ASAP-V: A Privacy-preserving Authentication and Sybil detection Protocol for VANETs

Thiago Bruno M. de Sales[a,*], Angelo Perkusich[a], Leandro Melo de Sales[a], Hyggo Oliveira de Almeida[a], Gustavo Soares[a], Marcello de Sales[b]

[a] *Embedded Systems and Pervasive Computing Laboratory, Federal University of Campina Grande, Campina Grande, PB, Brazil*
[b] *Intuit, Inc. San Diego, C.A, USA*

Abstract

Node authentication, non-repudiation and anonymous communication are key roles to provide security in Vehicular Ad Hoc Networks (VANETs). On the other hand, the trade-off between authentication/non-repudiation and anonymous communication may lead to a harmful type of network attack called *sybil* attack. In such an attack, a malicious node behaves as if it is a large number of nodes. In this paper, we propose an anonymous authentication and *sybil* attack detection protocol for VANETs called *ASAP*-V. Experimental results suggest that *ASAP*-V is more robust against *sybil* attacks, with lower average detection time than the state-of-art works, also without false-positive and false-negative detections.

Keywords:
VANET, Security, Authentication, Privacy, Sybil Attack

1. Introduction

The Vehicular Ad Hoc Network (henceforth VANET) is an emerging type of Mobile Ad Hoc Network (MANET) that aims at providing vehicular safety

[*] Corresponding author

Email addresses: thiago.sales@ee.ufcg.edu.br (Thiago Bruno M. de Sales), perkusich@dee.ufcg.edu.br (Angelo Perkusich), leandro@embedded.ufcg.edu.br (Leandro Melo de Sales), hyggo@dsc.ufcg.edu.br (Hyggo Oliveira de Almeida), gsoares@computacao.ufcg.edu.br (Gustavo Soares), Marcello_deSales@intuit.com (Marcello de Sales)

applications, optimized vehicular traffic routing, and real-time applications for drivers and passengers, such as mobile infotainment [1]. Vehicles act as mobile nodes[1] that can send message to other vehicles and to road side units (RSUs), which are fixed infrastructures along the roads that may provide vehicle connectivity in sparse or low density areas. The communication between vehicles is called V2V (Vehicle-to-Vehicle), while the communication between vehicle and RSU is called V2I (Vehicle-to-Infrastructure). The IEEE 802.11p [2] standard is used for wireless communication.

The concepts behind VANET are bringing new challenges to a diverse of network research areas, including security [3]. Security has been considered a critical concern due to VANET's open wireless nature, since no authentication and association procedures are in 802.11p. Thus, VANET requires fundamental security aspects in the application layer such as the vehicles' message authentication and non-repudiation.

However, authentication and non-repudiation require a one-to-one correspondence between vehicle and identity, which may allow a malicious entity to build a vehicle's route profile. This may compromise users' privacy and lead to several user safety problems such as kidnapping, and undesirable tracking for mobile advertisement [4]. However, through a simple and secure, yet powerful mathematical and logical analysis, researchers [5] identified that the more suitable identity assignment to vehicles is to each one to store multiple identities (also called multiple *pseudonyms*), without sharing any identity with others, each identity unique to each vehicle.

Initially proposed by Raya et. al.[6], there have been many other multiple-pseudonym-based approaches for securing location privacy in VANET [7], [8], [9] and [10]. Many researchers adopt the concept of Mix Zones [11] to prevent malicious entities from linking different vehicle's pseudonyms [12], [4], [13], [14], [15] and [16]. In this case, the main goal is to spatially and temporally build groups of vehicles in order to allow them to change their pseudonyms without compromising users' privacy.

Even so, the multiple-pseudonym approach leads to a simple, but harmful type of network attack called a *sybil* attack [17]. In *sybil* attacks, a malicious node behaves as if it is a large number of nodes. Since a vehicle may have multiple valid identities to control its privacy, a malicious vehicle can send multiple messages with different identities to inform false events in VANETs.

[1] We use the terms vehicles, nodes and cars interchangeably.

Some examples include false on-road obstacles and false emergency braking warning along a road, or creating an illusion of traffic congestion through beacon messages by claiming to be at different locations. The presence of a *sybil* node increases packet loss, and decrease the packet delivery ratio and the aggregated throughput due to incorrect routing paths [18], [19] and [20]. When a vehicle is not a *sybil* node, it is called *legitimate vehicle*.

To avoid a vehicle from keeping, at the same time, multiple identities in multiple-pseudonym-based approaches, other works propose a single vehicle to store only one identity at a time [21], [22] and [23]. To change its identity, each vehicle requests from the RSUs along the road, one new identity that is valid only in the region where the RSU is responsible for. These approaches lack flexibility and they are highly dependable on the RSUs deployment methods [24].

Based on a deep investigation of the state of the art approach, this paper proposes a decentralized privacy-preserving authentication and *sybil* attack detection protocol for VANETs called *ASAP*-V (*Authentication and Sybil Attack detection Protocol for VANETs*). The authentication process is based on the multiple-pseudonym approach to provide location privacy for the users. Non-repudiation is also achieved through the Group Signature Scheme [25]. Our approach uses the anonymity set theory [26] in a multilevel fashion to detect and avoid *sybil* attacks, while still providing users' privacy control.

The contributions of this paper are summarized as follows:

- a new privacy-preserving authentication and *sybil* detection protocol;
- decentralization of the *sybil* detection approach, which does not require a fixed infrastructure during detection time;
- an approach without false-negative and false-positive *sybil* detections, even without the support of an infrastructure;
- it is able to detect *sybil* attacks from both beacon and event-driven messages;
- our results suggest that the detection protocol provides lower average *sybil* attack detection time than the state-of-the-art approaches.

The remainder of the paper is organized as follows: Section 2 details the proposed privacy-preserving authentication and *sybil* detection approach.

Section 3 discusses the experiments and results of the approach, while Section 4 discusses the related works about privacy-preserving *sybil* detection attacks. Finally, Section 5 presents the conclusions and future work.

2. A Privacy-preserving Authentication and Sybil Attack Detection Protocol for VANET

This section details the *ASAP*-V architecture. The goal is to provide strong privacy-preserving authentication and non-repudiation while detecting *sybil* attacks.

2.1. The ASAP-V protocol description

The *ASAP*-V protocol is divided into four phases: the registration phase (Phase 1); the temporary identity (pseudonym) assignment phase (Phase 2); the *sybil* detection phase (Phase 3); and the prosecution phase (Phase 4). For the next sections, Table 1 summarizes the notations for the protocol description.

Table 1: Notations.

Symbol	Description
v_c	A Vehicle c.
RSU_n	The n^{th} RSU along the road.
$cert_a$	Digital certificate of an entity a.
$k^+_{a,n}$	The a's n^{th} public key.
$k^-_{a,n}$	The a's n^{th} private key.
$cert_{a,n}$	The a's n^{th} public key digital certificate.
TK_a	a's set of temporary public/private key pairs (pseudonyms).
gsk_c	Group signing key of vehicle c.
gpk	A group public key.
grt_a	Group revocation token of vehicle a.
RL	List of revoked *group revocation tokens.*
tmp	Current timestamp.
tmp_{ctn}	The timestamp that the content ctn was digitally signed.
$threshold_X$	Denotes the maximum (X = *max*) or minimum (X = *min*) timestamp value to define time intervals.

continued on next page

Table 1 – *continued from previous page*

Symbol	Description
$Signed_a^{ctn}$	Entity a signed content cnt.
$Sign(\bullet)$	A digital signature function.
$a \Rightarrow b : ctn$	Entity a sends content ctn to entity b.
$Verify(\bullet)$	Cryptography verification function.
$E(\bullet)$	An encryption function.
$D(\bullet)$	A decryption function.
$sybil_{v_c}$	Vehicle v_c is a *sybil* node.

The *ASAP*-V protocol runs in a system SV, defined as:

$$SV = \langle V, RSU, ca \rangle,$$

such that,

- $V = \{v_0, v_1, ..., v_p\}$ is the set of all registered vehicles ($p \in \mathbb{N}$);
- $RSU = \{RSU_0, RSU_1, RSU_2, ..., RSU_r\}$ is the set of Road Side Units (RSUs) registered in SV. The RSUs share a unique public/private key pair $(k^+_{RSU}, k^-_{RSU}) \in K_{RSU}$, and $cert_{RSU}$ denotes the RSUs' public key digital certificate;
- *ca* denotes a Certificate Authority (CA), which is responsible for managing key materials and for storing vehicles' and RSUs' data in SV. The tuple $\langle(k^-_{CA}, k^+_{CA}, cert_{CA})$, *gmsk*, K-AS $\rangle$ represents a CA such that k^+_{CA} and k^-_{CA} denote its public and private key pair, and $cert_{CA}$ the public key's digital certificate. A CA belongs to the government and is the only entity that can trace vehicles' owners identities from vehicles' messages. This procedure is only possible through CA's group management signing key *gmsk*. Finally, K-AS denotes the set of Anonymity Sets [27], discussed in Section 2.2.

2.2. Vehicle Registration and Authentication (Phase 1)

Figure 1 depicts the registration phase. A CA is responsible for managing the unique vehicles' identities by using the Group Signature Scheme [25]. A vehicle v_c has its own group signing key (gsk_c) and also shares the group public key (gpk) with others. Vehicles also store the digital certificate of the CA ($cert_{CA}$) and the digital certificate of the road side units ($cert_{RSU}$).

Moreover, each vehicle receives a set TK_c of w temporary (identities, or pseudonyms) asymmetric key pairs and their digital certificates ($k^+_{c,i}$, $k^-_{c,i}$, $cert_{c,i}$, $1 \leq i \leq w$). In case of judicial disputes, the CA is the only entity that can trace the original vehicle identity from a given message m_c using its group manager secret key ($gmsk$).

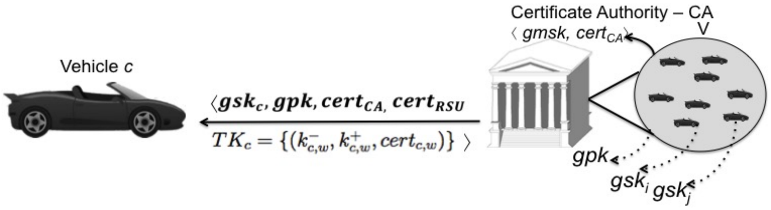

Certificate Authority – CA
⟨ gmsk, cert_CA⟩
V
Vehicle c
⟨gsk_c, gpk, cert_CA, cert_RSU
TK_c = {(k^-_{c,w}, k^+_{c,w}, cert_{c,w})} ⟩
gpk
gsk_i
gsk_j

Figure 1: Vehicle registration and authentication.

