Transportation Research Part C xxx (2011) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Hindering false event dissemination in VANETs with proof-of-work mechanisms

Esther Palomar *, José M. de Fuentes, Ana I. González-Tablas, Almudena Alcaide

Department of Computer Science, University Carlos III of Madrid, Avda. Universidad 30, 28911 Madrid, Spain

ARTICLE INFO

Article history:
Received 14 December 2010
Received in revised form 4 July 2011
Accepted 15 August 2011
Available online xxxx

Keywords:
VANET
Event dissemination
Data authentication
Digital certificate
Proof-of-work
Non-repudiable evidence

ABSTRACT

Solutions for a secure data dissemination in Vehicular Ad-hoc Networks (VANETs) are increasingly adopting robust cryptography schemes as more lightweight, trust-based approaches often lead to serious problems such as illusion, collusion and sybil attacks, to name a few. This article shows that it is possible to hinder the dissemination of false warning events in VANETs and limit the amount of messages a node can send within a given period of time by applying well-known cryptographic techniques. To achieve this twofold goal, the method we present is based on two simple concepts already used in security mechanisms to provide accountability and to combat spam and denial of service attacks, namely the use of certificates and Proof-of-Work (POW) systems respectively. Basically, our scheme not only discourages nodes from transmitting fake event warning messages but also serves as an effective non-repudiation evidence for different types of dishonest behavior within a VANET. Our analyses on both the performance and security of our scheme show its feasibility in VANETs.

1. Introduction

Vehicular Ad-hoc networks (VANETs) are a promising communication scenario which allow vehicles to interchange information. In this way, new applications and services can be offered to the vehicle, leading to improvements on different issues like road safety or infotainment (Olariu and Weigle, 2009). Driving assistance is one of the key areas to apply these improvements. Receiving data referred to the traffic status can help the driver making the appropriate decisions. It allows the driver to better know the current context, as her knowledge is not restricted to what is on her field of vision. In particular, such vehicular emergency warning systems cooperatively communicate with each other when they detect a dangerous event, like a bottleneck or a slippery road. The message structure ‘A la carte’ from the *SAE J2735 standard* enables such a warning communication (SAE, 2009a).

Basically, whenever a potential warning road condition is detected by the on-board component called On-Board Unit (OBU), the warning message system generates a new message and disseminates it beyond the immediate transmission range. As the received data can affect the driving task, they must be trustworthy. In other words, the received data must accurately reflect the road traffic status. Data forgery or alteration must be totally avoided. However, there are several ways to perform attacks in this area. In fact, each phase of the associated data management cycle (i.e. data creation, transmission, storage and evaluation) can be compromised, as shown in Fig. 1. In this data creation phase, both the sensors and their connections between the OBU and the Hardware Security Module (HSM) can be easily attacked and, therefore, fake data can be created (Wolf et al., 2004). With respect to the transmission, message routing in VANET also allows the intermediary nodes

* Corresponding author. Tel.: +34 916249422; fax: +34 916249129.
E-mail addresses: epalomar@inf.uc3m.es (E. Palomar), jfuentes@inf.uc3m.es (J.M. de Fuentes), aigonzal@inf.uc3m.es (A.I. González-Tablas), aalcaide@inf.uc3m.es (A. Alcaide).

0968-090X/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trc.2011.08.002

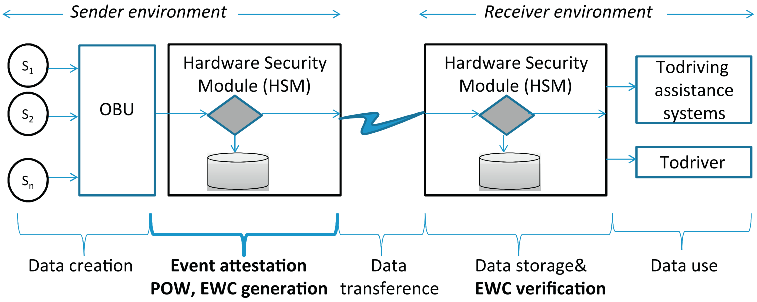

Sender environment
Receiver environment
S1
S2
Sn
OBU
Hardware Security Module (HSM)
Hardware Security Module (HSM)
Todriving assistance systems
Todriver
Data creation
Event attestation POW, EWC generation
Data transference
Data storage& EWC verification
Data use

Fig. 1. Phases of the data management cycle that our proposal implements with new functionality, i.e. in both the sender's HSM to generate the evidences and the receiver's HSM for verifying such evidences.

to manipulate the relayed data (Golle et al., 2004). Finally, the storage and evaluation phases can as well be attacked if they are not performed in a HSM (Papadimitratos et al., 2008a), with the consequence of storing fake data or getting a wrong conclusion from data evaluation.

