WEVAN – A mechanism for evidence creation and verification in VANETs

Jose Maria de Fuentes[1], L. Gonzalez-Manzano, A. I. Gonzalez-Tablas, J. Blasco
Computer Science and Engineering Department
University Carlos III of Madrid
Avda. Universidad, 30. E-29811 Leganes (Spain)
{jfuentes, lgmanzan, aigonzal, jbalis}@inf.uc3m.es

Abstract

There are traffic situations (e.g. incorrect speeding tickets) in which a given vehicle's driving behavior at some point in time has to be proved to a third party. Vehicle-mounted sensorial devices are not suitable for this matter since they can be maliciously manipulated. However, surrounding vehicles may give their vision on another one's behavior. Furthermore, these data may be shared with the affected vehicle through VANETs. In this paper, a VANET-enabled data exchange mechanism called WEVAN is presented. The goal of this mechanism is to build and verify evidences based on surrounding vehicles (called *witnesses*) testimonies. Due to the short-range nature of VANETs, the connectivity to witnesses may be reduced with time – the later their testimonies are requested, the lower the amount of witnesses may be. Simulation results show that if testimonies are ordered 5 seconds later, an average of 38 testimonies may be collected in highway scenarios. Other intervals and road settings are studied as well.

Keywords: Digital evidence; driving behavior; Vehicular ad-hoc networks (VANET); witness.

1 Introduction

Information technologies are being strongly improved in vehicular environments, leading to a new family of services collectively called Intelligent Transportation Systems (ITS). They are mainly based on sensors currently mounted on vehicles and a new communication network called Vehicular Ad-hoc NETwork (VANET).

This technological context may be the basis for non-safety related vehicular services. In particular, there are situations in which it is necessary for a driver to prove its recent driving behavior. Accidents and traffic offences are good examples. In case of an accident, it is useful to have an accurate behavior description to perform a fair liability attribution. Regarding traffic offences, this description may reveal that a purported illegal action (e.g. speeding, red-light crossing) was not committed. A natural data source for describing this behavior is the vehicle itself. This approach is present in previous contributions such as [1]. Nevertheless, in-vehicle sensors or their connecting buses can be tampered with. Several countermeasures have been proposed for both sensor and bus protection, but they are not widely implemented yet [2].

Approach overview. An alternative data source about a driver behavior is needed. VANETs can be useful for this purpose. As part of the VANET regular operation, vehicles send each other a message called *beacon*. It contains the sender speed and location, among other data [3]. As a vehicle is aware of the recent driving behavior of nearby ones, the former could act as *witness* of the latter. For this purpose, vehicles can combine the beacon data with their in-vehicle sensor measurements [4].

Using testimonies from other vehicles allows describing a given driver behavior in a more reliable way. An intuitive assumption is that a driver does not share any interest with others circulating around. Thus, there is no reason for a witness to lie in favour of the affected driver. Moreover, even if subornation could always be performed, it could be countered by imposing legal consequences to false testimonies, as it happens with current ones.

The challenge. Witness vehicles are reachable by the affected vehicle for a short time period, due to the restricted range of vehicular communications and the high mobility of vehicles. Therefore, the proposed

[1] Corresponding author: Jose Maria de Fuentes. E-mail: jfuentes@inf.uc3m.es . Phone: +34916245957
Fax: +34916249129

mechanism must allow witnesses to be inquired as soon as possible. Moreover, it has to deal with the unreliability of the wireless vehicular communication network.

Our contribution. In this work, a Witness-based Evidence generation protocol for Vehicular Ad-hoc Networks (WEVAN) is proposed. The evidence verification process is also described. It is applied to the scenario of defence against an offence notification, assuming that this notification is directly delivered to the vehicle. Its suitability for vehicular networks and computational devices is analysed. Moreover, the amount of available testimonies per evidence is also assessed through simulations. A previous result of this research line was already presented in [5]. However, the protocol presented herein outperforms the previous proposal in three main aspects: (1) it is based on an underlying enforcement process model, (2) it counters uncooperative behaviors by vehicles, and (3) it is evaluated taking into account realistic vehicular devices and networks, as well as different road traffic scenarios.

Scope. Evidences and testimonies will be both referred to a single behavior-describing variable which may be detected or estimated by a nearby vehicle. Particularly, this work will only consider *position* and *speed*. On the other hand, only rational attackers will be considered. Irrational attackers like jammers (i.e. entities that smurf the network with bogus data) are considered to be already countermeasured, for example by local eviction of misbehaving nodes [6].

Paper outline. The model and architecture are presented in Sections 2 and 3, respectively. The protocol is described in Section 4 and evaluated in Section 5. The related work is described in Section 6. Section 7 concludes the paper.

2 Model

This Section describes the considered model for this contribution. Section 2.1 introduces the participant entities and their identified interaction models. Section 2.2 describes the requirements that the proposal has to fulfil. Finally, Section 2.3 describes the working assumptions.

2.1 Participant entities

The creation of evidences may be applied to the enforcement process. Therefore, the participant entities are a subset of those that make up the enforcement process model. For this contribution, the model proposed by de Fuentes *et al.* will be considered [7]. Particularly, five entities are related to this process, namely the *Offending ITS-enabled vehicle*, the *CounterEvidence Analyser* (CEA), the *Designated-as-offender Contact Point* (DCP), the *Surrounding ITS-enabled vehicle* and the *Data Requester* (DR). Thus, the Offending ITS-enabled vehicle will perform two actions. First, it will send to the Authority (specifically, CEA) its claim on its past behavior. For this purpose, DCP will be used as the intermediary of this communication. In this way, the Offending vehicle does not have to care the specific CEA that has to be contacted, which may depend on the internal organization of the enforcement infrastructure.

The second action is to obtain the information (called *testimonies*) from surrounding vehicles. From the logical point of view, this interaction involves that at a certain point in time it is necessary to extract some information from these vehicles. These data are sent to CEA through DR.

2.1.1 Identified interaction models

There are three ways in which the identified entities may interact to perform the evidence generation process. They will be referred to as the *centralized* approach, the *decentralized* and the *combined* ones. In the first case, the Offending ITS-enabled vehicle relies on DR (which is seen as a single, central entity) to ask the Surrounding vehicles on behalf of the Offending one, to retrieve their information and to create the evidence. In the decentralized approach, it is the Offending ITS-enabled vehicle which asks for the Surrounding vehicle data, retrieves it and builds the evidence. In the combined one, it is the Offending vehicle who requests for testimonies, while DR collects them. The Offending vehicle then sends a summary of its expectations on the future evidence. Based on this summary, CEA (which receives the summary along with the data retrieved by DR) compiles the evidence and proceeds with its verification and evaluation.

From the enforcement model point of view, the three identified interaction models are suitable. Even if any of them matches exactly the data flows of the enforcement model presented in [7], all of them respect the definition of the entities and the logical division of their responsibilities. Thus, in all cases DR is focused on

retrieving data from witnesses, DCP is the entity that receives data from the offence-related stakeholder (the Offending vehicle, in this case) and CEA focuses on analysing the evidence, obtaining data from DR when necessary to perform this evaluation.

2.2 Requirements

There are four requirements that must be achieved by the devised solution. Each one is introduced below.

