership revocation and the dynamic membership (new nodes enter the group and old nodes leave the group) that will increase the complexity and overheads.

Another architecture which does not depend on pseudonyms are described in [27]. This approach uses Hash-base Message Authentication Code (HMAC). Before a vehicle sends a message, it requests a symmetric key from the RSU to use it in the HMAC code. Then, the vehicle signs its message by the HMAC code. The receiver vehicle authenticates the message from the adjacent RSU. This approach offers anonymity but it depends highly on the RSU which may be not available.

Some papers do not use the pseudonyms or the groups to preserve the privacy as proposed in [28]. It proposes a shared private/public key which is given to all of the legitimate vehicles. This key is renewed regularly by an authority where each vehicle has its own public/private key to communicate with the authority.

3.2 Merging Cloud Computing with VANET

Among the existing works, authors in [29] proposed a new concept called Vehicular Cloud (VC). VC used underutilized vehicle resources to form a cloud by aggregating vehicular computing resources. The authors considered that VC refers to a group of large autonomous vehicles included the computing, sensing, communication, and physical resources, where they could be coordinated and dynamically allocated to end users. It is worth to note that the proposed system did not take the advantage of the conventional cloud, and was only based on the vehicular resources. In contrast, VC resources cannot always be switched on, and often require the authorization of the vehicle's owner, which can be absent if the vehicle is in a steady state (e.g., vehicles in a parking lot).

Authors in [6] divided the VANET clouds into three major clouds: Vehicular Clouds (VCs), Vehicles using Clouds (VuCs) and Hybrid Clouds (HCs). The VC is subdivided into two categories: a static cloud which refers to stationary vehicles providing cloud services, and a dynamic cloud which is set up on the demand in an ad hoc manner. A VuC allows a VANET to connect to the traditional cloud with RSUs, whereas the HC is a combination of VC and VuC. Moreover, the vehicles can only interact with the traditional cloud through RSUs, which act as gateways. However, vehicles cannot be connected to the traditional cloud if the RSUs are not available, as in rural areas.

Mershed and Artil in [30], addressed the problem of enabling the vehicles in the VANET to discover their needed services from the mobile cloud servers, which are moving nearby. The authors proposed a system called CROWN, which depends on the RSUs that act as cloud directories and interfaces. To achieve that, RSUs make their recorded data available to enable vehicles to discover the required cloud services within the area that covered by the RSU.

To provide the safety and non-safety services in the vehicular applications, authors in [31] proposed the use of cloud computing services via RSUs. (Vehicular Cloud for Roadside) VCR scenarios architecture was proposed to allow vehicles to make benefits from the private and public vehicular cloud services. The previous efforts can be considered as help systems for vehicles, to access the conventional cloud through the RSUs via a cloud gateway, in order to find the requested cloud service without using any mobile computing resources.

A pure cloud formed by the vehicles which has been proposed in [32]. It is a new service paradigm called Sensor as a Service (SenaaS) for the vehicle communication platforms, it makes their components available, including vehicle sensors and devices, to third-party vehicle monitoring applications, as cloud computing resources called sensor-cloud service. This proposal lacks of the use of the traditional cloud to improve the computing capacity which is usually requested by vehicles.

To deal with the issue of the vehicles avoiding obstacles, a cloud-assisted system for autonomous driving was proposed in [33] and called Carcel. Carcel is a system that enables the cloud to collect information from the autonomous vehicle sensors as well as from the roadside infrastructure, to help vehicles avoiding obstacles, such as pedestrians and other vehicles, which may not be directly detected by the sensors on the vehicle.

Authors in [34] addressed the issue of seamless access to the Internet by making the use of cloud-based VANETs. In this study, the authors proposed a cloud-supported gateway model, which is called Gateway as a Service (GaaS), in order to provide an efficient gateway connectivity and to enhance the Internet usage experience for the vehicular networks.

3.3 VANET-cloud security

Authors in [35] dealt with the cloud security issue for vehicular networks by proposing a new secure provisioning model called Vehicle-to Cloud (V2C). V2C is composed of a provisioning infrastructure, which links two levels, the automobile user and the infrastructure provider. In the proposed model, the authors integrated three security modules to enhance the security, an authentication module, an authorization and access control policies module and an assurance module. The authentication module manages the identities and authenticates the entities in V2C. The authorization and access control policies modules set the access control policies for every automobile user. To correlate management actions with the desired requirements, the assurance module is deployed throughout V2C. V2Cfocuseson the cloud services required by the automobile users, and is served via the traditional cloud. This proposal is not satisfy the privacy preservation and the other security requirements in the VANET.

GeoEncrypt (Geolock) in the VANETs has been proposed in [36]. The idea is to use the geographic location of a vehicle to generate a secret key. Messages are encrypted with the secret key, and the encoded texts are sent to the receiving vehicles. The receiving vehicles must be physically presented in a certain geographic region specified by the sender, to be able to decrypt the message. They are proposed as a future work to integrate this model into security methods, and the shape of

the decryption region will be extended from the square shape to any shape in a professional manner.

Authors in [37], considered the concept of VuC framework and proposed another layer named TIaaS (Traffic Information as a Service) on the top of the Cloud Computing Stack. The service offers fine-grained traffic information for all the vehicles which are subscribed to TIaaS from the Cloud. The authors have proposed the Geolock-based encryption to provide the security, privacy and the conditional anonymity. S. Mallisseryet. Al. in [38], have proposed the Cloud Enabled Secure Communication in the VANET as a method to classify the VANET's messages. In addition, the vehicles and the RSU use the cloud as the medium of storage. This proposed algorithm verifies the identity and authenticity of the vehicles and messages exchanged. The message exchanged and certificates used in VANET are encrypted by using the geolocation key of the RSU. The use of geolocation key provides a location confidentiality against vehicles outside. This method does not satisfy the authorization, privacy, non-repudiation, availability and the revocation.