The second step of the registration phase aims to cluster each vehicle into multiple sets of vehicles $AS_{i,j}$ $(i, j \in \mathbb{N}^*; 1 \leq i \leq m, 1 \leq j \leq n)$. Each set $AS_{i,j}$ has all of the properties of the anonymity set theory. According to Figure 2, the anonymity sets $AS_{m,n}$ are organized in m levels, where each level has n anonymity sets. The proposed multilevel anonymity set architecture has the following properties, which from 1 to 3 aim at protecting users privacy, while property 4 aims at detecting *sybil* attacks:

1. Let K-AS = $\{AS_{1,1}$, $AS_{1,2}$, ..., $AS_{1,n}$, ..., $AS_{2,1}$, $AS_{2,2}$, ..., $AS_{2,n}$, ..., $AS_{m,n}\}$ be the set of all anonymity sets;
2. Each vehicle must belong to at least k sets ($1<k<n$) in each level. AS_c denotes the set of all anonymity sets that vehicle v_c belongs to ($AS_c \subseteq K\text{-}AS$).
3. Each anonymity set $AS_{i,j}$ has a digital certificate $cert_{AS_{i,j}}$ signed by CA, that is, $Signed_{CA}^{cert_{AS_{i,j}}}$. Thus, if vehicle v_c belongs to $AS_{i,j}$, then v_c must store $cert_{AS_{i,j}}$. Formally, $v_c \in AS_{i,j} \rightarrow cert_{AS_{i,j}} \in CERT_{AS_c}$ $\{i, j \in \mathbb{N}, 1 \leq i \leq m$ e $1 \leq j \leq n\}$, such that:
   (a) $CERT_{AS_c}$ is the set of all anonymity sets' digital certificates in which the vehicle v_c belongs to; and,
   (b) for a given time interval t, a vehicle v_c must choose a subset $CERT^t_{AS_c}$ of anonymity set digital certificates ($CERT^t_{AS_c} \subseteq CERT_{AS_c}$) that comprises only one digital certificate per level.

4. Any two vehicles v_c and $v_{c'}$ in $AS_{1,j}$ must not belong to the same anonymity set of some lower level. Formally[2], $\forall v_c, v_{c'} \in AS_{1,j}$, $\exists\, i$ ($\forall\, r$ ($v_c \in AS_{i,r} \oplus v_{c'} \in AS_{i,r}$)) $\{i, j, r \in \mathbb{N} : 1 < i \leq m, 1 \leq j \leq n, 1 \leq r \leq n\}$. Therefore, $CERT^t_{AS_c} - CERT^t_{AS_{c'}} \neq \emptyset$.

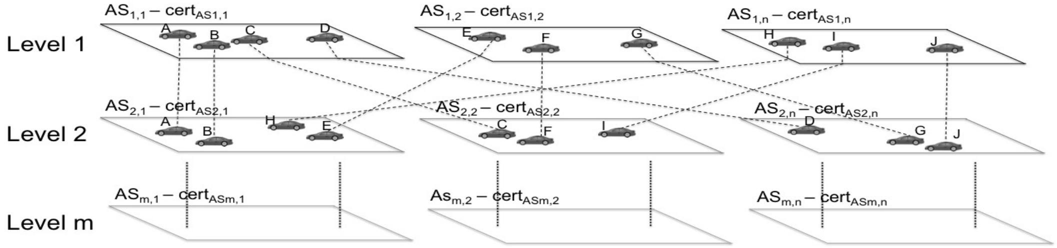

$AS_{1,1}$ – $cert_{AS1,1}$
$AS_{1,2}$ – $cert_{AS1,2}$
$AS_{1,n}$ – $cert_{AS1,n}$
Level 1
A
B
C
D
E
F
G
H
I
J
$AS_{2,1}$ – $cert_{AS2,1}$
$AS_{2,2}$ – $cert_{AS2,2}$
$AS_{2,n}$ – $cert_{AS2,n}$
Level 2
A
B
H
E
C
F
I
D
G
J
$AS_{m,1}$ – $cert_{ASm,1}$
$As_{m,2}$ – $cert_{ASm,2}$
$AS_{m,n}$ – $cert_{ASm,n}$
Level m

Figure 2: The Multilevel *ASAP*-V Architecture.

Finally, a vehicle v_c is represented by the following tuple:

$$v_c = \langle (gsk_c, gpk), TK_c, AS_c, CERT_{AS_c} \rangle.$$

2.3. Temporary Key Assignment (Phase 2)

In order to protect users' privacy, the multiple-pseudonym approach is used, and the second phase is responsible for managing pseudonym assignments to vehicles. In the *ASAP*-V protocol, cryptography asymmetric key pairs represent vehicle pseudonyms. For instance, the key pair $k^+_{c,i}/k^-_{c,i}$ and its digital certificate $cert_{c,i}$ is the i^{th} pseudonym of the vehicle v_c. Vehicles obtain temporary keys from authenticated RSUs available along the roads.

We modeled the proposed pseudonym renewal protocol using the CSP (Communication Sequencial Process) notation. However, before diving into the details of the protocol's authentication properties in CSP notation, we first give an overview of such protocol, as illustrated in Figure 3:

- In Step 1, a vehicle v_c requests the set of temporary keys from a given RSU. The vehicle v_c, through the group signature schema, signs its i^{th} temporary digital certificate ($cert_{c,i}$) and a random UUID (*Universely Unique IDentifier*) value, which generates the group signature σ. The *UUID* is a 128-bit identifier value and, within the context of ASAP-V

[2] The symbol $\oplus$ represents an exclusive-or operator.

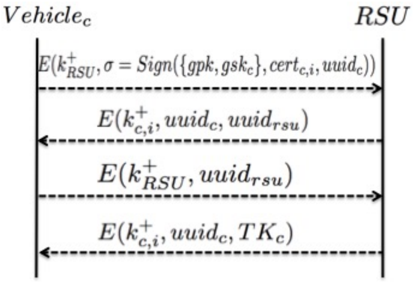

$Vehicle_c$
RSU
$E(k^+_{RSU}, \sigma = Sign(\{gpk, gsk_c\}, cert_{c,i}, uuid_c))$
$E(k^+_{c,i}, uuid_c, uuid_{rsu})$
$E(k^+_{RSU}, uuid_{rsu})$
$E(k^+_{c,i}, uuid_c, TK_c)$

Figure 3: Pseudonym renewal protocol.

protocol, it is used as a nonce value. Moreover, the vehicle v_c encrypts the signature σ using RSU's public key, which generates the authentication request message;

- In Step 2, the RSU checks two parameters: first, it decrypts the received message by using its private key k^-_{RSU}. Then, it verifies if the group revocation token (grt_c) of vehicle v_c is not in the Revocation List (RL) and if the message's timestamp (the time the request was sent) is within a reasonable threshold in order to avoid replay attacks [28]. The *Verify* function represents this process and is formally described in the Equation 1. The CA is responsible for periodically sharing an updated version of RL, which is detailed in Section 2.6.

$$Verify(gpk, \sigma, payload_c) = valid \leftrightarrow grt_c \notin RL \land (tmp - tmp_\sigma \leq threshold_{max}) \tag{1}$$

  If valid, the RSU creates a new UUID value ($uuid_{rsu}$) and returns to the vehicle, which aims at allowing vehicle's authentication and avoiding *man-in-the-middle* attacks. The RSU encrypts the received $uuid_c$ as well as the new UUID value with the i^{th} vehicle public key $k^+_{c,i}$, which is available in the vehicle's i^{th} digital certificate, received in the request message in Step 1. This response message is then sent back to v_c.

- In Step 3, the vehicle checks if the received UUID is the same as it sent previously. If so, v_c returns the received UUID ($uuid_{rsu}$) by encrypting it with the RSU's public key.

- In Step 4, the RSU checks if the proposed UUID ($uuid_{rsu}$) is the same as it had sent in Step 2. If so, the RSU has authenticated vehicle v_c. Hence, the RSU generates the new set of keys $TK_c = \{(k^+_{c,1}/k^-_{c,1}), (k^+_{c,2}/k^-_{c,2}), (k^+_{c,3}/k^-_{c,3})..., (k^+_{c,w}/k^-_{c,w})\}$ for vehicle v_c and authenticates each key. Therefore, we have $Signed^{cert_{c,i}}_{RSU}$:

$$\forall k^+_{c,i} \in TK_c(1 \leq i \leq w), Sign(k^-_{RSU}, k^+_{c,i}) = cert_{c,i}. \tag{2}$$

Finally, the RSU returns the new set of temporary keys TK_c to the vehicle v_c by encrypting the message with the i^{th} v_c's temporary public key received in Step 1, and the v_c's $UUID_c$ value. Vehicle v_c accepts TK_c if, and only if, $UUID_c$ received from RSU is the same as it sent in Step 1. Each key pair and its digital certificate represent a v_c's temporary identity, also called pseudonym.

Vehicles must store all anonymity set digital certificates and temporary keys in a tamper-resistant Hardware Security Module (HSM), also known as tamper-proof device (TPD). This avoids malicious users from copying keying material that belongs to other vehicles.

We now model the pseudonym renewal protocol as a network and detail the authentication property for this network as a trace specification. Within this context, Schneider [29] proposed an extension for CSP and introduced additional control events known as signals. These signal events are used in trace specifications to express the authentication goals of a protocol. Therefore, these signal events will be useful to show the correctness of the pseudonym renewal protocol, as will be discussed in Section 3.2.

The following CSP specification represents a vehicle process during the pseudonym renewal protocol execution. The signal $Running\_Vehicle.v_c.rsu$ indicates that the vehicle v_c is aware of its involvement in a run with the road side unit (rsu) and the nonce $uuid_c$ makes part of this run.

$$\begin{array}{l} Vehicle_c(v_c, uuid_c, cert_{c,i}, gpk, gsk_c) = \\ env?rsu : RoadSide \rightarrow send.v_c.rsu.\{\{uuid_c.cert_{c,i}\}_{Sign(gpk,gsk_c)}\}_{k^+_{RSU}} \rightarrow \\ \mathop{\Box}\limits_{\substack{k^+_{RSU} \in K_{RSU} \\ uuid_{RSU} \in Nonce_c \\ TK_c \subseteq Pseud}} \left(\begin{array}{c} receive.v_c.rsu.\{uuid_c.uuid_{rsu}\}_{k^+_{c,i}} \rightarrow \\ signal.Running\_Vehicle.v_c.rsu \rightarrow \\ send.v_c.rsu.\{uuid_{rsu}\}_{k^+_{RSU}} \rightarrow \\ receive.v_c.rsu.\{uuid_c.TK_c\}_{k^+_{c,i}} \rightarrow \\ signal.Commit\_Vehicle.v_c.rsu \rightarrow STOP \end{array} \right) \end{array}$$

To ensure that all protocol runs use different nonces (*uuids*), we use pairwise disjoint sets $Nonce\_V_a$ and $Nonce\_V_b$ to represent all of the nonces that vehicle v_a might use in a protocol execution in the role of $Vehicle_a$, and a vehicle v_b in the role of $Vehicle_b$, respectively. Therefore, we have the following specification:

$$\begin{aligned} V_{a,b} = {} & |||_{uuid_a \in Nonce\_V_a} Vehicle(a, uuid_a, ...) \\ & ||| \\ & |||_{uuid_b \in Nonce\_V_b} Vehicle(b, uuid_b, ...) \end{aligned}$$

For each *Running* signal, there is a corresponding *Commit* signal. In the Vehicle process, the signal $Commit\_Vehicle.v_c.rsu$ indicates that the vehicle has completed the protocol run and authenticated the communicating *rsu*.