Correctness. The protocol must enable the creation of a behavior-describing evidence *ev* for the Offending vehicle. Such an evidence must contain one or more testimonies from surrounding vehicles. The protocol must enable CEA to validate the aforementioned evidence. For this purpose, the following four conditions must hold:

> ***Condition 1 (supported evidence).*** *ev* has to contain at least one testimony referred to the offence identifier *offence-id* to which the evidence is related.
>
> ***Condition 2 (value consistency).*** Let *testimValue* be the perception of the behavior-related variable included in a testimony appearing in *ev*. Given that *claimedValue* is the Offending vehicle's claim on that variable, at least one testimony in *ev* must deviate from testimValue less than a predefined parameter *confidThreshold*.
>
> ***Condition 3 (time consistency).*** All testimonies contained in *ev* must contain a time mark t_{test} such that $t_{off} < t_{test} < t_{evid}$, being t_{off} the time of the offence and t_{evid} the time when the evidence is issued.
>
> ***Condition 4 (identity consistency).*** Every testimony appearing in *ev* must be signed by a different entity. Moreover, there must not be a testimony created by the Offending vehicle.

Confidentiality. Testimonies and evidences should only be available for CEA, apart from their issuers.

Authentic requests for testimonies. Only authentic requests should be processed by receiving vehicles. A request is said to be authentic if, on the one hand, it is related to a genuine previous offence notification and, on the other, it has not been modified since it was created.

Authentic testimonies. False testimonies should be identified as such by the receiving entity. A testimony is considered to be false if the contained data is not reasonable (e.g. a vehicle may not be driven at 600 kph), if its sender is not properly identified or if it is not possible to attest that it was present (i.e. near the Offending vehicle) at the time of the facts.

2.3 Working assumptions

The solution devised herein is suitable to work in scenarios where the following five conditions hold. First, a secure boot-up process exists through which the appropriate Authority (e.g. a Certification Authority, CA) installs all cryptography-related materials into vehicles, particularly in a secure cryptographic device called Hardware Security Module (HSM) [8]. This material contains the public-private keypairs along with the corresponding pseudonym-based public key certificates (referred to as $Cert_{E(t)}$).

The second assumption is that a Secure Location Verification service is being executed by vehicles, to determine which ones are actually in its vicinity [9].

The third assumption is that vehicles extract and store the behavior-related data from the received beacons. Furthermore, it is assumed that vehicles store during a period *p* the information provided by in-vehicle sensors and the full set of received beacons. This period *p* is assumed to be greater or equal than the interval between the offence and its notification. The storage required to fulfil this assumption is analysed in Section 5.1.

The last two assumptions are related to beacons. On the one hand, it is assumed that all beacons are signed by their issuers. On the other hand, once a vehicle receives a beacon from another one, the latter will also be receiving the beacons from the former. In this way, once vehicle A receives a beacon from vehicle B, both are sure that the other one may act as its witness.

3 Architecture

This Section introduces the architecture derived from the model presented in Section 2. The considered architecture is depicted in Figure 1, which shows the entities from the model described in Section 2 (marked with a broken line) and their technical realization. The participant entities are grouped according to the network environment they belong to, either the background or the vehicular one. Section 3.1 describes the background environment, whereas Section 3.2 introduces the vehicular one. Section 3.3 describes how both environments

are connected. The threat model is presented in Section 3.4. Finally, the selection of the interaction model among those presented in Section 2.1 is described in Section 3.5.

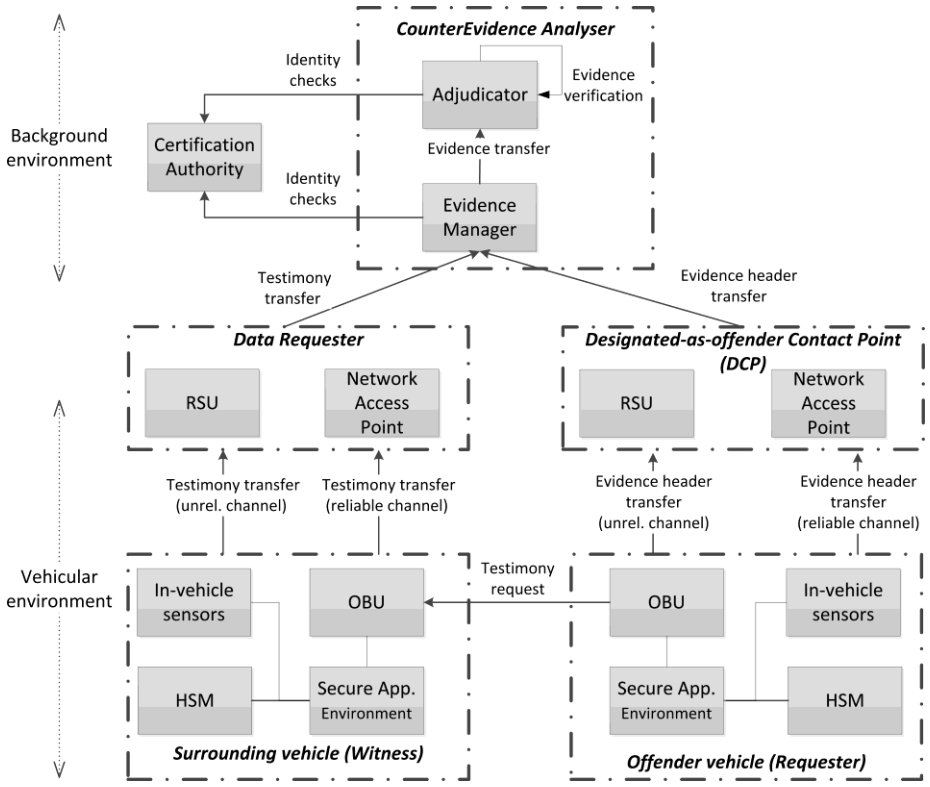

Figure 1. System architecture

3.1 Background environment

There are three entities in the background environment, namely the *Certification Authority* (CA), the *Adjudicator* (Adj) and the *Evidence Manager* (EM). CA manages (i.e. issues, transfers and revokes) pseudonymous public key certificates ($Cert_{E(t)}$) that bind a cryptographic key with a pseudonym assigned to the vehicle. Thus, CA is the top entity within a Public Key Infrastructure (PKI), and it is the only entity that is able to relate a pseudonym with a real identity. Adj decides about the imposed fine taking into account the evidence proposed by the offender. This evidence is previously managed by EM.

Concerning the evidence verification and adjudication conducted by Adj, both tasks are properly within the scope of CEA (recall Section 2). At the light of their respective descriptions, both Adj and EM collectively form the task developed by CEA in the model. All entities that form the background environment are static, and so they are placed in traditional computation nodes.

3.2 Vehicular environment

In the vehicular environment, vehicles are connected through a Vehicular Ad-hoc NETwork (VANET). For this purpose, they contain an *On-Board Unit* (OBU) which provides several communication interfaces (e.g. IEEE 802.11p, GPRS, etc.), as proposed in the CVIS project [10]. Given that each vehicle is identified by means of temporary pseudonyms, each OBU will be able to receive packets that are sent to one of its previous but recent pseudonyms to avoid routing problems [11].

Apart from the OBU and the HSM (recall Section 2.3), there are two additional in-vehicle devices, which are organized considering the OVERSEE architecture [12]. In this way, there exists a *Secure Application Environment* (SAE) where applications reside. From the SAE viewpoint, the proposed protocol is an application itself. These applications are transferred to the vehicle by means of a secure dissemination strategy, such as [13].