S. Olariuet. al. in [39] illustrates the power of the VC concept by enumerating a numerous application scenarios, for example, Remote Configuration and Car Performance Checking, Big traffic data analysis, Smart location-based advertisements and Vehicle Witnesses. The authors have emphasized more research challenges in the vehicular cloud including security issues.

Another work in [40] proposed that the cloud is associated with a number of grids. A city or traffic area is partitioned into grids. The grid size is predefined with two GPS coordinates. Each cell is associated with a virtual machine in the cloud. The virtual machine can dynamically requests resources from the cloud. Therefore, the traffic of the whole city can be mapped to the cloud. The customized security protocols can be configured and replaced in the VSecurity module.

Authors in [41] proposed a protocol to ensure both the message authentication and the privacy preservation. The proposed scheme is based on a secure elliptic curve digital signature algorithm approach. Here, the authors have considered that the transport authority is sending all the vehicle registration details to the RSU. This can be considered as an invasion of privacy on the vehicular users.

J. Serna et al. [42] proposed the basis of privacy mechanism that uses an authorization paradigm based on a Mandatory Access Control model, and a novel mechanism that propagates trust information based on a vehicles geolocation. With the change of the geographical location, the trust information is passed to a new regional CA. The authors have considered that the geographic location change happens only when a vehicle crosses the border of another country.

4. SECURITY REQUIREMENTS IN VANET

VANET should be protected against attacks by a security model. VANET security model should satisfy the following requirements and attributes as mentioned in [43] and [44].

1. Authentication
The identity of a vehicle should be verified to determine if it is a legitimate vehicle or not. Thus, the sender should authenticate each message before sending it. This prevents the intruders from sending false messages.

2. Authorization
Authorization establishes what each entity (vehicle, RSU and etc) is allowed to do in the system, e.g. the types of messages could be sent, information update rules and the protocols that are allowed to execute.

3. Data Integrity
Message integrity is very important. If a received message was altered by an attacker, the receiver should be able to detect it. Therefore, it is not enough to get a message from a legitimate sender but also the message itself should be verified. In addition, this requirement should detect the message repetition by an attacker.

4. Non-repudiation
A misbehaving vehicle may send incorrect information where the vehicle itself is legitimate, and the message is consistent. This behavior may lead to bad situations like accidents. The sender should not deny that he sent that message, so it should add a liability to the user of the messages which he sent.

5. Privacy

Drivers want to protect their identities from others. This is a very critical requirement. However, the problem is that the privacy conflicts with the authentication and non-repudiation concepts. Hence, many researches try to solve that problem.

6. Availability
The system should be available all the time because the disconnection for a short time may be dangerous. The system should be protected against the Denial of Service (DoS) Attack. This attack may be done by jamming the communication channel. Also, the availability includes these methods which ensure the authorities are available and should trust each other when a vehicle moves from a region to another.

7. Confidentiality
Confidentiality is a vital attribute to keep the content of the message secret if needed.

8. Entity Revocation
The ability to revoke the vehicles or the authorities is very important. For example, when a vehicle engages in malicious activity, it must be revoked.

5. PROPOSED VANET SECURITY AS A SERVICE (VSAAS)

5.1 VSaaS Architecture

Our work proposes VSaaS framework model to manage the security services in the VANET based on the cloud, and provide a secure VANET communication between the different entities, e.g. vehicles, authorities and etc. This framework model consists of different modules as shown in figure 1:

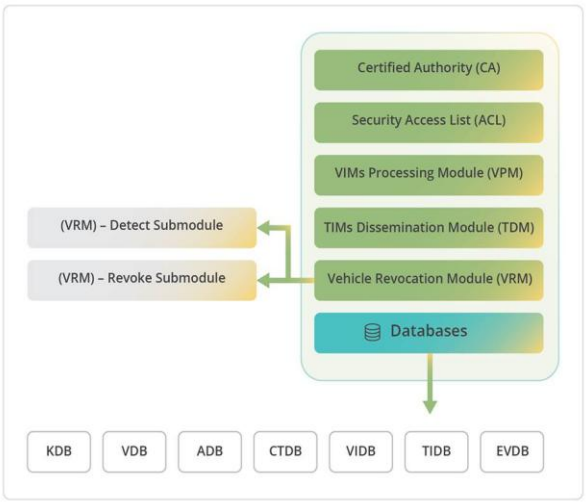

Figure1: VANET Security as a Service (VSaaS)

1. Certified Authority (CA): this module is the main one in the VSaaS because it is responsible for:

- Vehicles and Authorities Registration.
- Key Management mechanisms, to generate keys for different entities and renew these keys when they become expired.
- Authenticate the vehicles and their information messages, and authenticate the authorities that interacting with the VSaaS.
- Vehicle identity identification mechanism to preserve the privacy, and enable the traceability done only by the trusted authorities that have a permission to track vehicles.

2. Security Access List (ACL): this module is responsible for:

- Allowing/denying the inter-actions that will be done between the different entities (vehicles, authorities, VSaaS modules).
- Allowing/denying the intra-actions that will be done between the modules within VSaaS.

3. Vehicle Revocation Module (VRM): this module is divided into two parts: detecting sub-module and revoking sub-module. The first one should be constructed by some algorithms to detect a misbehaved vehicle (this part is out of our scope). The second part is responsible for revoking a misbehaved vehicle when detected.

4. VIMs Processing Module (VPM): this module is responsible for processing the Vehicle Information Messages (VIMs), which are sent by the vehicles as coarse-information messages and constructing fine-information messages, which called Traffic Information Messages (TIMs).

5. TIMs Dissemination Module (TDM): this module is responsible for disseminating the TIMs to the vehicles based on their locations.