The specification that represents a road side unit during the pseudonym renewal protocol execution is described as the *RoadSide* process. The signal $Running\_RSU.rsu.v_c$ indicates that the current road side unit is aware of its involvement in a run with vehicle v_c and the nonce *uuid* as part of this run. Moreover, the signal $Commit\_RSU.rsu.v_c$ ensures that the road side unit has authenticated the communicating vehicle.

$$\begin{aligned} & RoadSide(rsu, TK_c, uuid_{rsu}, k^+_{RSU}) = \\ & env?v_c : Vehicle \rightarrow receive.rsu.v_c.\{uuid_c.cert_{c,i}\}_{k^+_{RSU}} \rightarrow \\ & \mathop{\Box}_{\substack{cert_{c,i} \in TK_{c'} \\ cert_{c,i} \vdash k^+_{c,i} \\ uuid_c \in Nonce\_V_c}} \left(\begin{array}{c} send.rsu.v_c.\{uuid_c.uuid_{rsu}\}_{k^+_{c,i}} \rightarrow \\ receive.rsu.v_c.\{uuid_{rsu}\}_{k^+_{RSU}} \rightarrow \\ signal.Running\_RSU.rsu.v_c \rightarrow \\ send.rsu.v_c.\{uuid_c.TK_c\}_{k^+_{c,i}} \rightarrow \\ signal.Commit\_RSU.rsu.v_c \rightarrow RoadSide(rsu, TK, uuid, k^+_{RSU}) \end{array} \right) \end{aligned}$$

Based on the requirement that the road side unit must handle more than one protocol run, it is important to specify that for each communicating vehicle, a set of different temporary keys TK will be generated. In order to support many concurrent protocol runs, we define the road side unit as follows:

$$RoadSideUnit(rsu) = \mathop{|||}_{\substack{TK_c \subseteq Pseud \\ uuid_r \in Nonce\_RSU}} RoadSide(rsu, TK_c, uuid, K^+_{RSU})$$

where the set $Pseud$ has infinity temporary public key pairs, and the set

Nonce_RSU includes infinity *uuid* values. The above specifications will be important to prove the correctness of the proposed pseudonym renewal protocol, where a third process, called Intruder, is assumed to be in complete control of the network. We will model our network as a $SYSTEM$ specification, where the process *Vehicle* and *RoadSide* communicate with each other only through an *Intruder* process.

2.4. Sending and Receiving Messages on V2V Communications

From now on, a legitimate vehicle v_c may send messages to other vehicles. Figure 4 depicts the message format and its field goals. Each message is digitally signed with the i^{th} v_c's temporary private key ($k^{-}_{c,i}$), which aims at providing message authenticity and integrity; the *Evn* field is the message's event type (e.g., beacon, accident warning etc.); $cert_{AS_{1,j}}$ is the (current) first level anonymity set digital certificate ($cert_{AS1,j} \in CERT^{t}_{AS_c}$) that the vehicle v_c belongs to (this field allows one or more anonymity set digital certificates, as discussed in Section 2.5); σ is the group signature of data d, which allows privacy-preserving non-repudiation; $cert_{c,i}$ is the i^{th} v_c's public key digital certificate that also allows other vehicles to evaluate the correctness of the message's digital signature; and tmp_{mc} is the timestamp in which the vehicle v_c signs the message m_c, which aims to detect replay attacks.

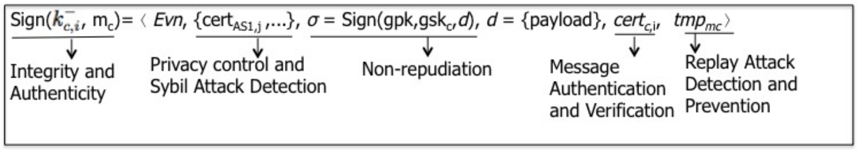

Sign(k⁻c,i, mc)= ⟨ Evn, {certAS1,j ,...}, σ = Sign(gpk,gskc,d), d = {payload}, certc,i, tmpmc ⟩
Integrity and Authenticity
Privacy control and Sybil Attack Detection
Non-repudiation
Message Authentication and Verification
Replay Attack Detection and Prevention

Figure 4: The format of beacon or event-driven messages and the field goals.

The multiple-pseudonym approach proposed herein may use any of the pseudonym change approaches available in the literature [12], [4], [13], [14], [15] and [16]. However, the change of pseudonyms may imply a change on the current anonymity set digital certificates $CERT^{t}_{AS_c}$ that the vehicle v_c must use, as defined on the Property 3 (b) of the Multilevel $ASAP$-V Architecture.

The TPD will not change the current subset from $CERT^{t}_{AS_c}$ to $CERT^{t+1}_{AS_c}$ if the last change occurred in a time less than τ units of time. This decision helps to avoid *sybil* attacks, which comprise the third phase of the $ASAP$-V protocol and is discussed on the next section.

When a vehicle v_a receives a message m_c from a vehicle v_c, it needs to verify m_c in two steps: first, the v_c's $cert_{c,i}$ authenticity; and second, if m_c

is a new message (not originated from a replay attack). In the former case, $cert_{c,i}$ is authentic if, and only if, $cert_{c,i}$ was digitally signed from an RSU, as well as $cert_{c,i}$ is still valid regarding its lifetime. In the second case, m_c is valid if, and only if, v_c signed m_c and if m_c has been uttered only recently.

2.5. The Sybil Attack Detection (Phase 3)

The third phase of the *ASAP*-V protocol is the *sybil* attack detection itself. A *sybil* attack may be explored from malicious users that modify their vehicles to launch the attack. The *sybil* attack is defined as follows:

Definition 1. *In the sybil attack, a vehicle uses multiple identities to disseminate the same false event. This vehicle is the sybil vehicle.*

In VANETs, vehicles disseminate events in order to provide vehicular safety applications. Examples may include accident reporting, an approaching safety vehicle, and electronic emergency braking warnings, to name a few. These events are classified as sporadic events. Beacon messages, may also be an event and are classified as periodic events. An event is defined as follows:

Definition 2. *An event, as defined in [30], is represented as a tuple $\langle evt, l, t\rangle$, where evt is the event type, l and t are the location and the time interval in which the event occurred, respectively.*

The privacy-preserving *sybil* detection phase explores the multilevel anonymity set architecture to detect *sybil* attacks from beacon and event-driven messages. The properties 1 to 3 provide privacy control, while property 4 allows the *sybil* detection process. The basic detection concept is as follows: any two or more messages with different pseudonyms, which disseminate the same event, cannot include the same subset of anonymity set digital certificates. Hence, during a *sybil* attack, the system must evaluate the Equation 3, where V_l is the set of vehicles in the transmission range in a given location l.

$$\forall(v_c, v_{c'}) \in V_l(\, |CERT^t_{AS_c}| = |CERT^t_{AS_{c'}}| \wedge CERT^t_{AS_c} - CERT^t_{AS_{c'}} \neq \emptyset) \quad (3)$$

In beaconing-like messages, if any vehicle v_b receives beacon messages from vehicle v_a, $(v_a, v_b) \in V_l$, and these messages include the same subset $CERT^t_{AS_b}$ of v_b' anonymity set digital certificates, then both vehicles gradually attach, into their beacon messages, the digital certificate of the next

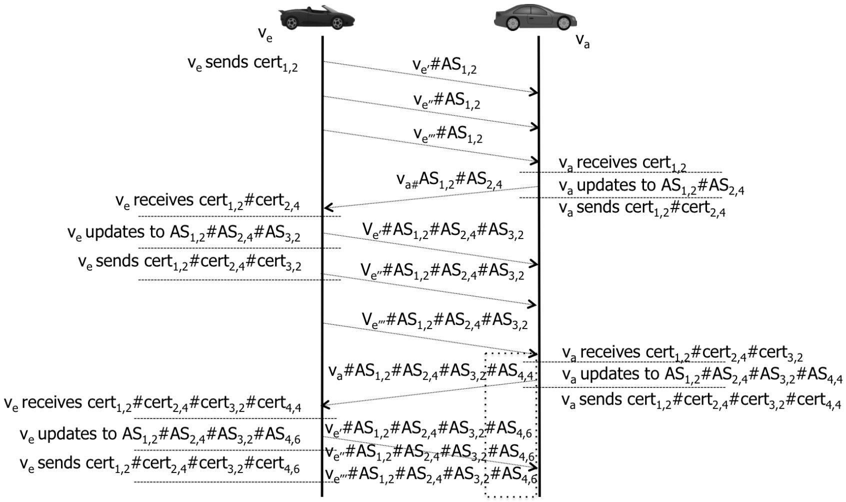

v_e
v_a
v_e sends $cert_{1,2}$
$v_{e'}$#$AS_{1,2}$
$v_{e''}$#$AS_{1,2}$
$v_{e'''}$#$AS_{1,2}$
v_a receives $cert_{1,2}$
$v_{a\#}AS_{1,2}$#$AS_{2,4}$
v_a updates to $AS_{1,2}$#$AS_{2,4}$
v_a sends $cert_{1,2}$#$cert_{2,4}$
v_e receives $cert_{1,2}$#$cert_{2,4}$
v_e updates to $AS_{1,2}$#$AS_{2,4}$#$AS_{3,2}$
$V_{e'}$#$AS_{1,2}$#$AS_{2,4}$#$AS_{3,2}$
v_e sends $cert_{1,2}$#$cert_{2,4}$#$cert_{3,2}$
$V_{e''}$#$AS_{1,2}$#$AS_{2,4}$#$AS_{3,2}$
$V_{e'''}$#$AS_{1,2}$#$AS_{2,4}$#$AS_{3,2}$
v_a receives $cert_{1,2}$#$cert_{2,4}$#$cert_{3,2}$
v_a#$AS_{1,2}$#$AS_{2,4}$#$AS_{3,2}$#$AS_{4,4}$
v_a updates to $AS_{1,2}$#$AS_{2,4}$#$AS_{3,2}$#$AS_{4,4}$
v_a sends $cert_{1,2}$#$cert_{2,4}$#$cert_{3,2}$#$cert_{4,4}$
v_e receives $cert_{1,2}$#$cert_{2,4}$#$cert_{3,2}$#$cert_{4,4}$
v_e updates to $AS_{1,2}$#$AS_{2,4}$#$AS_{3,2}$#$AS_{4,6}$
$v_{e'}$#$AS_{1,2}$#$AS_{2,4}$#$AS_{3,2}$#$AS_{4,6}$
v_e sends $cert_{1,2}$#$cert_{2,4}$#$cert_{3,2}$#$cert_{4,6}$
$v_{e''}$#$AS_{1,2}$#$AS_{2,4}$#$AS_{3,2}$#$AS_{4,6}$
$v_{e'''}$#$AS_{1,2}$#$AS_{2,4}$#$AS_{3,2}$#$AS_{4,6}$

Figure 5: Detecting *sybil* attacks from beacon messages. To detect a *sybil* attack, messages with different pseudonyms carry the same subset of anonymity set digital certificate.

deeper anonymity set until the Equation 3 is satisfied, where all messages are distinguishable among each other.