Each vehicle is also equipped with *sensors*, which give information related to the vehicle current status (position, speed) and to its surroundings.

3.3 Connection between environments

The connection between both the background and the vehicular environment is performed through *Road Side Units* (RSUs), which are static nodes placed aside the roads that participate in the VANET. Thus, the RSU task involves receiving some data from the Offending vehicle (as it is done, in the enforcement model, by the Designated-as-offender Contact Point) and from witness vehicles (as it is done in the enforcement model by the Data Requester). All RSUs are connected to EM. Apart from this connection, there exists a resilient channel between the vehicular entities and EM, which ensures that packets eventually arrive. One typical environment for such a channel is a location-restricted connection like an at-home network. This channel is built periodically, for example at a daily basis. The Network Access Point is the entity that enables the communication between OBUs and EM.

3.4 Threat model

Threats on correctness. There are two threats on this issue. First, every message sent through an unreliable network (as it is the case of the vehicular one) may be altered or lost. Second, the aforementioned messages may be never created, even if mandated by the protocol. One example of this is that OBUs may be compromised in such a way that they refuse to participate in the protocol.

Threats on confidentiality. The eavesdropping threat may happen in the vehicular environment (as usual in shared medium networks such as VANETs) as well as in the background network (due to its unreliability).

Threats on authentic requests for testimonies. A rational attacker may ask for testimonies referred to other vehicle as a means of obtaining some information about the victim's past behavior.

Threats on authentic testimonies. A false testimony is not beneficial for a well-behaving vehicle, as it may lead to legal consequences. However, a rational attacker may be interested in creating testimonies without being in the surroundings of the offender, if a reward is given by the offender. Apart from this threat, a malfunctioning sensor may originate inaccurate testimonies.

3.5 Selection of the interaction model

Taking into account the interaction models identified in Section 2.1, in this Section they are comparatively analysed. Furthermore, the most suitable one is selected. Without entering into the details of the exchanged messages for each particular setting, some conclusions may be reached from the general features of each approach. These features are the *system scalability*, its *auditability* and its *effectiveness* (see Table 1).

	Centralized	Decentralized	Combined (selected)
System scalability	-	++	+
System auditability	++	-	++
System effectiveness	+	- -	+

Table 1. Analysis of approaches for the testimony collection and evidence generation. The rating for each feature ranges from ++ (totally fulfilled) to -- (poorly fulfilled)

Regarding the system scalability, it must be noted that the decentralized choice is more scalable than the remaining approaches, as the workload from EM is reduced. Even considering that EM's computational power greatly overcomes that offered by vehicles, the amount of offences that may be detected (at a nation-wide scale) at the same time suggests that EM may become a bottleneck. However, the feasibility of this approach should be analysed, as several real-time ITS services will be running at the same time over the (constrained) vehicular computational device. On the other hand, the combined approach seems to appropriately balance the requirements from both parts. However, experimental evaluations with real vehicular hardware will be interesting to assess this issue.

The system auditability measures whether it is possible to reliably determine the operations that have been performed to achieve a result. In this context this is a critical feature, as there could be consequences after the execution of this mechanism, e.g. call for maintenance due to the lack of response by a witness. In this regard, the decentralized approach is less suitable than the remaining options. As all the inter-vehicle communications are performed over an unreliable channel, it would be impossible to determine whether the absence of a testimony (needed by a certain Offending vehicle) is due to the loss of the request, of the testimony or the uncooperative behaviour from the witness [14]. However, a lazy Offending vehicle could claim that it sent a request but did not receive a testimony, thus forcing EM to collect it. In this way, the Offending vehicle could

save resources, but it could never be determined whether its claim was trustworthy. The centralized variant is similar to the combined approach in this issue, as in both cases EM (which is trusted) takes part in the process, using the resilient channel.

The system effectiveness measures the capacity of the system to create evidences based on testimonies. The decentralized approach is again inappropriate for this context. To understand this issue, it is important to note that a testimony that is not beneficial for the Offending vehicle could cause it to take reprisals against the witness. Moreover, it is reasonable to assume that if the Offending vehicle would know the value of the testimonies, it will remove the ones that are not favourable to it to avoid wasting resources by creating evidences that are against its interests. In this way, a witness holding a non-profitable value for the Offending vehicle would never answer in the decentralized choice. Therefore, this approach would prevent these testimonies to be managed. On the contrary, the system effectiveness offered by the centralized version and the combined one is similar, as both enable a private communication between the EM and every witness. Thus, these unfavourable testimonies could be freely sent to EM. They could be used to enable the Authority to complement its proof against the offender. For this reason, we consider that the effectiveness of the combined approach (and, similarly, of the centralized version) is better than the decentralized one.

At the light of these considerations, the combined approach is the most suitable one as it addresses successfully all the analysed features. For this reason, it will be selected for the development of this contribution.

4 Protocol specification

The proposed protocol is composed by three parts, namely the testimony collection, the evidence generation and the evidence verification. Furthermore, there are two exceptional situations caused by data loss that must be properly handled, one concerning the offending vehicle and the other related to witnesses. The following subsections describe, first, the data structures (Section 4.1) and operations at stake (Section 4.2) and, afterwards, each of the aforementioned process parts and the exceptional processes. Figure 2 depicts graphically the whole process. As the Offending vehicle will send the request for testimonies to witnesses, in the following it will be referred to as the Requester (R), whereas Witnesses will be referred to as W_i.

4.1 Data structures

Apart from the aforementioned beacons, there are four data structures in this work, namely *testimony*, *request*, *evidence header* and *evidence*. Table 2 summarizes their contents and size. A testimony $Testim_{E1}$ allows one vehicle E_1 to describe a behavior-related variable of another vehicle E_2 at a time t_{test}. In order to retrieve a testimony, R (i.e. the Offending vehicle) sends a request $Req_{R(Treq)}$. In order to prevent a third party to impersonate R, the request is divided into two parts, each one signed under a different identity, $R(t_{off})$ and $R(t_{req})$, which are the sender's pseudonyms when the offence is committed and the request is created, respectively. Finally, the most complex data structure is the evidence ($Evid_{EM}$). It is formed by an evidence header, a set of supporting testimonies and the time t_{evid}. The header $EvidHdr_{R(Tevid)}$ contains: (1) the identity of the Requester in the moment of the evidence ($R(t_{evid})$), (2) its claim on its past behavior (called *claimedValue*), (3) the identification of the offence *offence-id*, (4) the beacons that show that witnesses were in R's surroundings at t_{off} (plus their corresponding public key certificates), and (5) the time marks t_{off} and t_{evHdr}.