6. Databases: this module stores different types of data as:

- **Keys Database (KDB):** which is used to store different types of keys such as the CA public/private keys and the TDM's shared secret key. This database is managed by CA.
- **Vehicles Database (VDB):** which is used to store all the vehicles information and their keys. This database is managed by the CA.
- **Authorities Database (ADB):** which is used to store all the authorities' information and their keys. This database is managed by the CA.
- **Vehicle Information Messages Database (VIDB):** which is used to store the VIMs. This database is managed by the CA. VPM and VRM have their permission to access this database.
- **Traffic Information Messages Database (TIDB):** which is used to store the TIMs. This database is managed by VPM. TDM has its permission to access this database.
- **Cellular Towers Database (CTDB):** which is used to store all the cellular towers information, their coordinates and routes. This database is managed by system administrators. TDM has its permission to access this database.
- **Event Viewer Database (EVDB):** which is used to store all the events within the VSaaS framework model. System administrators and different modules have a permission to access this database especially for some reporting issues.

5.2 Vehicle Information Messages (VIMs) and Traffic Information Messages (TIMs)

In the traditional VANET, safety messages are the messages or the beacons which are broadcasted by vehicles, to warn the other vehicles about the traffic situations like congestions and collisions. These messages contain information like location, speed, direction and acceleration. These messages are sent into the forms of the V2V and V2I communications. Hence, these messages are divided into two types: periodic and event driven messages, which are sent when a hazardous situation occurs. Periodic messages are considered to be an important type of messages that supports a decision that has been taken in the safety applications. Periodic messages are broadcasted to surrounding vehicles, but it may leads to a wasted bandwidth consumption, especially in the dense environment, in addition to increasing the probability of a storm problem occurring.

This work proposes the VIMs, which are shaped in a form of V2Cloud communication, as an alternative of the safety messages which are shaped in a form of V2V and V2I. VIMs include the current position, speed, direction, timestamp and the heading information, which are sent directly to the cloud infrastructure, and stored in the Vehicle Information Database (VIDB) if they were authenticated by the CA. Also, this work proposes the vehicles which will send asynchronous VIMs when the change in parameters exceeds a certain percentage (needs more experiments to determine it), or when the hazardous conditions are occurred. Other Modules in the VSaaS may have a permission to access the VIDB, such as the VPM which is made to process the VIMs, and construct fine-grained information which called Traffic Information Messages (TIMs). TIMs will be stored in the Traffic Information Database (TIDB). TDM module disseminates the TIMs to the vehicles based on their location.

In addition, this work proposes a Message Type (MT) field in each VANET message sent in the V2Cloud between the

different entities. MT is chosen to be a value of 32-bit length. MT is used to identify each message e.g. who is the sender and receiver of the message? And what is the aim of the message. Some examples are shown in table 1.

Table1: Examples of Messages in V2Cloud and VSaaS

MT	Sender	Receiver	Message Name	Description
7	CA	Vehicle	KillVehicle	CA Sends Revoked Message to Vehicle N
8	Vehicle	CA	SecureVehicleInformationMessage	Vehicle sends secure VIMs to CA
9	CA	Vehicles	SecureTrafficInformationMessage	CA sends secure TIMs to Vehicles based on location
13	CA	Authority	killAuthority	CA Sends Revoked Message to Authority M

5.3 Vehicle Entity in VSaaS

Before giving vehicle N a license to work, it should be registered in the CA by taking its physical Vehicle Identification Number (VIN) in some ways, to ensure that the VIN is true. VIN consists of 17 digits [45] and was officially described in ISO standard 3779 in February 1977, and revised at last in 1983. CA extracts all the vehicle's information from this VIN. Also, the owner's information should be given. Then, CA generates the Vehicle Identification Number (VID_N) and the public/private keys (Pub_N, Prv_N) for the vehicle N. In our work, VID_N is chosen to be a value of 64-bit length. This length can present more than 18 billion of values. May be other values can be used. The public/private keys (Pub_N, Prv_N) have a long lifetime (a year for example). When a vehicle renews its license, the CA will generate and install new keys on the vehicle.

Each vehicle has a Tamper-Proof Device (TPD) installed by the manufacturer, to store all the secret information used in VANET. CA preinstalls the (Pub_N, Prv_N) and Pub_{CA} on each vehicle N's TDP in addition to the VID_N. Hence, CA has public/private keys (Pub_{CA}, Prv_{CA}). The public and private keys for vehicles and the CA are generated according to the public-key cipher algorithms RSA. Also, the (Pub_{CA}, Prv_{CA}) was considered to have a medium lifetime (a month for example). When CA's public and private keys are renewed, CA broadcasts **the CAKeyUpdateforVehicle** message to all the vehicles which contain the new Pub_{CA}. Vehicles that did not receive the **CAKeyUpdateforVehicle** message according to different reasons, can send a **VehicleRequestCAKeyUpdate** message to request the new Pub_{CA}.

For liability, vehicles' identities should be added to the vehicles' messages, but this requirement contradicts with the privacy. Therefore, vehicles' identities should be hidden (encrypted) from the others, only CA can identify the vehicles' identities. To solve it, CA generates a symmetric key called Privacy Key K_{PRIV} used to encrypt/decrypt the vehicles' identities. All the registered (trusted) vehicles have the same privacy key K_{PRIV}. This key has a medium lifetime (a month for example). The key size is selected to be 128-bit, which is a common size for the symmetric ciphers AES. K_{PRIV} is preinstalled on the vehicle N's TDP when the vehicle N is registered with CA. It is worth to mention that the privacy key K_{PRIV} provides authentication and privacy. Authentication is achieved because only the registered and trusted vehicles have this privacy key K_{PRIV}. Using the same privacy key K_{PRIV} by all the vehicles at the same time to authenticate the messages or a part of them, provides anonymity which achieved the privacy.