Figure 5 depicts the *sybil* attack detection. The malicious vehicle v_e fires a *sybil* attack by sending 3 different beacon messages that describe 3 location points. Since beacon messages are broadcasted to all vehicles in the transmission range, the vehicle v_a observes that other vehicle(s) is (are) transmitting messages that contain the same first level anonymity set digital certificate ($AS_{1,2}$). v_a attaches the next anonymity set digital certificate (e.g.: $cert_{2,4}$) and sends it on its next beacon message.

When v_e receives v_a's message, it must have to update to the next two levels. Since v_e knows that it also belongs to anonymity set $AS_{2,4}$, it also must include the third anonymity set digital certificate (e.g.; $cert_{3,2}$), otherwise vehicles in the vicinity will only drop the messages and the *sybil* attack has no effect. On the other hand, the other vehicles (e.g.: v_b, v_c, v_d, v_f, v_g) only store v_e's and v_a's messages. After receiving v_e's messages, v_a attaches the fourth level anonymity set digital certificate (e.g.: $cert_{4,4}$) and sends it all together. Finally, the malicious vehicle sends its fourth level anonymity set digital certificate (e.g.: $cert_{4,6}$).

Note that if a malicious vehicle v_e sends messages with multiple identities, these messages will always carry the same set of anonymity set digital certificates. That is, only messages from identities $v_{e'}$, $v_{e''}$ and $v_{e'''}$ do not satisfy Equation 3. Therefore, vehicles v_a, v_b, v_c, v_d, v_g and v_f store the messages from v_e as a set of n messages $M_{e,n}$, which is used for prosecution purpose (Phase 4, detailed in Section 2.6).

After a short time interval receiving messages with different identities, but still containing the same anonymity set digital certificates, the legitimate vehicles may conclude that the messages with these identities come from a *sybil* node (e.g.: v_e). Equation 4 defines this short time interval that the vehicles v_a, v_b, v_c, v_d, v_g and v_f must wait for vehicle v_e to send the next anonymity set digital certificate after receiving the last one.

The time interval is evaluated for each group of messages that contain the $AS_{1,j}$'s anonymity set digital certificate. The m variable is the maximum number of anonymity set levels, pm is the number of anonymity set digital certificates already presented by the target vehicle, and V_l is the number of neighboring nodes in the transmission range. Therefore, after $\delta_{AS_{1,j}}$ ms after receiving the last anonymity set digital certificate, the vehicles v_a, v_b, v_c, v_d, v_g and v_f may evaluate the vehicle v_e as a *sybil* node.

$$\delta_{AS_{1,j}} = beacon\ interval + (m - pm) * V_l/m \tag{4}$$

Equation 4 defines a dynamic behavior in which $\delta_{AS_{1,j}}$ must be as high as the number of vehicles in the transmission range is high, but, in order to minimize the impact of the *sybil* attack, $\delta_{AS_{1,j}}$ smoothly decreases as the number of presented anonymity set digital certificates per level within the beacon messages increases. A vehicle evaluates Equation 4 for each new beacon message that it receives and contains the anonymity set digital certificate $cert_{AS_{1,j}}$. If a malicious vehicle try to evade the *sybil* attack after it had been fired and before achieving the last level of possible anonymity sets, then the $\delta_{AS_{1,j}}$ time interval will timeout and the sample suspected messages will be used in the prosecution phase.

On the other hand, hidden terminal scenarios (caused by fading) may lead to false-negative *sybil* detections. To avoid hidden terminals in *ASAP*-V protocol, we propose the *sybil attack signaling message*, which we also call as *First Level Warning* (FLW) message. It aims at allowing one vehicle to announce to neighboring vehicles that there are two or more vehicles in the vicinity that belong to the same first level anonymity set. This signaling message aims to avoid *false-positive* detections in hidden terminals scenarios. For instance, as depicted in Figure 6, vehicle B (v_b) broadcasts signaling messages with identities A and C. Once all vehicles may listen to the *signaling* messages in the broadcast channel, the vehicles A and C may detect that they are suspected and may attach the next anonymity set digital certificates in their next beacon messages.

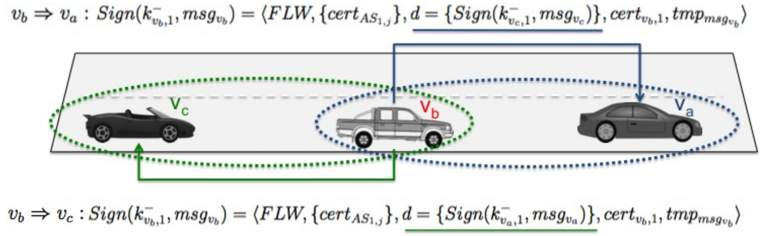

$v_b \Rightarrow v_a : Sign(k^-_{v_b,1}, msg_{v_b}) = \langle FLW, \{cert_{AS_{1,j}}\}, \underline{d = \{Sign(k^-_{v_c,1}, msg_{v_c})\}}, cert_{v_b,1}, tmp_{msg_{v_b}} \rangle$
v_c
v_b
v_a
$v_b \Rightarrow v_c : Sign(k^-_{v_b,1}, msg_{v_b}) = \langle FLW, \{cert_{AS_{1,j}}\}, \underline{d = \{Sign(k^-_{v_a,1}, msg_{v_a})\}}, cert_{v_b,1}, tmp_{msg_{v_b}} \rangle$

Figure 6: Vehicle v_b sends FLW message as *signaling message* to vehicles v_a e v_c.

To detect if two vehicles v_a and v_c are hidden terminals to each other, one vehicle (e.g: v_b) must evaluate if their transmission signals do not reach the other one. Let P_{x,pos_y} be the power of the transmission signal of vehicle v_x at position y. Thus, we must evaluate if $P_{a,pos_c} < P_{min}$ and $P_{c,pos_a} < P_{min}$, where P_{min} is the minimum power required to receive a beacon message successfully.

To evaluate P_{a,pos_c} and P_{c,pos_a}, we first need to estimate the initial power of the signals transmitted from v_a and v_c, that is, P_{a,pos_a} and P_{c,pos_c}. Let α be a constant associated to the exponential decay of the power of the electromagnetic wave as the signal travels along the communication channel in a dissipative dielectric, and $d(_{v_x,v_y})$ the euclidean distance between vehicles v_x and v_y.

$$P_{a,pos_b} = P_{a,pos_a} \cdot e^{-\alpha \cdot d(_{v_a,v_b})} \quad \text{(The transmitted signal power of } v_a \text{ at } v_b\text{'s position.)}$$

$$P_{a,pos_a} = P_{a,pos_b} \cdot e^{\alpha \cdot d(_{v_a,v_b})} \quad \text{((2) The initial transmitted signal power of } v_a\text{)}$$

$$P_{c,pos_b} = P_{c,pos_c} \cdot e^{-\alpha \cdot d(_{v_c,v_b})} \quad \text{(The transmitted signal power of } v_c \text{ at } v_b\text{'s position)}$$

$$P_{c,pos_c} = P_{c,pos_b} \cdot e^{\alpha \cdot d(_{v_c,v_b})} \quad \text{((1) The initial transmitted signal power of } v_c\text{)}$$

$$P_{a,pos_c} = P_{a,pos_a} \cdot e^{-\alpha \cdot d(_{v_a,v_c})} \quad \text{((3) The transmitted signal power of } v_a \text{ at } v_c\text{'s position)}$$

$$P_{c,pos_a} = P_{c,pos_c} \cdot e^{-\alpha \cdot d(v_c, v_a)}$$

((4) The transmitted signal power of v_c at v_a's position)

Therefore, applying (1) to (3) and (2) to (4), the transmitted signal power of v_a at v_c's position, and the v_c at v_a's position are defined in Equations 5 and 6, respectively:

$$P_{a,pos_c} = P_{a,pos_b} \cdot e^{\alpha \cdot d(v_a, v_b)} \cdot e^{-\alpha \cdot d(v_a, v_c)} \tag{5}$$

$$P_{c,pos_a} = P_{c,pos_b} \cdot e^{\alpha \cdot d(v_c, v_b)} \cdot e^{-\alpha \cdot d(v_c, v_a)} \tag{6}$$

Thus, if $P_{a,pos_c} < P_{min}$ and $P_{c,pos_a} < P_{min}$, vehicle v_b must send FLW messages to vehicles v_a and v_c. Hence, both vehicles may attach their next anonymity set digital certificates. Since FLW messages are sent in a broadcast manner, all other vehicles in the vicinity will also receive v_b's FLW message. This approach avoids multiple FLW messages to the same scenario.

In order to detect *sybil* attacks from event-driven messages, each vehicle v_i must attach all current anonymity set digital certificates $(CERT^t_{AS_i})$ in the message. Suppose the vehicle v_a reports an emergency braking alert (EBBL). Hence, the messages would be as follows: $Sign(k^-_{a,1}, m_a) = \langle$EBBL, $(\boldsymbol{cert_{AS_{1,2}}}, \boldsymbol{cert_{AS_{2,4}}}, \boldsymbol{cert_{AS_{3,2}}}, \boldsymbol{cert_{AS_{4,4}}}, \boldsymbol{cert_{AS_{5,1}}}), \sigma = Sign(gpk, gsk_a, d)$, $d = ..., cert_{a,1}, tmp_{m_a}\rangle$. According to property 4 of the multilevel anonymity sets architecture, it is impossible for two different vehicles to announce the same event with the same *anonymity set digital certificates.* This approach definitely avoids a *sybil* attack from event-driven messages without compromising the privacy of the vehicles.

2.6. Sybil Attack Prosecution Phase (Phase 4)

Once a misbehaved vehicle v_e is detected, all other vehicles v_i in l store v_e's messages as a set of sample n suspected messages $M_{e,n}$. Thus, a prosecution protocol is executed.