Data structure	Contents	Size (bytes)
Beacon$_{X(t)}$	$S_{R(t)}$(R(t), speed, position, t)	**4**+**2**+**10**+**2**+*56* = 74
Req$_{R(Treq)}$	(*part1*, *part2*) where part1 = $S_{R(Treq)}$(R(t_{off}), t_{off}), part2= $S_{R(Toff)}$(offence-id, type), being type = *position* or *speed*	Part1 = **4**+**2**+*56* Part2 = 4+1+*56* Total = 123
Testim$_{Wi(Ttest)}$	$S_{Wi(Ttest)}$ (W_i(t_{test}), offence-id, R(t_{off}), [position or speed], t_{test})	**4**+4+**4**+[**10** or **2**]+**2**+*56* = = 80 (position testimony) or 72 (speed testimony)
EvidHdr$_{R(Tevid)}$	$S_{R(Tevid)}$ (R(t_{evid}), offence-id, claimedValue, t_{off}, Beacon$_{W1(Toff)}$, Cert$_{W1(Toff)}$,…, Beacon$_{Wn(Toff)}$, Cert$_{Wn(Toff)}$, t_{evHdr})	**4**+4+[**10** or **2**]+**2** + + *nw* · (74 + *125*) + *56*
Evid$_{EM}$	S_{EM}(EvidHdr$_{R(Tevid)}$, SupportingTestim, t_{evid}) where SupportingTestim ={Testim$_{W1(Ttest)}$, ... ,Testim$_{Wn(Ttest)}$}	Size(EvidHdr) + 2 + + *nw* · Size (Testim) + *56*

Table 2. Summary of data structures: Contents and size. Size values in bold are taken from SAE J2735 standard [3], whereas those in italics are from IEEE 1609.2 [15]. Key*: nw* = number of witnesses

4.2 Cryptographic operations and auxiliary functions

In the context of this process, public key cryptography is considered. Particularly, to protect the confidentiality of messages, public key encryption (noted as $E_{X(t)}(M)$) and its corresponding decryption ($E^{-1}{}_{X(t)}(M)$) will be applied. On the other hand, to ensure their integrity and data origin authentication, digital signatures ($S_{X(t)}(M)$) and their verifications ($S^{-1}{}_{X(t)}(M)$) are in use.

Apart from cryptographic operations, entities are able to execute seven operations. Vehicles may look for behavior-related data from other vehicles through the *lookupBehRecord* function. They may also retrieve the public key certificate of other entity using the *lookupCert* function. To find suitable witnesses for a given vehicle, it may execute the *findNeighbours* operation. This operation relies on the Secure Location Verification service. For each one, it returns the beacon that shows that it was near that vehicle, along with the public key certificate for verifying it. Once a testimony is created, the receiving entity can store/retrieve it using the *storeTestimony/retrieveTestimony* operations. In order to check if a claim is supported by a given testimony, *contains* enables finding whether a given value is within an interval (for speeds) or region (for positions).

As opposed to the previous operations, there is an operation (*checkPseudonymsEntity*) that is only available for the CA. This operation enables determining whether two different pseudonyms belong to the same entity.

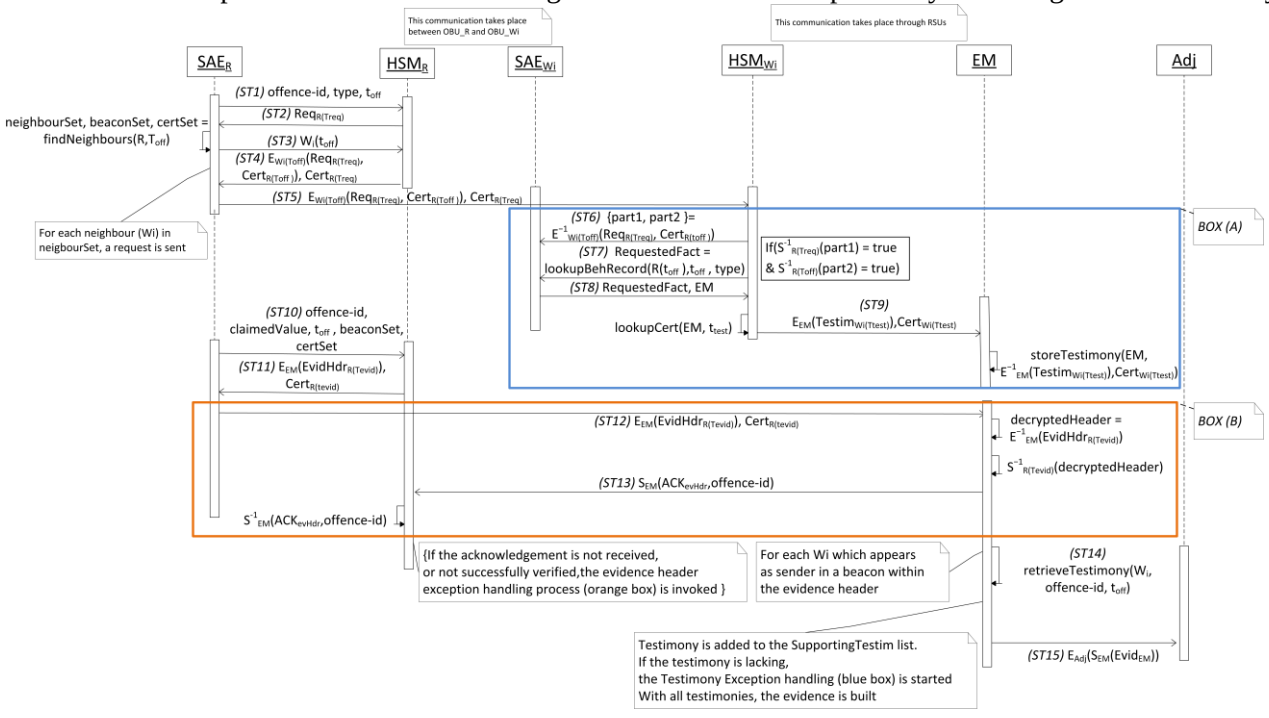

Figure 2. Proposed evidence generation protocol

4.3 Testimony collection

Once a vehicle has received a fine notification, its SAE determines whether it is suitable to ask for evidences to challenge the fine. In such a case, SAE extracts the offence identifier and the time of the offence from the offence notification to build the request. Furthermore, it determines which behavior-related variable should be witnessed, and sends all these data to the HSM to build the request (steps ST1-ST2 in Figure 2). In order to determine the vehicles that are candidate to be witnesses, the function *findNeighbours* is used to establish which vehicles were around in the moment of the offence (t_{off}). If there was no single vehicle able to participate as witness, it is not possible to create any evidence so the process is stopped. Otherwise, for each of these witness vehicles, the request is sent (steps ST3-ST5). Apart from being signed, the request is encrypted as it contains a private statement: the Requester, which is currently using pseudonym $R(t_{req})$, was using pseudonym $R(t_{off})$ at t_{off}. For the same reason, the public key certificate $Cert_{R(Toff)}$ is also encrypted. It must be noted that the Requester is able to encrypt data for witnesses as it stores, for some time interval, their public key certificates (recall Section 2.3).

Once a Witness W_i receives and decrypts the testimony request (ST6), it verifies the signature. The verification includes checking the status of Requester's certificates, which is important to avoid creating testimonies for a vehicle which is in an irregular situation. If the verification is correct, it searches within its memory any information relevant to R in t_{off} (ST7). If it exists, a testimony is prepared and sent encrypted to EM (ST8-ST9). The encryption is necessary to avoid third parties to be aware of the witnessed value. Significantly, R should not realize of this value to avoid retaliation against W_i in case that the testimony is against R's interests. However, the public key certificate necessary to verify the signature is not encrypted, as it only contains public information. All these data are stored by EM and will be used to create the evidence afterwards.