When the vehicle N sends a message or requests different keys from the CA, TDP will add Encrypted VID (EVID). EVID value is produced by concatenating the VID_N to the current reading (xy-coordinates) from the tamper GPS installed by the manufacturer, then encrypting the all with the privacy key K_{PRIV}. Show algorithms in table 2. EVID value is a portion of any message sent to the CA, where the entire message should be encrypted by the Pub_{CA}. As a result, only CA can decrypt the message by its private key Prv_{CA}. Concatenating the xy with the VID_N before encryption, should ensure that the EVID value must be different in each message, and mitigates the linking between the two messages generated from the same vehicle.

Table2: Algorithms to Hide/Identify Vehicle Identity

Algorithm: Hiding Vehicle Identity
Input: VID_N
Output: Encrypted VID_N (EVID)
1. Read xy-coordinates from Tamper GPS
2. EVID = $Enc_{sym}(VID_N \| xy, K_{PRIV})$
3. Return EVID

Algorithm: Identifying Vehicle Identity
Input: Encrypted VID_N (EVID)
Output: VID_N
1. $VID_N \| xy = Dec_{sym}$ (EVID, K_{PRIV})
2. Extract VID_N
3.Return VID_N

To send secure VIMs, vehicle N's TDP concatenates the message (m), the message type (MT), the time stamp (t) and the EVID together. Then encrypting the all by the CA's public key Pub_{CA} to form a secure VIM. Vehicles send secure VIMs to the CA to verify and authenticate them. CA receives the secure VIMs and decrypts them by its private key Prv_{CA}. CA validates the time stamp (t) and extracts the EVID to authenticate the Vehicle Identity VID_N. If (t) and (VID_N) are valid, CA extracts the message (m) and stores it in the VIDB. VPM processes VIMs to construct fine-grained information, which called Traffic Information Messages (TIMs), and stores them in the TIDB. TDM gets TIM and concatenates it to (t) and (MT), then encrypts the all by a symmetric shared key called Dissemination Key KD to form the secure TIMs. After that, TDM determines the route of the TIMs based on the location of the cellular towers coordinates, which are stored in a Cellular Tower Database (CTDB) to disseminate them. Vehicles that receive the secure TIMs, decrypt them by the Dissemination Key (KD) and verify time stamp (t) to get the TIM. Show Algorithms in table 3.

When the vehicle N is registered with the CA, it preinstalls the KD on the vehicle N's TDP in addition to the previous secret information: (Pub_N, Prv_N), Pub_{CA}, VID_N and K_{PRIV}. This key has a short lifetime (a day for example). The key size is selected to be 128-bit which the common size is for the symmetric ciphers AES. When the CA renews the KD, it broadcasts the **DissKeyUpdate** message, which contains the new Dissemination Key (KD) to all the vehicles. Vehicles that did not receive the **DissKeyUpdate** message because of the

different reasons, can request the new Dissemination Key. In addition to that, CA will pass the new key to the DTM. Hence, when the CA sends the new KD, it is signed by the CA's private key to ensure the CA authentication, and encrypted by the key K_{PRIV}, to achieve the confidentiality and vehicles authentication. In addition, KD is a shared symmetric key which is better in the performance.

Table3: Algorithms to send VIMs and Disseminate TIMs

Algorithm: Sending Secure Vehicle Information Message (VIM)
Input: Vehicle Information Message m and VID_N
Output: secure VIM
1.Get current timestamp t
2. Set Message Type (MT)=8
3. EVID = Hiding Vehicle Identity Algorithm (VID_N)
4. M = m || t || MT || EVID
5. secure VIM = Enc_{pub} (M, Pub_{CA})
6.Return secure VIM

Algorithm: CA Verifying Secured Vehicle Information Message (VIM)
Input: secure VIM
Output: Vehicle Information Message m and VID_N, or null
1. M = Dec_{prv} (secure VIM, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. VID_N = Identifying Vehicle Identity Algorithm (EVID)
6. If VID_N is false, then return null and stop
7. Extract m from M
8. Return m and VID_N
9. Store all information into VIDB.

Algorithm: TDM Disseminating Secure Traffic Information Message (TIM)
Input: Traffic Information Message m
Output: secure TIM and route
1.Get current timestamp t
2. Set Message Type (MT)=9
4. M = m || t || MT
5. secure TIM = Enc_{sym} (M, KD)
6. Determine the route for this TIM based on the location of cellular towers coordinates stored in CTDB.
7.Return secure TIM and route

Algorithm: Vehicle Verifying Secured Traffic Information Message (TIM)
Input: secure TIM
Output: Traffic Information Message m or null
1. M = Dec_{sym} (secure TIM, KD)
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
6. Extract m from M
7. Return m

All the messages in the VANET environment should be protected against the replay attack. This protection is achieved by adding the time information to the message which is called timestamp (t). When the vehicle or the CA receives a message, it will check the validity of its timestamp (t). TDP has an internal clock .TDP is responsible for adding timestamps to the message before sending them in addition to check the validity of timestamps in the received messages.

5.4 Vehicle Revocation

When a misbehaved vehicle is detected, it should be revoked by Vehicle Revocation Module (VRM). The algorithm which determines the misbehaved vehicle, is out of our work scope. This work aimed to design a security model framework for the (V2Cloud).

In the previous works, where there is no general VANET security model framework, each vehicle should have a Revocation Key List, and check if the received messages were generated by the trusted vehicles or not. This was done by different approaches, but these approaches had many issues. For example, it needs a large storage space at each vehicle. Moreover, including more parameters or keys in the safety messages, leads to larger message size, and needs more computations to verify the messages at the receiver side. In addition, the need of searching in a very huge number of keys, costs more time.