In Step 1, each vehicle v_i generates a digitally signed prosecution message and sends it to the nearby RSU. Each vehicle v_i that detected the *sybil* vehicle (e.g.: v_a and v_b) uses it's group signing key (e.g.: gsk_a and gsk_b) to digitally sign the prosecution message, and also attach the sample of suspected messages evaluated as *sybil* messages (e.g.: $m_{e,1}$, $m_{e,2}$, $m_{e,3}, ..., m_{e,n} \in M_{v_e,n}$). In Step 2, the RSU forwards the received prosecution messages to the CA. In

Step 3, the CA extracts the suspected messages from the prosecution message and traces their owners using its *gmsk* key. If all n messages describe the same event *evt* and are originated from the same vehicle v_e (it digitally signed all messages), then the CA resolves each message to the same unique vehicle identity (i.e.: a *sybil* vehicle). Equation 7 formally describes the verification process. Finally, the CA inserts the malicious vehicle's group revocation token (e.g.: grt_e) into the revocation token list (RL), and sends the RL to all RSUs in Step 4.

$$TraceAll(gmsk, M_{v_e,n}) = \forall m_i, m_j \in M_{v_e,n}, (evt_i \in m_i = evt_j \in m_j) \wedge \\ (Signed^{\sigma_i}_{v_e} \wedge Signed^{\sigma_j}_{v_e}) \leftrightarrow sybil_{v_e}\{i, j, n \in \mathbb{N} : 1 \leq i \leq n, 1 \leq j \leq n, i \neq j\} \tag{7}$$

3. Protocol Analysis and Experimental Results

This section presents the experimental results of the proposed solution. First, we analyze the management, storage, computation, and communication overheads; mainly w.r.t. the cryptography key management and its processing. Afterwards, we show the correctness verification of the pseudonym renewal protocol, as well as an analysis of the proposed anonymous communication model. Finally, we present the simulation results of the proposed *sybil* detection approach.

3.1. Management, Storage, Computation and Communication Overheads

In spite of the number of security properties involved, the following overhead analysis shows that $ASAP$-V is a lightweight protocol within the context of VANETs.

- *Management overhead*: the CA is only responsible for managing the anonymity set digital certificates and the group signing keys, which do not change frequently. In addition, vehicles must only manage the pseudonyms renewal that requires minimal changes;

- *Storage overhead*: the CA stores the anonymity set digital certificates, which takes $n * m * 56$ bytes long using a 224bits Elliptic Curve Digital Signature Algorithm (ECDSA) [31], the group public key with size $O(log\,|V|)$, which takes $|V| * 800$ bytes long, and the group membership certificate of size $O(1)$, which takes $|V| * 64$ bytes long using Group Signatures with Almost-for-free Revocation (GSAFR) [32]. It is important to note that the CA does not need to store vehicle pseudonyms, which reduces the storage overhead found in other works, such as the Zhou's approach [30] and [33]. Moreover, the RSUs store the Revocation List of size *O(log r)* (which contains each vehicle's group revocation tokens), which is also small when compared with traditional revocation lists of the public key infrastructure (which stores all non-valid public keys).

- *Computation Overhead*: we implemented our security algorithms on a 2.9 GHz Intel Core i7 processor with 8 GB of RAM, for V2V and V2I communications:

  - On V2V communication: to sign a message in a V2V communication, a vehicle v_c first signs the payload d (σ) with its group signing key gsk_c using GSAFR, which takes 11 *ms* with computation and size of cost $O(1)$; and afterwards, the whole message with its i^{th} temporary private key $k^{-}_{c,i}$, which takes 0.1 *ms* using ECDSA. Thus, v_c takes 11.1 *ms* to sign the whole message. On the other hand, when another vehicle v_e receives the message, it verifies the message's authentication in two steps: first it verifies the sender's (v_c) public key $k^{+}_{c,i}$ authenticity, which is available in the digital certificate $cert_{c,i}$; and second, the whole message authentication itself. Thus, a vehicle must first check the $cert_{c,i}$ authentication using the RSU's public key k^{+}_{RSU}, which takes 0.4 *ms*, and then the whole message's authentication, which also takes 0.4 *ms*. The total message verification process takes 0.8 *ms*. In short, a vehicle may sign 90 messages/s, while it may verify 1250 messages/s (or 2500 messages/s after checking the first time).
  - On V2I communication: a vehicle signs the payload data (UUID and the v_c's i^{th} digital certificate) with its group signing key, which takes 16 ms (with computation cost of $O(log\, 1)$), and the request message m_c with ECDSA, which takes 1 ms (Step 1); when a RSU receives the message m_c (Step 2), it checks the group signature authentication in 132 *ms*, with computation cost of $O(1)$, while it generates each w key pair in w*83 *ms*, and signs w key pairs that takes w*1 *ms*. Hence, the total computation cost is 132 *ms* + w*83 *ms* + w*1 *ms*.

- *Communication Overhead*:
    - On V2V communication: the beacon message size basically requires one anonymity set digital certificate $cert_{AS_{1,j}}$, which takes 56 bytes in a $224bit$ ECDSA; the group signature σ of the payload d, which takes 225 bytes (128 bits security level) with signature size of $O(1)$; the i^{th} temporary digital certificate $cert_{v,i}$, which takes 56 bytes; and finally, the whole message authentication, which also takes 56 bytes. The minimum message size to be transmitted is 393 bytes. On the other hand, as the number of anonymity set digital certificates increases due to the *sybil* detection phase, the message size is 56 bytes longer.
    - On V2I communication: during Phase 2, a vehicle v_c signs the pseudonym renewal request message including a group signature, which takes 225 bytes (128 bits security level) with signature size of $O(1)$; and attaches the i^{th} temporary digital certificate, taking 56 bytes. Hence, the total message size is 281 bytes. The RSU response includes the new set of temporary key pairs TK_v of size w, which is $w * 56$ bytes longer, as well as the message authentication, which also takes 56 bytes. Thus, the total response size is $w * 56 + 56$ bytes.

3.2. Correctness Verification of the Pseudonym Renewal Protocol

In this section, we formally verify the correctness of the pseudonym renewal protocol (Phase 2). In Section 2.3, we described the pseudonym renewal protocol with a trace specification that the system needs to satisfy. In order to verify its correctness, we used the ranking function ρ proposed by Schneider [34], which is also described in this section.

Consider an intruder that has complete control of the channels *send* and *receive*. Thus, it has the capabilities of blocking, replaying, spoofing and manipulating any messages that appear on any of the public channels in the network. An intruder may be a malicious vehicle, for instance, that monitors the communication channels and can see all messages begin transmitted through the channels *send* and *receive*.

Within this context, let *Intruder* to denote the intruder process. For each participant $a \in \mathcal{U}$ (e.g.: vehicle or RSU) that sends and receives messages through the channels, a CSP process $PART_a$ represents the behavior of the participant. We define the complete network $SYSTEM$ as

$$SYSTEM = (|||_{a \in \mathcal{U}} PART_a) \underset{(send,receive)}{\|} Intruder \tag{8}$$

where all participants $\mathcal{U}$ synchronizes with *Intruder* on *send* and *receive* channels. Finally, let the symbol $\vdash$ denote the *generate relation*, as proposed by Schneider [29], to represent what messages m may be generated from a given set of messages S (e.g.: $S \vdash m$). In this case, the $\vdash$ relation is used to define a recursive definition of *Intruder* as follows:

$$\begin{aligned} Intruder(S) = send.a.b.m \rightarrow Intruder(S \cup m)\Box \\ \Box_{a,b \in \mathcal{U}, S \vdash m} receive.a.b.m \rightarrow Intruder(S) \end{aligned} \tag{9}$$

The *Intruder* process receives a set of messages S that is in the possession of the intruder. The definition of Intruder models the behavior of an intruder such that it may wish to block, spoof or manipulate some messages, as well as it allows the intruder to possess any initial public knowledge about the network such as vehicles' and RSUs' identities and their respective digital certificates. Schneider denotes IK as the set of initial knowledge of the intruder, therefore, we have Intruder(IK).

Our proof strategy is based on Schneider's approach, where a trace specification that denotes the authentication property needed to be satisfied by *SYSTEM.* Our first observation is that when the signal *Commit* appears in the *SYSTEM*, the correspondent *Running* signal must come beforehand. Thus, let R be the set of Running signals, and T be the set of Commit signals. The authentication property is given as R **precedes** T. Equation 10 summarizes the condition for RSU's authentication of Vehicles and for Vehicle's authentication of RSU, respectively.

$$\begin{aligned} \textbf{SYSTEM sat } Running.v_c.rsu_q.uuid_{rsu_q} \textbf{ precedes } Commit.rsu_q.v_c.uuid_{rsu_q} \\ \textbf{SYSTEM sat } Running.rsu_q.v_c.uuid_c \textbf{ precedes } Commit.v_c.rsu_q.uuid_c \end{aligned} \tag{10}$$

If *SYSTEM* can be proved to satisfy such specification, then the protocol is proved correct for the property of authentication. Therefore, in order to achieve this, we adopt the following strategy: if the signal *Running* is prevented from occurring in *SYSTEM*, then the following signal *Commit* is not possible in *SYSTEM.* Thus, *Commit* should not appear in any trace *tr* of $SYSTEM \underset{Running}{\parallel} STOP$ in both authentication sides. Hence, we formally have Equation 11.

$$SYSTEM \underset{Running}{\parallel} Stop \ \mathbf{sat} \ tr \upharpoonright Commit = \langle\rangle \tag{11}$$

From now on, we can construct the rank function ρ for the pseudonym renewal protocol and evaluate the different conditions provided in the rank function theorem to verify the correctness of the protocol.

Let $\mathcal{U}$ be the set of vehicles' and RSUs' identities ($\forall cert_{c,i} \in TK_c$), $\mathcal{N}$ be the set of all possible nonces (UUIDs) and $\mathcal{K}$ be the set of all public key pairs $(K^+_{c,i}, K^-_{c,i}) \in TK_c$. The set of all such atoms is $\mathcal{A} = \mathcal{U} \cup \mathcal{N} \cup \mathcal{K}$. Moreover, consider a message space $\mathcal{M}$ that contains all the messages and signals that appear during the pseudonym renewal protocol execution, such that $m \in \mathcal{A} \Rightarrow m \in \mathcal{M}$. The rank function ρ maps events and messages to integers, that is, $\rho : \mathcal{M} \to \mathbb{Z}$. Therefore, we divided that message space into two parts:

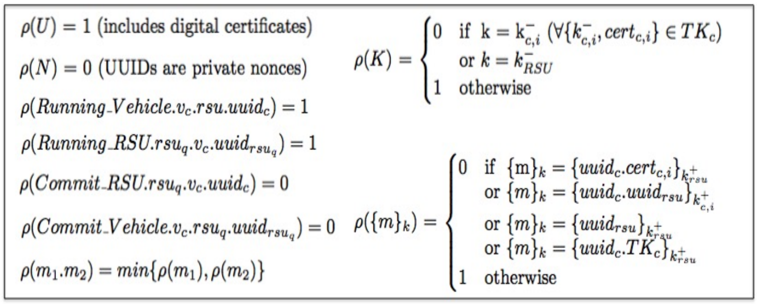
1. $\mathcal{M}_{p^-} = \{m \in \mathcal{M} | \rho(m) \leq 0\}$: this part assigns a non-positive rank, which means that an intruder should never get hold of message m;
2. $\mathcal{M}_{p^+} = \{m \in \mathcal{M} | \rho(m) > 0\}$: this part assigns a positive rank that aims at allowing an intruder to get hold of message m without compromising the protocol.