4.4 Evidence generation

When R (specifically, its SAE) estimates that all witnesses have had enough time to send their testimonies, it starts the creation of the evidence header. For this purpose, it sends to HSM_R the offence identifier, the time of the offence, the set of designated witnesses (including the beacons and the corresponding public key certificates) and its estimation (claimedValue) on the behavior-describing variable (steps ST10-ST11). This header is then sent to EM through one RSU, encrypted to prevent other vehicles to learn the status data of witnesses (ST12). Again, the public key certificate is not encrypted as it is not confidential. EM then decrypts and verifies the evidence header signature. If it is not correctly signed, the evidence header is discarded. Otherwise, EM acknowledges it (ST13). It must be noted that if the acknowledgement is not received/verified within a reasonable time interval (considering EM processing speed and transmission delays), R starts the corresponding Exception Handling procedure (see Section 4.6).

If the evidence header was correctly received and verified, EM compiles the evidence incorporating the corresponding testimonies based on the witness list provided in the evidence header (ST14). If any of them has not been received, the Testimony Exception Handling procedure is marked to be started once the witness connects using the reliable channel (see Section 4.6). Once all the available testimonies have been collected, the process of generating evidence is finished. EM transfers it to Adj, which will verify it before the adjudication process (ST15).

4.5 Evidence verification

The evidence verification process (not shown in Figure 2) is executed by Adj and starts by verifying the signatures on the evidence and on each of the beacons contained in the evidence header. It should be noted that the signature on the evidence header was already verified by EM during the evidence generation. If any of these verifications fail, the whole evidence is discarded, as it is conceptually invalid. This also applies in case that it is one beacon which is not successfully verified. It should be noted that the vehicle should have already verified this beacon, so an invalid signature indicates that the vehicular devices are not operating regularly.

In case that all the aforementioned verifications are successful, the checks on the content may start. First, it is evaluated if the verification is performed in a moment later than that in which the evidence was created. In such a case, each of the testimonies is analysed. If its signature is verified, then several checks are applied over the evidence contents: coherence of times, of identities and of the behavior-describing values. Thus, the testimony must be created at a reasonable time (i.e. after the fine notification but before the evidence time). It should be noted that there is no need to verify if the testimony is issued by one of the witnesses designated by R, as it is ensured by the process followed by EM to create the evidence. However, all participants (i.e. R and all

W_is) must be different among them. To this regard, Adj uses the *checkPseudonymsEntity* operation to ensure that the different pseudonyms are not related to the same entity. In case that an identity fraud is detected, the verification process is aborted and the CA is contacted to reveal the identity of the involved entity. Similarly, Adj takes the same decision if R is not related to the offence identified by offence−id.

If all the previous inspections are successful, the operation *contains* is used to determine whether the witnessed value supports R's claim, i.e. belongs to a confidence interval around claimedValue, using a predefined confidence parameter *confidThreshold*. The process is repeated for all beacons contained in the evidence header. At the end of this process, Adj determines (1) if the evidence is valid and (2) the amount of testimonies that support the offending vehicle's claim.

4.6 Exception handling

There are two exceptional situations, caused by the data loss in the communication channel. The first one is the absence of an expected testimony, which may happen if the witness did not receive the request, the testimony itself was lost or even the purported witness did not know the requesting vehicle. The second one is the lack of acknowledgement for the evidence header, which makes the Requester be unaware of the successful starting of the evidence generation by the EM. This may be caused because either the evidence header or its acknowledgement was lost in transmission.

In order to manage these situations, a data exchange will take place. In order to avoid the uncertainty caused by the channel unreliability, these exception handling mechanisms are run over the resilient channel between the vehicular entities and EM (recall Section 3.3).

Particularly, the Testimony Exception handling procedure consists essentially on the repetition of the testimony generation part of the regular process (see box (A) in Figure 2, steps ST6-ST9), which is started by request of EM. For this purpose, EM sends all the required data: which offence is related (offence-id), who was involved ($R(t_{off})$) and when it happened (t_{off}). If there is no response from the vehicle or the testimony signature is not correct the vehicle is called for maintenance to verify the vehicular devices. In the extreme situation in which there is no valid testimony from any of the witnesses, the evidence would not be created due to the lack of supporting data.

On the other hand, in the Evidence Header exception handling it is the vehicle who repeats the evidence header transfer to EM (see box (B) in Figure 2, steps ST12-13). This process only finishes when the acknowledgement is successfully received by R. It should be noted that this process may be run by R simply because the acknowledgement, but not the evidence header itself, was lost. In such a case, EM would have already performed the steps of the Evidence generation algorithm that are beyond the acknowledgement (recall Figure 2). Otherwise it is necessary for EM to proceed with these steps.

5 Evaluation

In this Section, the proposed mechanism is assessed using two ways. First, a performance evaluation is shown in Section 5.1. Second, the fulfilment of the imposed requirements is analysed in Section 5.2.

5.1 Performance evaluation

In this Section, the performance of the proposed approach is evaluated. Due to the unreliability of the vehicular network, the high mobility of vehicles and their limited computational resources, the more challenging environment is the vehicular one. Therefore, this analysis will focus on how the protocol performs in such an environment. Particularly, two indicators will be considered, namely (1) the computational and storage cost for vehicles and the impact in the vehicular network (analysed in the three first subsections of this Section) and (2) the amount of testimonies per evidence that may be achieved in different road scenarios (analysed in the last subsection of this Section).

5.1.1 Vehicular computational and network cost

Prior to estimate costs, it is necessary to define the computational and network available resources. Concerning the computational platform, a commercial vehicular HSM (CycurV2X[2]) is considered. It is estimated that the most costly operations are related to cryptographic calculations. Thus, only these operations will be considered in this analysis.

[2] https://www.escrypt.com/products/cycurv2x/, last accessed in July, 2013.

Process	Data / Cryptographic operations	Vehicular proc. time (ms)	Transmission time (ms)
Testim. Collection (Requester)	Request: 2 SIG + *nw* · ENC	14.31 + *nw* · · 670.51	0.47 · *nw*
Testim. Collection (Witness)	Request: 1 DEC + 2 SIG VERIF Testimony: 1 SIG + 1 ENC	708.1 + 2 · σ_v	0.73
Evid. Generation (Requester)	Evid. Header: 1 SIG + 1 ENC Evid. Header acknow: 1 SIG VERIF	341.59 + σ_v	0.52
Testimony exception handling	Testim. enquiry: 1 DEC +1 SIG VERIF Testimony: 1 SIG + 1 ENC	216.21 + σ_v	0.53
Evidence header exception handling	Evid. Header: 1 SIG + 1 ENC Evid. Header acknow: 1 SIG VERIF	341.59 + σ_v	0.52

Table 3. Vehicular computational and network costs. Key: *ENC/DEC* = Encryption/Decryption ; *SIG/SIG VERIF* = Signature/Sig. verification ; *nw* = number of witnesses ; σ_v = certificate status verification time

According to figures provided by its manufacturer, CycurV2X performs ECIES encryption of 16 bytes in 27.938 milliseconds (21.26 ms. for decryption). ECDSA signatures are performed in 7.156 ms. (27.114 ms. for its verification, plus a time σ_v to verify the public key certificate status). Both ECIES and ECDSA are selected for compliance to the current standard in security of vehicular networks (IEEE 1609.2, [15]). Related to the network resources, a typical inter-vehicle DSRC (Dedicated Short Range Communications) network is considered. This network has a bandwidth of 6 Mbps [16].