In our work, CA is receiving all vehicle messages and storing them in the VIDB. VRM is divided into two parts: detecting sub-module and revoking sub-module. The first one should be constructed by some algorithms to detect the misbehaved vehicle (this part is out of our scope). This sub-module has its permission to access the VIDB to check the behavior and messages of the vehicles, in order to detect any misbehaved vehicle. If it was detected, the second part, which is responsible for revoking the misbehaved vehicle, changes the vehicle status to invalid, and create the revoked report (RR) that indicates to why the vehicle will be revoked. Then, CA will send a **killVehicle** message to that vehicle. This message includes RR which is signed by the CA's private key (Prv_{CA}), to ensure that the message is generated by the CA. After that, concatenate the signed RR with the Message Type (MT), timestamps (t), and encrypt all of them by the vehicle's public key Pub_N, to ensure that only the misbehaved vehicle N can decrypt this message by its private key Prv_N. When the misbehaved vehicle N receives the **killVehicle** message and decrypts it, the vehicle's TDP checks the validity of timestamps and verifies the signed revoked report part by the CA's public key Pub_{CA}. Finally, vehicle N's TDP will stop working and erase all the secret information. Show algorithms in table 4.

Table4: Algorithms to Revoke Vehicles

Algorithm: CA Sending Revoked Message to Vehicle N
Input: VID_N and revoke report RR
Output: KillVehicle message
1. Get current timestamp
2. Set Message Type (MT)=7
3. SignRR = $Sign_{prv}$(RR, Prv_{CA})
4. M = t || MT || SignRR
5. KillVehcile = Enc_{pub}(M, Pub_N)
6. Return KillVehcile

Algorithm: Vehicle N Receiving Revoked Message
Input: KillVehicle
Output: N'TDP turn-off or null
1. M = Dec_{prv} (KillVehcile, Prv_N)
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. $VerifySign_{pub}$(SignRR, Pub_{CA}) = false, then return null and stop
6. Erase all keys and turn off TDP

5.5 Authority Entity in VSaaS

This work classifies the authorities which are registered with the CAs into three different types according to their permission:

- **Traceability Authority (TA)** which has a permission to track the vehicle identity and location.
- **Transportation and Traffic Monitor Authority (TTMA)** which has a permission to monitor and analyze the transport traffic, depending on the information recorded in the TIDB.
- **Manufacturers** which have a permission to provide all the firmware updates, and check the vehicle performance remotely, depending on the sensor messages which are sent periodically by the vehicle itself. (This part is out of our work scope).

A Virtual Private Network (VPN) mechanism was proposed to connect these authorities with the VSaaS. The governmental body, which manages the VSaaS, defines the authority M by giving the CA all the authority's information, such as (name, type, address, telephone number, contact person, email, IP address and permissions). The CA generates the Authority Identification Number (AID_M) which is chosen to be a value of 16-bit length. And by this length, more than 65000 of the values can be presented. May be other values can be used. Also, CA generates the authority's public-private keys pair (Pub_M, Prv_M). The public and private keys are generated according to the public-key cipher algorithms RSA. Also, the (Pub_M, Prv_M) was considered to have a long lifetime (a year for example). Finally, CA generates the secret shared key K_{SM} that will be used to exchange the information and messages between the authority M and the VSaaS modules such as CA. The key K_{SM}'s size is selected to be 128-bit, which is the common size for the symmetric ciphers AES. This key has a medium lifetime (a month for example). Each authority has a local security process (SP), which has a specific identifier SPID chosen to be a value of 16-bit length. And by this length, more than 65000 of the values can be presented. May be other values can be used. The local SP is responsible for interacting with the VSaaS modules in a secure way. Hence, the current Pub_{CA} and Authority Identification Number (AID_M) have been given locally to the authority M when the site has been installed.

To register the authority, the new authority generates a temporary symmetric key K_{TMP} used only in the registration process (used for one time). The key size is selected to be 128-bit which the common size is for the symmetric ciphers AES. The new authority sends an **AuthorityRegister** message to the CA where the message consists of MT, t, AID_M and K_{TMP}. Then, encrypting the all by the CAs' public key Pub_{CA} to ensure that only the CA can decrypt this message. When CA received the **AuthorityRegister** message, CA decrypts it by its private key Prv_{CA}. Then, it validates the timestamp (t). After that, the CA extracts the AID_M and the temporary symmetric key K_{TMP} to validate them. Finally, the CA forms the **OkRegister** message that consists of MT, t and a signed part which consists of the Pub_M, Prv_M, SPID and the K_{SM} which also are signed by the Prv_{CA}, then encrypting the all by K_{TMP}. When the authority M receives the **OkRegister** message, it decrypts **OkRegister** message by the temporary symmetric key K_{TMP}, and verifies the signed portion by the Pub_{CA} to ensure that all the secret keys and information are generated by the CA. Then, authority M stores all the secret keys and information which are sent by the CA and it also erases the K_{TMP}. Now, authority M creates a local security process SP with SPID to interact with the CA in a secure way. Authority M' SP forms **ACKAuthorityRegister** message that consists of the MT and t. This message is encrypted by K_{SM}. Finally, the CA verifies the **ACKAuthorityRegister** message, and erases the K_{TMP} and be ready to exchange the messages with authority M. Show algorithm in table5. Moreover, authorities can request a new CA's public key and a shared key K_{SM} when they are expired.