In other words, it is desirable for a process P to never transmit a message of non-positive rank, unless P has previously received a message with a non-positive rank. More formally, for a process P,

$$P \ \boldsymbol{maintains} \ \rho \Leftrightarrow \forall \, tr \in traces(P) \cdot \rho(tr \Downarrow receive) > 0 \Rightarrow \rho(tr \Downarrow send) > 0 \tag{12}$$

which means that P will never transmit any message m of $\rho(m) \leq 0$ unless P has received some message *m'* of $\rho(m') \leq 0$ previously. Since the communication channel is public - and the intruder can control it - any message that flows through the channels must be of positive rank, otherwise, if messages with non-positive rank is sent, then the intended secrecy of the message is compromised.

Figure 7 presents a rank function for the proposed pseudonym renewal protocol. The Rank Function Theorem proposes four properties that if the rank function (and so the underlying *SYSTEM*) satisfies these properties, then no messages of non-positive rank can circulate in $SYSTEM \underset{R}{\|} Stop$. For instance, an intruder can not send ilegal messages from its IK nor from messages it sees during the protocol execution. Moreover, honest nodes maintains ρ while being restricted on Running signal. On the other hand, the failure of a rank function to satisfy the conditions of the theorem may imply a flaw in the protocol.



$\rho(U) = 1$ (includes digital certificates)

$\rho(N) = 0$ (UUIDs are private nonces)

$\rho(Running\_Vehicle.v_c.rsu.uuid_c) = 1$

$\rho(Running\_RSU.rsu_q.v_c.uuid_{rsu_q}) = 1$

$\rho(Commit\_RSU.rsu_q.v_c.uuid_c) = 0$

$\rho(Commit\_Vehicle.v_c.rsu_q.uuid_{rsu_q}) = 0$

$\rho(m_1.m_2) = min\{\rho(m_1), \rho(m_2)\}$

$$\rho(K) = \begin{cases} 0 & \text{if } \mathrm{k} = \mathrm{k}^-_{c,i}\ (\forall \{k^-_{c,i}, cert_{c,i}\} \in TK_c) \\ & \text{or } k = k^-_{RSU} \\ 1 & \text{otherwise} \end{cases}$$

$$\rho(\{m\}_k) = \begin{cases} 0 & \text{if } \{\mathrm{m}\}_k = \{uuid_c.cert_{c,i}\}_{k^+_{rsu}} \\ & \text{or } \{m\}_k = \{uuid_c.uuid_{rsu}\}_{k^+_{c,i}} \\ & \text{or } \{m\}_k = \{uuid_{rsu}\}_{k^+_{rsu}} \\ & \text{or } \{m\}_k = \{uuid_c.TK_c\}_{k^+_{rsu}} \\ 1 & \text{otherwise} \end{cases}$$

Figure 7: A Rank function for the proposed pseudonym renewal protocol.

The properties of the Rank Function Theorem are as follows. At each property, we describe the analysis of the pseudonym renewal protocol.

- Property 1 - $\forall\, m \in IK \cdot \rho(m) > 0$: this property states that the intruder knowledge may only have positive rank. In our case, the set IK contains all public digital certificates, such as the i^{th} digital certificate $cert_{c,i}$ of any vehicle v_c or $cert_{RSU}$, that all correspond to public keys (and also represent vehicle identities or pseudonyms). There is nothing in this set that is of non-positive rank. Therefore, the condition is satisfied;

- Property 2 - $\forall\, S \subseteq \mathcal{M}, m \in \mathcal{M} \cdot ((\forall\, m' \in S \cdot \rho(m') > 0) \wedge S \vdash m) \Rightarrow \rho(m) > 0$: this property states that a set of positive rank messages may only generate positive rank messages. In our case, any positive rank message allows an intruder to guess a non-positive rank message. The four messages of non-positive rank - in the subcases $\rho(\{m\}_k)$ are encrypted under public keys, which their correspondent private keys are also non-positive. This avoids the *Intruder* from sending these four messages - and also from find out the nonce values, which are also non-positive. Thus, this condition is also satisfied;

- Property 3 - $\forall\ t \in T \cdot \rho(t) \leq 0$: this property states that none of the events in T can be of positive rank. In our case, both signal events $Commit\_RSU.rsu_q.v_c.uuid_c \in T$ and $Commit\_Vehicle.v_c.rsu_q.uuid_{rsu_q} \in T$ are of non-positive rank. Hence, this condition is satisfied;

- Property 4 - $\forall a \in \mathcal{U} \cdot PART_a \underset{R}{\|} Stop$ **sat** maintain positive ρ: this property states that every process in the *SYSTEM* needs to maintain ρ while being restricted on the events in set R. In our case, $Running\_Vehicle.v_c.rsu.uuid_c \in R$ and $Running\_RSU.rsu_q.v_c.uuid_{rsu_q} \in R$. Thus, we need to verify if the two communicating process maintain ρ. The restriction on *Vehicle* process is as follows:

$$\begin{array}{ll} Vehicle_c \underset{Running\_Vehicle.v_c.rsu_q}{\|} Stop = & \Box_b \\ & send.v_c.rsu_b.\{uuid_c.cert_{c,i}\}_{k^+_{RSU}} \rightarrow \\ & receive.v_c.rsu_b.\{uuid, uuid_{rsu}\}_{k^+_{c,i}} \rightarrow \\ & if\ rsu_b = rsu_q \wedge uuid = uuid_c\ then \\ & STOP \\ & else\ Running\_Vehicle.v_c.rsu_q.uuid_{rsu} \rightarrow \\ & send.v_c.rsu.\{uuid_{rsu}\}_{k^+_{RSU}} \rightarrow STOP \end{array}$$

In the choice operator $\Box_b$, b represents the other participants that vehicle v_c may communicate with. According to the modified process described above, if $uuid \neq uuid_c$, then the vehicle v_c is not enabled to transmit $\{uuid_{rsu}\}_{k^+_{RSU}}$. Hence, the RSU will never run the correspondent commit signal ($Commit\_RSU.rsu_q.v_c.uuid_{rsu}$). Therefore, *SYSTEM* maintains positive rank and the vehicle authenticates the current RSU. In order to RSU authenticate a given vehicle v_c, the same modified process is made in the *RoadSide* process, such as follows.

$$\begin{array}{ll} RoadSide_q \underset{Running\_RSU.rsu_q.v_c}{\|} Stop = & \Box_b \\ & receive.rsu_q.v_b.\{uuid_b, cert_{b,i}\}_{k^+_{RSU}} \rightarrow \\ & send.rsu_q.v_b.\{uuid_b.uuid_{rsu_q}\}_{k^+_{b,i}} \rightarrow \\ & receive.rsu_q.v_b.\{uuid_b, uuid\}_{k^+_{RSU}} \rightarrow \\ & if\ v_b = v_c \wedge uuid = uuid_{rsu_q}\ then \\ & STOP \\ & else\ Running\_RSU.rsu_q.v_b.uuid_b \rightarrow \\ & send.rsu_q.v_b.\{uuid_b.TK_b\}_{k^+_{b,i}} \rightarrow STOP \end{array}$$

As previously detailed, in the choice operator $\Box_b$, b represents the other participants that RSU rsu_q may communicate with. if $uuid \neq uuid_{rsu_q}$, then the RSU rsu_q is not enabled to transmit $\{TK_b\}_{k^+_{b,i}}$. Hence, the vehicle will never run the correspondent commit signal ($Commit\_Vehicle.v_c.rsu_q.uuid_c$). Therefore, $SYSTEM$ maintains positive rank and the RSU authenticates the current vehicle v_c.

Finally, as detailed in Equation 11, the $SYSTEM$ is proved to satisfy such specification, and the protocol is proved to be correct for the property of authentication. As seen in the analysis above, when the signal *Running* is prevented from occurring in $SYSTEM$, then the following signal *Commit* was not possible in $SYSTEM$. Thus, *Commit* does not appear in any trace *tr* of $SYSTEM$.

3.3. Analysis of Anonymous Communication

The *anonymity* of a vehicle means that the vehicle is not identifiable within a set of vehicles, the vehicles' anonymity set. A system with N active vehicles, the maximum degree of anonymity is achieved when an eavesdropper sees all vehicles equally probable as being the originator of a message. Therefore, we applied a normalized Shannon's Entropy method [35] in order to quantify the uncertainty of information and to evaluate the degree of anonymity of the vehicles in a geographical area.

We compare the entropy of the anonymity set compared to the maximum entropy of the system after a vehicle exposed its i^{th} level anonymity set digital certificate. Therefore, we compare how distinguishable this vehicle is within the set of possible vehicles if an eavesdropper sees its network messages in a given location.

Equation 13 defines the maximum entropy $H^M_{AS_{i,j}}$ of a given vehicles' anonymity set $AS_{i,j}$. Let $N_{AS_{1,j}}$ be the number of vehicles in the anonymity set $AS_{1,j}$ (first level).

$$H^M_{AS_{1,j}} = log_2(N_{AS_{1,j}}) \tag{13}$$

Equation 14 defines the anonymity set entropy $H^X_{AS_{i,j}}$ after a vehicle exposed its i^{th} level anonymity set digital certificate. An eavesdropper assigns p_{v_c} as the probability that a vehicle v_c sent a specific message.

$$H^X_{AS_{i,j}} = -\sum_{k=1}^{N} log_2(p_{v_c}) \tag{14}$$

The information the eavesdropper has learned after observing the i^{th} anonymity set digital certificate is $H^M_{AS_{1,j}} - H^X_{AS_{i,j}}$. We divide by $H^M_{AS_{1,j}}$ to normalize the value. Therefore, Equation 15 defines the degree of anonymity $d_{AS_{i,j}}$ of a specific vehicles' anonymity set $AS_{i,j}$:

$$d_{AS_{i,j}} = 1 - \frac{H^M_{AS_{1,j}} - H^X_{AS_{i,j}}}{H^M_{AS_{1,j}}} = \frac{H^X_{AS_{i,j}}}{H^M_{AS_{1,j}}} \tag{15}$$

The degree of anonymity $d_{AS_{i,j}}$ ranges between 0 - when a vehicle appears as being the originator of messages with probability 1 - and 1 - when all vehicles that belong to the anonymity set $AS_{i,j}$ appear as being the originator with the same probability.

Table 2 presents an analysis of the proposed anonymous communication. We considered 80 million[3] vehicles, 420 groups/level and each vehicle in 20 groups/level. The *number of vehicles together per group* means that those vehicles in the same anonymity set of the first level ($AS_{1,j}$) are still together in level i. Therefore, when $i = 6$, all vehicles satisfy property 4 of the proposed multilevel architecture.

Table 2: Simulation parameters.