Table 3 details the cryptographic operations performed by vehicles in the main steps of the process. The evidence verification is not included in Table 3 as it is an internal process performed by Adj. Note that the provided performance data are referred to 16 bytes, but the data structures have a different size. Thus, it is necessary to extrapolate these values for each size, which depends on the cryptographic algorithm design. As ECIES is based on a stream cipher, it is reasonable to estimate that there will be a linear relationship between the message size and the encryption time. On the other hand, ECDSA is based on SHA-224 or SHA-256 hash functions [15]. As it uses a message block size of 64 bytes, which is greater than the data to sign by the vehicle (recall Table 2), we will consider that the signature time is the same in all cases.

Regarding the transmission costs, there are two relevant factors, namely the *propagation delay* and the *network transmission* one. The propagation delay is assumed to be negligible. To calculate the network transmission delay (see Table 3), message sizes from Table 2 are considered. Thus, for the situation in which one witness is present (i.e. *nw* = 1), Equation 1 shows the time taken for the whole process including both processing and transmission costs. In case that an exception happens, it is necessary to wait for the resilient channel to be available. As it typically means the time to arrive to the physical place where this channel exists, there is no reasonable estimation for this value. For this reason, in Equation 1 only the case with no exceptions is considered.

$$T_{evid-gen} = T_{crypto}(\text{Testim. Collection(Req)}) + T_{trans}(\text{Testim. Collection(Req)}) + T_{crypto}(\text{Testim. Collection(Wit)}) + + T_{send-testimony} + T_{crypto}(\text{Evid. Generation(Req)}) + T_{trans}(\text{Evid. Generation(Req)}) = (14.31 + 1 \cdot 670.51) + 0.47 + +(708.1 + 2 \cdot \sigma_v) + 0.26 + (341.59 + \sigma_v) + 0.52 = 1735.76 + 3\,\sigma_v \text{ ms.} \quad \text{(Eq. 1)}$$

In the previous expression, $T_{trans}(x)$ refers to the transmission time shown in Table 3 for each part of the process, whereas $T_{crypto}(x)$ represents its cryptographic processing (also shown in Table 3). Moreover, time $T_{send-testimony}$ has been calculated by simply isolating the transmission costs of the testimony from T_{trans}(Testim. Collection (Wit)), given that the request transmission had been already taken into account in T_{trans}(Testim. Collection(Req)). According to Equation 1, the process takes around 1.8 seconds (plus 3 times the certificate status verification time) to be completed. In this calculation, the transmission of public key certificates (as shown in Figure 2) is considered. These certificates have 125 bytes, according to their structure and contents defined in IEEE 1609.2 standard [15].

5.1.2. Vehicular storage needs for the witness

In general words, the witness is forced to (1) perform a connection to EM using the resilient channel at a periodic basis (typically, daily) and (2) if necessary, give the pending testimonies using that connection (being called for

maintenance if it is not performed). For this purpose, vehicles have to store the behavior-related data contained in incoming beacons.

In order to estimate the storage needs, it is necessary to determine the amount of incoming beacons. This amount is determined by the density of vehicles that are around a given one in its connectivity range. This density ranges from 40 vehicles / km^2 to 320 vehicles / km^2 [17]. Taking into account that beacons are sent every 100 ms. ([3]) and that DSRC range is 1 km., vehicles may be receiving from 400 to 3200 beacons per second. For each one, a total amount of 18 bytes is necessary for its storage: 2 bytes for the speed value, 10 bytes for the positional (latitude, longitude, elevation) information, 4 bytes for the vehicular identifier and 2 for the time mark [3]. This leads to the amount of storage required for one second. Generalizing this value for a one-hour trip (which seems to be reasonable for an urban environment), the maximum storage required in the worst case (i.e. higher density) is 207,36 Mb. Apart from this information, it is necessary to store the testimonies that have been sent in the whole period, as they may have been lost in the vehicular channel. Considering 100 testimonies in the aforementioned trip, in the worst case (i.e. position testimonies) they require 2400 bytes of storage – the digital signature included in the testimony does not have to be stored. Taking into account the current state-of-the-art in storage technologies, the required capacity seems to be suitable for the vehicular context.

5.1.3 Vehicular storage needs for the requester

In the time interval between the offence and the notification (referred to as t_{gap}), the Requester has to store (1) its in-vehicle sensor information and (2) the set of received beacons. The first information is needed to evaluate whether the received notification is fair or not based on its perceived driving behavior. The second information is required to build the evidence header, as the beacons of purported witnesses have to be included in this structure. It should be noted that the second information is different to that required to the witness: in this case, not only the beacon sensorial data must be stored, but the *whole beacon* itself.

Apart from these data, the Requester has to store the evidence headers that have not been acknowledged. However, it is estimated that this storage need is negligible as it is only necessary when evidence is to be created and only if the vehicular channel transmission is not successful.

Concerning the storage of sensorial information, it depends on five factors: the amount of sensors *nsen*, their sampling speed *samsp*, the size of the sensorial values $sval_i$ and the time mark of each sample *tmark*, and t_{gap}. Particularly, the requester storage RqSt is given by Equation 2.

$$RqSt = t_{gap} / samsp \cdot (tmark + \sum_{i=0\ldots nsen} sval_i) \quad \text{(Eq. 2)}$$

For the context of this contribution, only position and speed sensors will be considered (*nsen* = 2). The sampling speed *samsp* =100 ms. will be taken, which coincides with the beaconing rate assumed in current standards. The size of the sensorial value is 2 bytes for the speed value ($sval_0$) and 10 bytes for the positional information ($sval_1$). The time mark size *tmark* is 2 bytes [3]. Concerning the interval t_{gap}, the values 5, 30, 60, 180 and 300 seconds will be considered. Thus, in the worst case considered (t_{gap} = 300 s.), RqSt =42000 bytes.

With respect to the storage of received beacons, the calculation follows an analogous reasoning as that presented in Section 5.1.2. The difference is that in this case it is necessary to store the whole beacon, but only during the period t_{gap}, as former beacons are from vehicles that are not suitable as witnesses. Each beacon requires 199 bytes of storage, including its associated public key certificate. Considering this value and the previous ones for t_{gap}, in the less favourable context (i.e. the highest vehicular density and the greater t_{gap}), Requester stores 191,04 Mb. As it happened with the storage needs for the witness (recall Section 5.1.2), this amount is considered as suitable for vehicles, given that they do not have strict space/weight/battery constraints for storage devices.

5.1.4 Experimental evaluation

In order to assess the amount of testimonies per evidence that may be achieved in a road scenario, several simulations have been conducted using the NS-2 simulator. Five representative scenarios have been considered, namely an urban section from the city of Eichstätt, a highway stretch, a highway crossing section, a secondary road and a Manhattan-like map. In each one, 250 vehicles have been simulated over 600 seconds. Every 10 seconds one randomly chosen vehicle is nominated to be committing a speeding offence, thus launching the

proposed protocol. The vehicular movement has been created using both SUMO [3] and CityMob [4].The transmission parameters are derived from the expected performance of DSRC communications including the wireless frequency (5.9 Ghz), data rate (6 Mb/s) and reception range (300 m.) [1]. With respect to the routing strategy, one-hop broadcast has been chosen. As this is the most basic dissemination strategy (as there is no forwarding between nodes), it avoids delays caused by a routing strategy such as [18] or its associated threats like black hole attacks [19].