Table5: Authority Registration Algorithms

Algorithm: Authority Sending Registration Request
Input: Pub_{CA} and AID_M
Output: AuthorityRegister message
1.Authority generates temporary symmetric key K_{TMP}
2. Message Type =10
3. Get current timestamp t
4. M =MT || t || AID_M || K_{TMP}
5. AuthorityRegister = Enc_{pub} (M, Pub_{CA})
6. Return AuthorityRegister

Algorithm: CA Receiving Authority Registration Request and Sending Security Information to Authority
Input: AuthorityRegister
Output: OkRegister or null
1. M = Dec_{prv} (AuthorityRegister, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. CA Extracts AID_M and K_{TMP}.
6. CA Validates AID_M = false, then return null and stop
7. CA set Message Type =11
8. Get current timestamp t
9. SignokRegister= $Sign_{prv}$((Pub_M, Prv_M) || SPID || K_{SM} , Prv_{CA})
10. M = MT || t || SignokRegister
11. OKRegister = Enc_{sym} (M, K_{TMP}).
12. Return OKRegister to authority M

Algorithm: Authority Receiving Security Information from CA and send back ACK
Input: okRegister
Output: ACKAuthorityRegister message or null
1. M = Dec_{sym} (okRegister, K_{TMP})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. $VerifySign_{pub}$ (SignokRegister, Pub_{CA}) = false, then return null and stop
6. Extract and Store Security Information: (Pub_M, Prv_M) || SPID || K_{SM}
7. Erase temporary symmetric key K_{TMP}
8.Authority creates local security process with SPID
9. M' SP set Message Type =12
10. Get current timestamp t
11. M =MT || t
12. ACKAuthorityRegister = Enc_{sym} (M, K_{SM})
13. Return ACKAuthorityRegister

If any authority is compromised, CA will send a **killAuthority** message to this authority to destroy M' local security process and erase all the keys. The compromised authority needs to register again with the CA according to the governmental security rules.

Traceability Authority (TA) is an authority which has a permission to track the vehicle identity and location. Today, tracking vehicles depends on the Vehicle Identification Number (VID_N). Vehicle Information Messages (VIMs) are stored in the (VIDB) that is managed by CA. To track vehicle N, TA should have a permission to access the required fields from the VIDB's records such as VID_N, xy-coordinates, speed

and etc. When TA has an order to track the vehicle N, which has the VID_N, TA's local security process SP will send **a Track** message to the CA, which consists of the MT, t, AID_M and the VID_N, then encrypting the all by the Pub_{CA}, to ensure that only the CA can decrypt this message. When the CA receives this message, it decrypts the message by the CA' private key and check the validity of the t, then, extracts the AID_M and the VID_N to verify them.

The CA informs the TA by the **startTracking** message to start tracking vehicle N. This message includes (MT, t and VID_N) encrypted by the TA's public key to ensure that only the TA can decrypt this message. When TA receives this message, its local security process starts to query and access all the needed information to track the vehicle N. Hence, the CA and authority TA' SP are using the shared secret key (K_{SM}) to encrypt all the traffics between them, to track the vehicle N through its VID_N. A symmetric key encryption was proposed because it is more efficient and faster than the public/private keys. When the TA completes the tracking order, it sends a **stopTracking** message to the CA. Show algorithms in table 6.

Table6: Algorithms to Track Vehicle N by TA

Algorithm: TA Sending Request to CA to Track Vehicle N
Input: VID_N and AID_M
Output: Track message
1. Get current timestamp t
2. Message Type =18
3. M =MT || t || AID_M || VID_N
4. Track = Enc_{pub}(M, Pub_{CA})
5. Return Track

Algorithm: CA Receiving Track Vehicle N Request from TA
Input: Track
Output: startTracking or null
1. M = Dec_{prv} (Track, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. AID_M = false, then return null and stop
6. VID_N = false, then return null and stop
7. CA set Message Type =19
8. Get current timestamp t
9. M = MT || t || VID_N
10. startTracking = Enc_{pub}(M, Pub_{TA})
11. Return startTracking

Algorithm: TA Receiving Start Tracking Vehicle N
Input: startTracking
Output: TA' local security process can access information in VIDB in secure way
1. M = Dec_{prv} (startTracking, Prv_{TA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. TA' local security process can access information in VIDB in secure way using shared secret key (K_{SM})

Algorithm: TA Sending Stop Tracking Vehicle N to CA
Input: VID_N
Output: stopTracking or null
1. Message Type =20
2. Get current timestamp t
3. M = MT || t || VID_N
4. StopTracking = Enc_{pub}(M, Pub_{CA})
5. Return StopTracking

Algorithm: CA Receiving Stop Tracking Vehicle N
Input: stopTracking
Output:
1. M = Dec_{prv}(StopTracking, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. The mission was completed

Transportation and Traffic Monitor Authority (TTMA) is the authority which has a permission to monitor and analyze the transportation traffic by depending on the information recorded in the TIDB. Therefore, the TTMA has a permission to read the information from the TIDB through the authority local security process. These information may contain the VID_N if the authority has a traceability permission, and it contain encrypted VID_N if the authority has not this permission. Moreover, the information forwarded from the CA to the TTMA should be encrypted by the secret shared key K_{SM}. The TTMA gets the secret shared key K_{SM} in the registration process.

5.6 Security Access List (ACL)

This module represents a set of permissions and rules to Allow/deny the inter-actions between the different entities (vehicles, authorities, VSaaS modules), and the intra-actions between the modules within the VSaaS. Our design of the VSaaS is modular. In the future, any type of the authorities, databases, new VSaaS's modules, can be easily and smoothly added to make specific tasks by defining its permissions, to interact with the different entities, databases and modules in the VSaaS. Table 7 shows the simple shape of the entities and modules permission against the different databases. Also, table 8 shows a simple way to define a permission for each authority according to its functions.