Level	N° of vehicles together	$H^X_{AS_{i,j}}$	$d_{AS_{i,j}}$
$i = 1$	3.809.524	21.87	1.00
$i = 2$	181.405	17.47	0.79
$i = 3$	8.638	13.08	0.59
$i = 4$	412	8.7	0.39
$i = 5$	19	4.4	0.20
$i = 6$	≈ 1	-0.10	0.00

When a vehicle v_c sends a message with its first level anonymity set digital certificate, its degree of anonymity is equal to 1, which means that if an attacker eavesdrops on the wireless channel, all vehicles in that group $AS_{1,j}$ appear as being the originator of the message with the same probability. As long as a vehicle v_c attaches its i^{th} anonymity set digital certificate on the beacon messages, the system exposes v_c's anonymity ($d_{AS_{i,j}}$) smoothly.

[3] According to the Brazilian's Natinal Traffic Department, at the end of 2014, this number includes cars, motorcycles and buses.

When v_c sends all current anonymity set digital certificates ($CERT^t_{AS_c}$) in a given time interval t, its anonymity degree is equal to zero, and v_c appears as being the originator of that message with probability 1.

On the other hand, when vehicle v_c exposes one anonymity set digital certificate per level ($CERT^t_{AS_c}$), it only exposed part of its anonymity. Vehicle v_c may select another combination of current anonymity set digital certificates $CERT^{t'}_{AS_c}$ among all 20^6 possibilities (for this scenario). This approach makes vehicle's privacy violation a difficult task. In addition, the probability that any two vehicles v_a and v_b in $AS_{1,j}$ will choose the same digital certificate of the $m-1$ lower levels is $\prod_{i=1}^{m-1} \frac{1}{20}$, which is very low. Therefore, the probability that a vehicle v_c will expose its m anonymity set digital certificates is also very low.

3.4. Sybil Attack Detection Evaluation Results

The *sybil* detection evaluation aimed to answer three questions:

1. What is the average time one $v_k \in V_l$ takes to decide that messages are from two different vehicles, and not from a (potential) *sybil* node?
2. What is the average time one vehicle takes to detect a *sybil* attack from *beacon* messages?
3. What are the false-positive (a legitimate node is evaluated as *sybil*) and false-negative (a *sybil* node was not evaluated as one) detection rates?

The simulation was performed by using the Veins simulation environment, Table 3 summarizes the simulation parameters.

Table 3: Simulation parameters.

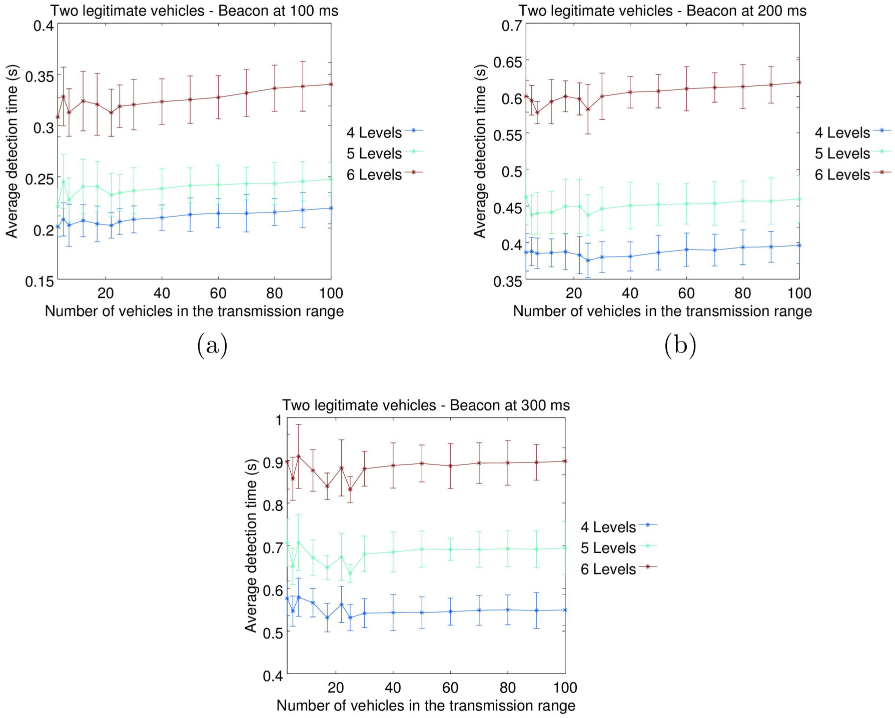
Parameter	Value
Total number of executions	900
Simulation Duration	between 10s to 120s
MAC and PHY protocols	802.11p
Transmission Power	20mW
Bit rate	18Mbps
Beacon rates	3, 5, and 10 beacon/s
Number of Vehicles	3, 5, 7, 12, 17, 22, 25, 30, 40, ..., 100
Mobility model	Krauss

continued on next page

Table 3 – *continued from previous page*

Parameter	Value
Average vehicle speed	between 15 m/s and 22 m/s
Anonymity set levels (m)	4, 5 and 6

Figures 8 and 9 illustrate the experimental results (average time on 95% confidence intervals) for the first and the second questions, considering 100 *ms*, 200 *ms* and 300 *ms* beacon message transmission intervals. This experimental approach is based on the assumption that it is possible to send beacons as frequently as possible but without overloading the communication channel [36]. The solution adaptively updates the beacon frequency based on the importance of messages and based on the available capacity of the wireless channel.



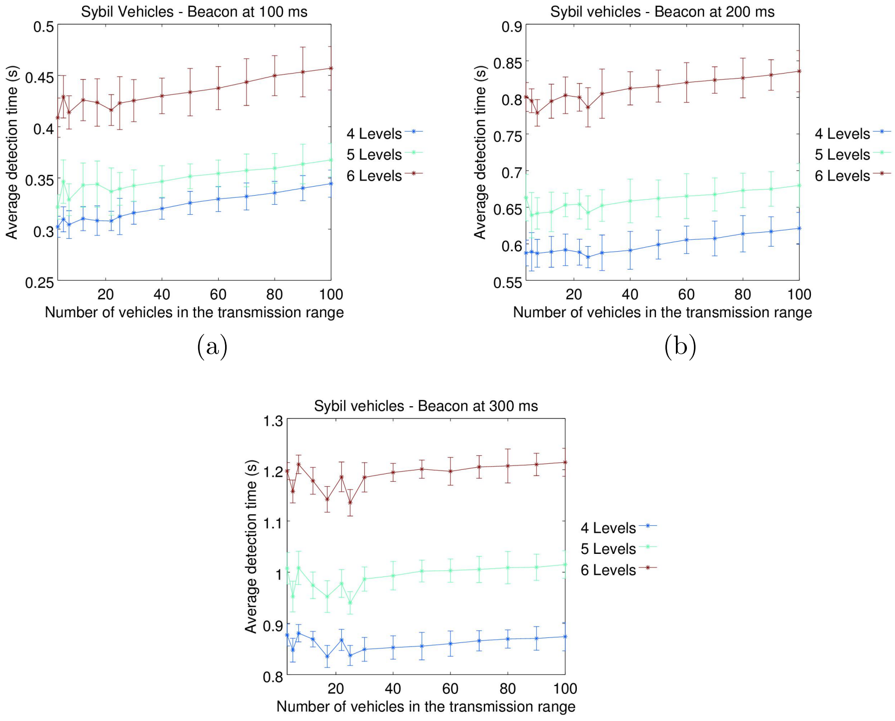
(a)

(b)

(c)

Figure 8: The average time to detect two legitimate vehicles in the transmission range. The vehicles belong to the same $m - 1$ anonymity sets.

Figures 8a, 8b and 8c depict the average time to detect two legitimate vehicles that belong to the same anonymity set at Levels 3, 4, or 5 for 100, 200, and 300 beacon intervals, respectively. In short, the time to detect was lower than 0.4, 0.7, and 1 second, respectively. Since the contention is expected to be higher as the number of vehicles increases, the impact on the results was quite minimal. Moreover, the $ASAP$-V has low impact on the V2V communication standard mainly due to two reasons: first, the vehicles at the same anonymity sets update to the last anonymity set level fast; and the probability that two or more vehicles, in the same transmission range, will choose the same $m-1$ anonymity set digital certificates $(CERT^{t}_{AS_v})$ is $\prod_{i=1}^{m-1}\frac{1}{k}$, which is very low (e.g. $k = 20$). Therefore, the proposed approach provides a stable average of detection time as the number of nodes increases.



(a)

(b)

(c)

Figure 9: The average time to detect a *sybil* vehicle when another legitimate vehicle in the transmission range belongs to the same $m-1$ anonymity sets.

Figures 9a, 9b and 9c depict the average time to detect a *sybil* vehicle with three different identities. Similarly, the wireless contention had minimum impact as the number of vehicle increases. On the other hand, the results exceeded 1 second for 300 *ms* beacon interval at Levels 5 and 6. This happens since the neighbor vehicles must evaluate Equation 4 in order to wait for the next anonymity set digital certificates before deducing a *sybil* attack.

Finally, the main parameters that may affect the *sybil* detection are the beacon time intervals and the number of anonymity sets (per level) in which two or more vehicles together belong to. The results are considered acceptable since the messages from a *sybil* vehicle are dropped (and kept for future purposes) by neighboring vehicles during the attack, and the average detection time is faster than other approaches (as discussed on next section). The proposed approach is totally resilient to *false-negative* and *false-positive* results because any given vehicle may not send messages that describe the same event with different anonymity set digital certificates.

4. Related Work

To the best of our knowledge, Lin et al [37] proposed one of the most efficient group-based authentication for VANETs. In such approach, each vehicle signs the whole message with its group signing key, and may verify the sender's message authenticity by using the group public key. However, the verification process based on group signatures are slower than traditional asymmetric key pairs, which reduces the message verification rate. For instance, it takes 8.5 *ms* on average to verify any received message. Therefore, a vehicle may only verify 125 messages/s, which is very low in high traffic jam. In addition, the verification process does not consider the revocation lists, which is also time-consuming and is proportional to the number of revoked vehicles. The *ASAP*-V protocol uses traditional public/private key pairs for message verification, which increases the verification rate.

Still within the context of group signature, Wu et al. [38] propose an efficient *sybil*-proof threshold authentication for VANETs. A message is viewed as trustworthy only after it has been endorsed by at least t vehicles, where t is a threshold. Since the approach requires a subset of other vehicles for message verification, it may suffer from message loss and delay. On the other hand, *ASAP*-V will have any message delay or loss if the number of vehicles in the communication range is above 250, which will not be feasible due to communication channel overhead. Therefore, vehicles will decrease its signal strength in order to reduce channel errors due to signal collisions and overheads.