An intuitive assumption is that the smaller t_{gap} is, the closer (consequently, the more reachable) the Witness may be from the Requester. Therefore, this analysis will be focused on determining the effect of t_{gap} in (1) the proportion of valid witnesses that are reachable and (2) the amount of testimonies that will be sent for each offence. The first indicator shows the relationship between the *potential* witnesses and the *actual* witnesses, whereas the second one shows the total amount of *actual* witnesses. In this way, it is possible to characterize both the achieved and missed testimonies.

Figure 3 shows the ratio of available witnesses in each scenario, using 5, 30, 60, 180 and 300 seconds for t_{gap}. Except from the highway, around 90 % of the witnesses are available if t_{gap} = 5 s. On the contrary, for t_{gap} = 300 seconds this proportion drops below 30 %. For the intermediate value of t_{gap} = 60 seconds, all scenarios except the Manhattan map allow for a proportion of around 50 %. There are two facts that should be analysed separately. First, the highway scenario never offers a ratio higher than 52 %. This is due to the high speed of vehicles, along with their potential greater speed differences, making it more probable to get out of range very soon. Second, the ratio offered by the Manhattan map gets lower faster than the remaining ones, significantly before t_{gap} = 30 seconds. This fact is a consequence of the map definition – once a vehicle turns in a street, it starts driving in a perpendicular direction to the other one.

On the other hand, Figure 4 shows the amount of available testimonies in each scenario considering the aforementioned values for t_{gap}. The highway scenario is the most convenient one, as it offers the maximum amount of testimonies for all values of t_{gap}. Remarkably, 38 testimonies are collected for t_{gap} =5s. This may be explained by the multi-lane feature of this kind of roads, which enables more vehicles to be in range. On the contrary, the Manhattan map is the one that offers the lower amount. This fact may be due to the fast dispersion of vehicles in this map according to the considered mobility pattern.

Although the amount of required testimonies to endorse a given claim is up to the Adjudicator, we assume that having less than 10 testimonies may be inconvenient. Based on this assumption, this protocol may be used in highways for every t_{gap}, whereas in secondary roads it is not suitable for t_{gap} = 300 s. In the Eichstätt and highway crossing settings, it is only suitable for t_{gap} < 180 s. It is not suitable for the Manhattan map under this criterion.

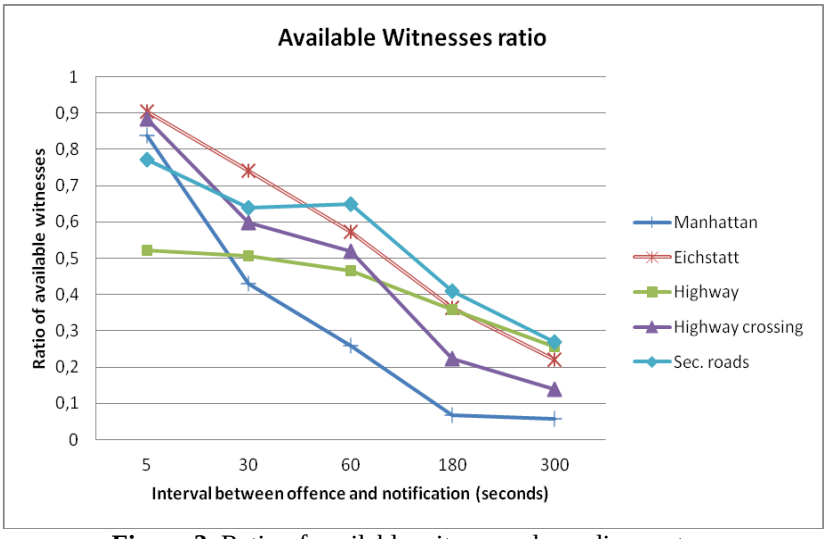

Figure 3. Ratio of available witnesses depending on t_{gap}

[3] http://sumo.sourceforge.net/, last accessed in July 2013.
[4] http://www.grc.upv.es/Software/citymob.html, last accessed in July 2013.

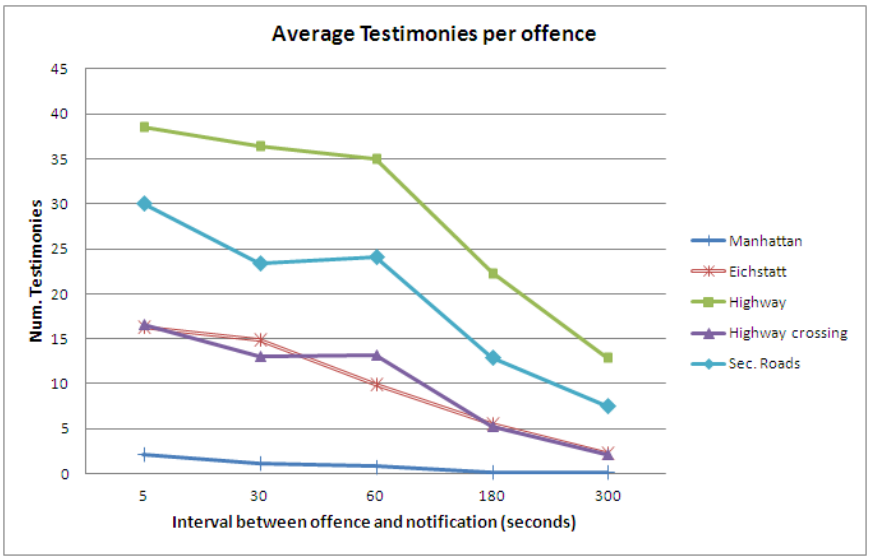

Figure 4. Testimonies per offence depending on t_{gap}

5.2 Security requirements analysis

This Section evaluates whether the imposed requirements are fulfilled.

Correctness. The verification process enforces that the evidence contains at least one supporting testimony (condition 1). In this way, evidences based on false claims by R are removed, as there would be no supporting testimonies. Moreover, the semantic checks ensure the consistency between at least one of the testimonies and R's claimed value (condition 2). The time consistency (condition 3) is also checked in the verification process. It must be recalled that this verification is possible since vehicles are assumed to be synchronized by means of the integrated navigation system. The verification process also checks that all pseudonyms at stake belong to a different entity (condition 4). Furthermore, it is checked that the W_is identified by R (i.e. listed in the evidence header) are the ones that generate the supporting testimonies.

Concerning the threat of messages never created or lost, the use of a resilient channel contributes to mitigate it for all messages except from requests. In such a case, the Testimony exception handling enables collecting the testimony even if the request was not received by the witness. Therefore, even if the request is lost, the correctness is not threatened. With respect to the message alteration, the use of digital signatures (created in a secure environment) makes it possible to detect this threat and to avoid impersonation.

Confidentiality. All messages exchanged in the vehicular environment are encrypted to its receiver. All messages exchanged in the vehicular environment are encrypted to its intended receiver – the request (encrypted to each W_i), the testimony and the evidence header (to the EM). Moreover, the created evidence is sent encrypted to Adj. These data are securely managed by their respective receivers (SAE_{Wi}, EM and Adj, respectively).

Authentic requests. The contents of the request ensure that R is the same entity to which the evidence has to be referred, as it has one part digitally signed under its identity ($R(t_{off})$). Moreover, another part is signed under its current identity ($R(t_{req})$), which prevents third parties to issue requests referred to others. The time mark t_{off} introduced in the first part counters the potential threat posed by replay attacks.