Table 7: Example of entities and modules permission

Module/ Database	KDB	VDB	ADB	VIDB	TIDB	CTBD
CA	Full	Full	Full	Full		Read
VRM		Read		Read		Read
VPM		Read		Read	Full	Read
TDM					Read	Read
TA Authority		Read		Read		Read
TTMA Authority					Read	Read

Table 8: Example of defining permission to each authority

AID_M	Traceability	Traffic Monitor	Manufacturer
002acb1555 zx0001	1	0	0
003zzz7654 9fagh2	0	1	0

6. CONCLUSION

This paper highlighted a number of previous related works which proposed the VANET security, merging the Cloud Computing with the VANET and the VANET-cloud security. Also, it proposed VANET based on the cloud (V2Cloud) and the design of a security model framework that is hosted on the cloud to manage the security services, and provide a secure VANET communication between the different entities e.g.

vehicles, authorities and etc. This security model framework is called VANET Security as a Service (VSaaS).

This paper showed that the VSaaS fulfilled the VANET's security requirements and protected the VANET against the different types of attacks. The implementation and the security analysis of our architecture along with the results of the performance will present in the second paper, to measure the security overhead of the secure Vehicle Information Messages (VIMs), which is sent by the vehicles to the cloud as a coarse-grained information

The VSaaS framework model is secure, efficient, modular, managed by cloud, resistant against attacks and fulfills the security requirements.

7. REFERENCES

[1] H. Hartenstein, K. Laberteaux , "VANET Vehicular Applications and Inter-Networking Technologies", John Wiley & Sons ISBN: 978-0-470-74056-9.January 2010.

[2] "Dedicate Short Range Communications (DSRC),"http://www.leearmstrong.com/Dsrc/DSRCHomeset.htm.

[3] C. Barberis, E. Gueli, L. Minh Tuan, G. Malnati and A. Nassisi, "A customizable visualization framework for VANET application design and development," Proc. Consumer Electronics (ICCE), 2011 IEEE InternationalConference on, 2011, pp. 569-570.

[4] S. Olariu, M. Eltoweissy and M. Younis, "Towards Autonomous Vehicular Clouds," ICST Transactions on Mobile Communications and Applications, vol. 11, no. 7-9, 2011, pp. 1-11.

[5] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "A view of cloud computing," Commun. ACM, vol. 53, no. 4, 2010, pp. 50-58; DOI10.1145/1721654.1721672.

[6] R. Hussain, J. Son, H. Eun, S. Kim and H. Oh ,"Rethinking Vehicular Communication: Merging VANET with Cloud Computing", IEEE International Conference on Cloud Computing Technology and Science, 2011, pp. 600-609;DOI 10.1109/CloudCom.2012.6427481.

[7] R. Hussain, Z. Rezaeifar, H. Oh, "A Paradigm Shift from Vehicular Ad Hoc Networks to VANET-Based Clouds ", Wireless Personal Communications: An International Journal Springer, 2015, vol. 83, no. 2, p.p 1131-1158;DOI 10.1007/s11277-015-2442-y

[8] M. A. Al Mamun, K. Anam, M. F. Onik, A M Esfar- E-Alam, "Deployment of Cloud Computing into VANET to Create Ad Hoc Cloud Network Architecture", Proceedings of the World Congress on Engineering and Computer Science, USA, 2012, vol. 1, ISBN 978-988-19251-6-9

[9] M. M. Manohara Pai , Nabil Ajam ; Radhika M. Pai ; Joseph Mouzna, "Transport and traffic rule violation monitoring service in ITS: A secured VANET cloud application", IEEE International Conference on Cloud Computing Technology and Science, 2015, pp. 213-218;DOI 10.1109/CCNC.2015.7157979

[10] G. Samara, W. Al-Salihy, R.Sures, Security issues and challenges of Vehicular Ad Hoc Networks (VANET), in IEEE 4th Int. Conf. on New Trends in Information Science and Service Science (NISS), 2010, pp. 393–398.

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

[12] M. Burmester, E. Magkos, V. Chrissikopoulos, "Strengthening Privacy Protection in VANETs," in IEEE Int. Conf. Networking and Communications, 2008 (WIMOB'08),pp.508–513,2008.

[13] L. Butty´an, T. Holczer, and I. Vajda, "On the effectiveness of changing pseudonyms to provide location privacy in vanets," in ESAS, 2007, pp. 129–141, 2007.

[14] L. Huang, K. Matsuura, H. Yamane, and K. Sezaki, "Enhancing wireless location privacy using silent period," in IEEE Wireless Communications and Networking Conference(WCNC),2005,pp.1187–1192.

[15] K. Sampigethaya, L. Huang, K. Matsuura, R. Poovendran, and K. Sezaki, "Caravan: Providing location privacy for VANET," in Escar 2005: 3rd Embedded Security in Cars Workshop, 2005.

[16] G. J. Freudiger, M. Raya, and M. Feleghhazi, Mix zones for location privacy in vehicular networks in Proc. First Int. Workshop on Wireless Networking for Intelligent Transportation Systems (WiNITS‟ 07), Vancouver, Canada, Aug. 2007.

[17] L. Buttyán, T. Holczer, and I. Vajda, On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs, Security and Privacy in Ad-hoc and Sensor Networks, Lecture Notes in Computer Science, Springer, Berlin / Heidelberg, vol. 4572, 2007, pp.129-141.

[18] AdiShamir , "Identity-based cryptosystems and signature schemes," Advances in Cryptoloty-Crypto‟84, LNCS, Vol. 196, Springer-Verlag, pp. 47 – 53, 1984.

[19] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairings," Advances in Cryptology-Asiacrypt, Springer-Verlag, pp. 514 – 532, 2001.

[20] P. Kamat, A. Baliga, W. Trappe, "An identity-based security framework for VANETs," Proceeding of International Conference on Mobile Computing and Networking, Los Angeles, California, USA, pp.94–95, 2006.