With respect to the *sybil* attack detections, Zhou et al. [30, 33] propose a privacy-preserving *sybil* attack detection protocol called P^2DAP. To detect a *sybil* attack, the approach needs the RSU and the CA. The drawback of such approach is that a *sybil* attacker will not be detected if there are no RSUs around. Hence, the approach is highly dependent on the RSU deployment methods and its availability (e.g.: DoS attack may compromise V2V communications). Moreover, experimental results show that a *sybil* detection may achieve 20 seconds due to high overhead imposed on RSUs, which is a high average time for real world on-road services.

Another strategy for detecting *sybil* attacks is based on a timestamp series approach [39, 40]. The approach explores the relationship between time and space, where two or more vehicles will not pass nearby the same RSU and send requests to it at the same time. This approach may compromise users' privacy since it requires vehicle authentication at each RSU, which allows third-party entities to assemble a vehicle routing profile. In addition, the scheme cannot be applied directly to an urban environment with a very complex roadway infrastructure, many signals and intersections, as deeply discussed in [40]. Our approach does not depend on the roadway infrastructure.

Finally, a *sybil* detection approach may use data from neighboring vehicles to filter malicious vehicles [41, 42]. In Grover's et al. approach [43], every vehicle builds a neighboring table (which contains vehicles' identities) with different time interval. After this process, each vehicle shares its neighboring table with other vehicles. If every vehicle has the same neighbors' identities for different time intervals, then these identities may belong to the same (*sybil*) vehicle. Nonetheless, a *sybil* vehicle may never be detected (*false-negative* results) if it changes its identities between consecutive time intervals. As previously discussed, our approach does not provide *false-negative* detections. To sum up, if different vehicles stay together for a long period of time, then it results in *false-positive sybil* detections.

We compare our proposed privacy-preserving *sybil* detection protocol to other similar approaches on Table 4. The main advantage of our scheme is its resilience to both *false-negative* and *false-positive* detections without a centralized infrastructure during detection time, which imposes less overhead on RSUs. In addition, it decreases the average time to detect *sybil* attacks. The dependency on a centralized infrastructure may also compromise VANET services if a more sophisticated network attack also makes such infrastructure unavailable.

Table 4: Comparison to other approaches.

	False-Positive False-Negative Resilient	Non-Repudiation	Beacon or Event	Infrastructure-Dependent	Roadway Infrastructure-Dependent
Our	**Both**	Yes	Both	**No**	No
[30]	Negative	Yes	Both	Yes	No
[39]	None	*	Both	Yes	Yes
[44]	None	*	Both	Yes	Yes
[40]	None	No	Both	Yes	Yes
[43]	None	*	Beacon	No	No
[42]	None	*	Beacon	No	No

5. Conclusion and Future Work

In this paper we have presented a privacy-preserving authentication and *sybil* detection protocol for vehicular ad hoc networks called *ASAP*-V. In order to provide users' privacy, the protocol provides a multilevel anonymity set architecture, with group signature and pseudonyms. The experimental results show its secure and efficiency. Moreover, *ASAP*-V is also resilient to *false-negative* and *false-positive* detections without the support of centralized infrastructures during *sybil* attack detection time. As future work, in order to avoid the RSU from forwarding each single prosecution message to the CA, the RSU must first evaluate if the messages belong to the same *sybil* detection process. This will avoid a CA from measuring redundant prosecution messages. Both V2I communications are still being evaluated.

The challenge in detecting *sybil* attacks in VANETs resides in the potential threats to users' privacy, since during *sybil* detection time, multiple identities must be linked to a one single, yet non-malicious, entity. On the other hand, when a CA may not be always available in VANETs, a potential solution for detecting *sybil* attacks must consider only the vehicles in the region of attack, where each vehicle must share control data in order to detect the attack.

The security aspects are one of the biggest forthcoming challenges for actually deploying the concepts of VANET. The reliability of the whole system may not be compromised due to the high impact it has on people's lives. In this context, among other security-related concepts, authentication, non-repudiation, user's privacy control, and *sybil* attack detections play key roles in vehicular environments and, therefore, they have gained a special attention from the research community.

References

[1] S. Al-Sultan, et al., A comprehensive survey on vehicular ad hoc network, Journal of Net. and Comp. Applications 37 (2014) 380–392.

[2] I. S. Association, et al., 802.11 ieee standard for information technology-local and metropolitan area networks-specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments.

[3] S. Gillani, et al., A survey on security in vehicular ad hoc networks, in: Commun. Technologies for Vehicles, Springer, 2013, pp. 59–74.

[4] J. Freudiger, et al., Mix-zones for location privacy in vehicular networks, in: Proc. of the first international workshop on wireless networking for intelligent transportation systems, 2007.

[5] J. J. Haas, et al., The impact of key assignment on vanet privacy, Security and Commun. Networks 3 (2-3) (2010) 233–249.

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

[7] A. Wasef, et al., Complementing public key infrastructure to secure vehicular ad hoc networks, Wireless Commun. 17 (5) (2010) 22–28.

[8] A. Bradai, H. Afifi, A framework using ibc achieving non-repudiation and privacy in vehicular network, in: Conf. on Network and Information Systems Security, IEEE, 2011, pp. 1–6.

[9] G. Calandriello, et al., Efficient and robust pseudonymous authentication in vanet, in: Proc. of the fourth ACM international workshop on Vehicular Ad Hoc Networks, ACM, 2007, pp. 19–28.

[10] Y. Sun, et al., An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications, Vehicular Technology, IEEE Trans. on 59 (7) (2010) 3589–3603.

[11] A. R. Beresford, F. Stajano, Location privacy in pervasive computing, Pervasive Computing, IEEE 2 (1) (2003) 46–55.

[12] R. Lu, et al., Pseudonym changing at social spots: An effective strategy for location privacy in vanets, Vehicular Technology, IEEE Trans. on 61 (1) (2012) 86–96.

[13] L. Buttyán, et al., On the effectiveness of changing pseudonyms to provide location privacy in vanets, in: Security and Privacy in Ad-hoc and Sensor Networks, Springer, 2007, pp. 129–141.

[14] K. Sampigethaya, et al., Caravan: Providing location privacy for vanet, in: in Embedded Security in Cars, 2005.

[15] L. Buttyán, et al., Slow: A practical pseudonym changing scheme for location privacy in vanets, in: Vehicular Networking Conf., IEEE, 2009, pp. 1–8.

[16] S. Du, et al., Mixzone in motion: Achieving dynamically cooperative location privacy protection in delay-tolerant networks, Vehicular Technology, IEEE Trans. on 62 (9) (2013) 4565–4575.

[17] J. R. Douceur, The sybil attack, in: Revised Papers from the First International Workshop on Peer-to-Peer Systems, Springer-Verlag, London, UK, UK, 2002, pp. 251–260.

[18] T. Leinmüller, E. Schoch, Greedy routing in highway scenarios: The impact of position faking nodes, in: Proc. of Workshop On Intelligent Transportation, 2006.

[19] J. Grover, et al., Performance evaluation and detection of sybil attacks in vehicular ad-hoc networks, Recent Trends in Net. Security and Applications (2010) 473–482.

[20] S. Ramachandran, V. Shanmugan, Impact of sybil and wormhole attacks in location based geographic multicast routing protocol for wireless sensor networks, Journal of Computer Science 7 (7) (2011) 973–979.

[21] Y. Hao, et al., A distributed key management framework with cooperative message authentication in vanets, IEEE J.Sel. A. Commun. 29 (3) (2011) 616–629.

[22] M. Verma, D. Huang, Segcom: secure group communication in vanets, in: 6th IEEE Consumer Commun. and Networking Conf., IEEE, 2009, pp. 1–5.

[23] R. Hussain, et al., Towards privacy aware pseudonymless strategy for avoiding profile generation in vanet, Information Security Applications (2009) 268–280.

[24] T. Wu, et al., A cost-effective strategy for road-side unit placement in vehicular networks, IEEE Trans. on Commun. 60 (8) (2012) 2295–2303.

[25] D. Chaum, E. Van Heyst, Group signatures, in: Advances in Cryptology, Springer, 1991, pp. 257–265.

[26] L. Sweeney, k-anonymity: a model for protecting privacy, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10 (5) (2002) 557–570.

[27] A. Pfitzmann, M. Hansen, Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology, Tech. rep. (Feb. 2008).

[28] P. Syverson, A taxonomy of replay attacks [cryptographic protocols], in: Computer Security Foundations Workshop VII, Proc., IEEE, 1994, pp. 187–191.

[29] S. Schneider, Verifying authentication protocols in csp, Software Engineering, IEEE Trans. on 24 (9) (1998) 741–758.

[30] T. Zhou, et al., P2dap-sybil attacks detection in vehicular ad hoc networks, Selected Areas in Commun., IEEE Journal on 29 (3) (2011) 582–594.

[31] N. Koblitz, Elliptic curve cryptography, Mathematics of Computation 48 (177).

[32] B. Libert, et al., Group signatures with almost-for-free revocation, in: Advances in Cryptology, Springer, 2012, pp. 571–589.

[33] T. Zhou, et al., Privacy-preserving detection of sybil attacks in vehicular ad hoc networks, in: Mobile and Ubiquitous Systems: Networking & Services. Fourth Annual International Conference on, IEEE, 2007, pp. 1–8.

[34] S. Schneider, Security properties and csp, in: Security and Privacy, IEEE Symposium on, IEEE, 1996, pp. 174–187.

[35] C. Dıaz, et al., Information theory and anonymity, in: Proc. of the 23rd Symposium on Information Theory in the Benelux, 2002, pp. 179–186.

[36] C. Sommer, et al., Traffic information systems: efficient message dissemination via adaptive beaconing, Commun. Magazine, IEEE 49 (5) (2011) 173–179.

[37] X. Lin, et al., Gsis: a secure and privacy-preserving protocol for vehicular communications, Vehicular Technology, IEEE Trans. on 56 (6) (2007) 3442–3456.

[38] Q. Wu, et al., Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications, Vehicular Technology, IEEE Trans. on 59 (2) (2010) 559–573.

[39] S. Chang, et al., Footprint: Detecting sybil attacks in urban vehicular networks, Parallel and Distributed Systems, IEEE Trans. on 23 (6) (2012) 1103–1114.

[40] S. Park, et al., Defense against sybil attack in vehicular ad hoc network based on roadside unit support, in: Military Commun. Conf., IEEE, IEEE, 2009, pp. 1–7.

[41] C. Piro, et al., Detecting the sybil attack in mobile ad hoc networks, in: Securecomm and Workshops, 2006, IEEE, 2006, pp. 1–11.

[42] Y. Hao, et al., Cooperative sybil attack detection for position based applications in privacy preserved vanets, in: Global Telecommun. Conf., IEEE, IEEE, 2011, pp. 1–5.

[43] J. Grover, et al., A sybil attack detection approach using neighboring vehicles in vanet, in: Proc. of the 4th international Conf. on Security of information and networks, ACM, 2011, pp. 151–158.

[44] C. Chen, et al., A robust detection of the sybil attack in urban vanets, in: Distributed Computing Systems Workshops. 29th IEEE International Conf. on, IEEE, 2009, pp. 270–276.