Authentic testimonies. The verification process checks the plausibility of a given testimony. In this way, sensor errors (accidental or on purpose) are properly handled if these checks offer enough reliability. Therefore, the proposed approach satisfies this condition as much as real-life Court situations: witnesses may be good-willing but they may offer wrong testimonies due to their perception errors.

Moreover, the Secure Location Verification process, along with the beacons contained in the evidence header, ensure that the witness was present when the offence was committed. As the employed cryptographic material is securely loaded into the HSM, and given that this device is firmly attached to the vehicle, only this vehicle (which is necessarily different to R) is able to correctly sign a message.

6 Related work

The small amount of contributions related to evidence generation in vehicular scenarios is described here. The most representative ones are related to accident reconstruction. In [1], Hardware Security Modules (HSMs) are employed to register all the events produced by the own car. Once the crash has happened, involved vehicles send informative beacons to the surrounding vehicles to alert them. The contribution developed here takes into account not only the own vehicle's sensor measurements, but also data coming from surrounding vehicles. As it is now required for an attacker to compromise (or to collude) surrounding vehicles, the threat is lower than that of [1].

Evidence generation is also present in the security framework presented by Lin *et al.* [20]. Their focus is on building a secure and private communication protocol that ensures efficient traceability when needed. Thus, they consider as evidence a signed message sent by a given vehicle, using ID-based cryptography and group signatures as the underlying cryptographic mechanisms. In this paper, the evidence is a signed data structure that contains a signed claim by the requesting vehicle and a set of supporting (signed as well) testimonies. Furthermore, the cryptographic approach selected is based on public key cryptography, being compliant with the IEEE 1609.2 standard [15].

Given that the set of witness vehicles for a given one in a certain moment could be seen as a group, previous works on group formation and communication could be considered [21, 22]. Given the high mobility of vehicles, it could be possible that the intended group members were not present when they are requested. Thus, the use of group communications have been discarded in this work, as this choice would not always be suitable.

7 Conclusions. Future work

In this work, a protocol for creating and verifying evidences about a vehicle's recent behavior has been presented. Data employed for creating this evidence is obtained from neighbouring vehicles, which act as witnesses. The impact on the vehicular network and computational resources has been illustrated, as well as its suitability to representative road scenarios through simulations.

Ongoing work is focused on considering other alternative protocol designs, beyond the comparison presented in this paper. In the future work, the suitability of retransmission strategies and state-of-the-art routing algorithms will be studied, as well as other communication technologies (such as GPRS). Moreover, the suitability of the protocol in road traffic scenarios with a heavy workload (e.g. greater amount of detected offences per second or other coexistent ITS services) will be assessed. Finally, a real-world implementation of the proposal is envisioned, to ensure the suitability of the proposal in a practical context.

Acknowledgement

This work has been partially founded by Ministerio de Ciencia e Innovacion of Spain (project E-SAVE, grant TIN2009-13461), which has encouraged us to submit our results to relevant scientific publications. Authors would like to thank Dr. André Weimerskirch for providing us the performance figures of the commercial OBU device. Also, authors thank the reviewers for their useful comments that helped in improving this work, as well as Mr. Jan Holle for his comments on this research line.

References

[1] S. U. Rahman, U. Hengartner. Secure crash reporting in vehicular Ad hoc networks. In: Proc. of International Conference on Security and Privacy in Communications Networks. 2007.

[2] M. Wolf, A. Weimerskirch, C. Paar. Security in automotive bus systems. In: Proc. 2nd Workshop on Embedded Security in Cars (ESCAR), 2004.

[3] Society of Automotive Engineers (SAE). J2735 standard: Dedicated Short Range Communications - Message Set Dictionary. 2009.

[4] N. Lo, H. Tsai. Illusion attack on VANET applications - A message plausibility problem. In: Proc. IEEE Globecom Workshops. 2007.

[5] J. M. de Fuentes, A.I. Gonzalez-Tablas, A. Ribagorda. Witness-based evidence generation in Vehicular Ad-Hoc Networks. In: Proc. Embedded Security in Cars Conference (ESCAR), 2009.

[6] M. Raya, *et al.* Eviction of misbehaving and faulty nodes in vehicular networks. IEEE Journal on Selected Areas in Comms. Vol 25. No.8. (2007)

[7] J. M. de Fuentes, A. I. Gonzalez-Tablas, J. L. Hernandez-Ardieta, A. Ribagorda. Towards an automatic enforcement for speeding: enhanced model and ITS realization. IET Intelligent Transport Systems, vol.6, no.3, pp.270-281, September 2012. Available at: http://hdl.handle.net/10016/14412
[8] F. Kargl, *et al.* Secure vehicular communication systems: implementation, performance, and research challenges. IEEE Communications Magazine. Vol 46, No. 11. (2008)
[9] Papadimitratos, P. *et al.* Secure vehicular communication systems: design and architecture. IEEE Communications Magazine. Vol 46, No. 11. (2008)
[10] Q-Free. CVIS platform: The future of Intelligent Transport Systems. CVIS project. 2010.
[11] E. Fonseca, A. Festag, R. Baldessari, R. Aguiar. Support of Anonymity in VANETs - Putting Pseudonymity into Practice. In: Proc. of Wireless Communications and Networking Conference, 2007.
[12] H.Cankaya, *et al.* Towards a shared digital communication platform for vehicles. In: Proc. of ITS World Congress, 2011.
[13] D. Chuan, Y. Lin, M. Linru, C. Yuan. Towards a Practical and Scalable Trusted Software Dissemination System. FTRA Journal of Convergence,Vol.2, No.1. (2011) 53-60
[14] J.-L. Ferrer-Gomilla, J. A. Onieva, M. Payeras, J. Lopez. Certified electronic mail: Properties revisited. Computers and Security. Vol. 29 No.2 (2010) 167-179.
[15] Institute of Electrical and Electronics Engineers (IEEE). 1609.2: Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages. (2006)
[16] J. B. Kenney, Dedicated Short-Range Communications (DSRC) Standards in the US. Proceedings of the IEEE. Vol. 99, No. 7. (2011)
[17] W. Viriyasitavat, *et al.* Network Connectivity of VANETs in Urban Areas. In: Proc. of IEEE Conf. on Sensor, Mesh and Ad Hoc Comm. and Networks. 2009.
[18] M. Imani, M. Taheri, M. Naderi. Security enhanced routing protocol for Ad hoc networks, FTRA Journal of Convergence, Vol.1, No.1. (2010) 43-48
[19] F. Tseng, L. Chou, H. Chao. A survey of black hole attacks in wireless mobile ad hoc networks. Human-centric Computing and Information Sciences (HCIS), Vol 1, No. 1, (2011). http://www.hcis-journal.com/content/1/1/4
[20] X. Lin, X. Sun, P.-H. Ho and X. Shen. GSIS: A Secure and Privacy-Preserving Protocol for Vehicular Communications. IEEE Trans. on Vehicular Tech. Vol. 56, No. 6. (2007)
[21] A. Aikebaier, T. Enokido, M. Takizawa. Trustworthy Group Making Algorithm in Distributed Systems. Human-centric Computing and Information Sciences (HCIS), Vol 1., No.1. (2011) http://www.hcis-journal.com/content/1/1/6
[22] M. Raya, A. Aziz, J.-P. Hubaux. Efficient secure aggregation in VANETs. In: Proceedings of the 3rd ACM international workshop on Vehicular ad hoc networks, 2006. New York, NY, USA: ACM. pp.67-75.