[21] P. Kamat, A. Baliga, W. Trappe, "Secure, Pseudonymous, and Auditable Communication in Vehicular Ad Hoc Networks," J. Security and Comm. Networks, vol. 1, no. 3, pp. 233 – 244, 2008.

[22] C. Lai, H. Chang, Chei Chung Lu, "A secure anonymous key mechanism for privacy protection in VANET," 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST), pp. 635 – 640, 2009.

[23] J. Sun, C. Zhang, Y. Zhang, Y. Fang, "An Identity-Based Security System for User Privacy in Vehicular Ad Hoc Networks," Parallel and Distributed Systems, IEEE Transactions on vol. 21 , no. 9, pp. 1227 – 1239, 2010.

[24] J. Guo, J.P. Baugh, and S. Wang, A group signature based secure and privacy-preserving vehicular

communication framework, in Mobile Networking for Vehicular Environments, pp. 103–108, 2007.

[25] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: a secure and privacy preserving protocol for vehicular communications" IEEE Transaction on Vehicular Technology, vol. 56, no. 6, pp. 3442-3456, 2007.

[26] G. Calandriello, P. Papadimitratos, J.P. Hubaux, and A. Lioy, "Efficient and Robust Pseudonymous Authentication in VANET" in Proc. 4th ACM int. Workshop on Vehicular Ad Hoc Networks, pp. 19–28, 2007.

[27] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, "RAISE: an efficient rsu-aided message authentication scheme in vehicular communication networks", in Proc. IEEE ICC 2008, Beijing, China, May 19–23, 2008.

[28] A. Bayrak, T. Acarman: "A Secure and privacy protecting protocol for VANET", Intelligent Vehicles Symposium (IV), 2010 IEEE, pp. 579 – 584, 2010.

[29] S. Olariu, T. Hristov, and G. Yan, "The Next Paradigm Shift: From Vehicular Networks to Vehicular Clouds," Developments in Mobile Ad Hoc Networking: The Cutting Edge Directions, Wiley, 2012.

[30] K. Mershad and H. Artail, "Finding a STAR in a Vehicular Cloud," IEEE Intelligent Transportation Systems, vol. 5, no. 2, 2013, pp. 55–68.

[31] D. Baby et al., "VCR: Vehicular Cloud for Road Side Scenarios," Advances in Comp. and Info. Tech., 2013, pp.541–52.

[32] N. Zingirian and C. Valenti, "Sensor Clouds for Intelligent Truck Monitoring," Proc. IEEE Intell. Veh. Symp., 2012, pp. 999–1004.

[33] S. Kumar, S. Gollakota, and D. Katabi, "A Cloud-Assisted Design for Autonomous Driving," Proc. 1st ACMWksp. on Mobile Cloud Comp., 2012, pp. 41–46.

[34] T. W. Lin, J. M. Shen, and H. C. Weng, "Cloud-Supported Seamless Internet Access in Intelligent Transportation Systems," Wireless Personal Commun., 2013, pp. 1–26.

[35] S. Rangarajan et al., "V2C: A Secure Vehicle to Cloud Framework for Virtualized and On-Demand Service Provisioning," Proc. ACM Int'l. Conf. Advances in Comp., Commun. and Informatics, 2012, pp. 148–54.

[36] G. Yan and S. Olariu, "An efficient geographic location-based security mechanism for vehicular ad hoc networks," in Proc. IEEE Int. Symp. TSP, Macau SAR, China, Oct. 2009, pp. 804–809.

[37] R. Hussain, F. Abbas, J. Son, and H. Oh, "Tiaas: Secure cloud-assisted traffic information dissemination in vehicular ad hoc networks," IEEE International Symposium on Cluster Computing and the Grid, vol. 0, pp. 178–179, 2013.

[38] S. Mallissery, M. Pai M M, R. M Pai, and S. A.., "Cloud Enabled Secure Communication in Vehicular Ad-hoc Networks," IEEE International Conference on Connected Vehicles and Expo (ICCVE) , Vienna, pp. 596–601, 2014

[39] S. Olariu, T. Hristov, and G. Yan, The Next Paradigm Shift: From Vehicular Networks to Vehicular Clouds. Wiley-IEEE Press, 2013, ch. Mobile Ad Hoc Networking: Cutting Edge Directions, Second Edition (eds. Stefano Basagni and Marco Conti and Silvia Giordano and AndlvanStojmenovic), pp. 645–700.

[40] G. Yan, D. Wen, S. Olariu, and M. Weigle, "Security Challenges in Vehicular Cloud Computing" IEEE International Transactions on Intelligent Transportation Systems , vol. 14, pp. 284–294, 2012.

[41] B. Mishra, S. K. Panigrahy, T. C. Tripathy, D. Jena, and S. K. Jena, "A secure and efficient message authentication protocol for vanets with privacy preservation," in Proceedings of the 2011 World Congress on Information and Communication Technologies, December 2011, pp. 880 – 885.

[42] J. Serna, J. Luna, and M. Medina, "Geolocation-based trust for vanet's privacy," in Proceedings of the 4th International Conference on Information Assurance and Security, September 2008, pp. 287–290.

[43] G. Samara, W. Al-Salihy, R.Sures, Security issues and challenges of Vehicular Ad Hoc Networks (VANET), in IEEE 4th Int. Conf. on New Trends in Information Science and Service Science (NISS), 2010, pp. 393–398.

[44] G. Samara, W. Al-Salihy, R. Sures, Security Analysis of Vehicular Ad Hoc Networks (VANET), in IEEE 2th Int. Conf. on Network Applications, Protocols and Services, 2010, DOI: 10.1109/NETAPPS.2010.17.

[45] "Vehicle Identification Number (VIN)" http://www.autohausaz.com/mercedes-auto-arts/mercedes-vehicle-identification-numbers.html