In particular, suppose a scenario where the major motivations for the attacker are to either prevent other vehicles from taking some road segment or indirectly suggest an alternative one. In this context, she may create a false traffic-jam warning message regarding that route and disseminate it in bulk. Similarly, if a vehicle happens to suffer from a sensor's accidental defective functioning, it could start, without its keeper knowing it, sending a significative amount of wrong messages while circulating. As a consequence, receiver vehicles will have to process these messages and dedicate some time to assess the plausibility of the message, which would be more or less effective depending on the specific error in the sender vehicle. However, plausibility checks (Ostermaier et al., 2007; Lo and Tsai, 2007), by their own, do not prevent receivers wasting its processing resources. In fact, if intelligently planned, plausibility checks can be ineffective for receiver vehicles to detect such false messages. This situation worsens even more if the attacker controls several nodes, as it is usual to assume in this context that a message will have more reputation if received from several sources or more times. These scenarios suggest that an a-priori countermeasure to limit such a flooding-based attack is needed, whilst an accountability service is also desired as an a-posteriori mechanism to penalize such dishonest behaviors.

Given the huge amount of attacks that can compromise the data dissemination, several research contributions have focused on designing mechanisms to face this threat (Raya and Hubaux, 2007). However, they are mainly intended to avoid the undesirable consequences that can happen from a false context information given by other vehicles such as the so called *intelligent collisions* (Blum and Eskandarian, 2004). In addition, most of them concentrate on strengthening the evaluation phase, specially by introducing reasoning techniques to decide the plausibility of a given message. Despite their promising results, these techniques, when deploying separately, do not offer a complete protection against the mentioned attacks.

1.1. Overview of our proposal

In this work, we present a complementary approach to that of the previous contributions. Our goal is to hinder the dissemination of false event warning messages in VANETs and limit its undesirable consequences in receiving nodes, including the waste of resources. To achieve this two-fold goal, we propose a mechanism that, on one hand, provides accountability regarding the messages sent by a node and, on the other hand, decreases the very creation of false messages a node can send to some extent.

For the latter purpose, we propose a mechanism that requires senders of event warning messages (EWMs) to perform a non-negligible amount of computation and then attach an evidence of this computational work to the EWMs, yet original in the VANET context up until now. This computation cost (called *Proof-Of-Work*, POW) should be enough to discourage dishonest vehicles from flooding the same message (e.g. under different pseudonyms or compromising multiple nodes for that purpose). Moreover, the workload imposed by the POW should be reasonable enough according to, first, the computation capabilities of current OBUs and, second, the immediacy required for vehicular communications. In this context, POW mechanisms serve as an a-priori countermeasure against warning data forgery (Daza et al., 2008; Ostermaier et al., 2007; Golle et al., 2004) by imposing a limiting factor for the amount of messages that vehicles could send. Experiments conducted will show that the selected POW functions set the necessary upper bounds to the amount of fake messages transmitted whilst demanding a moderately hard effort for legitimate users.

To give evidence about the performed POW, inter-vehicle warning messages will contain a new structure called Event Warning Certificate (EWC) that comprises the event-related data along with the POW evidence, among other fields. Thus, a given EWM is only considered if there is a valid evidence that attests that the sender performed the required POW, i.e. a moderately hard computational task which is securely parameterized to avoid pre-computations.

With regard to the former goal, our proposal acts also as an a-posteriori countermeasure against vehicles who transmit fraudulent EWMs since senders' signatures attached to EWCs enable tracing dishonest transmissions afterwards (Raya and

Hubaux, 2007). EWMs are signed by the sender, allowing a posteriori identification of nodes that have sent messages containing false data, making it possible to hold senders accountable for their actions, and the taking of the appropriate measures (repair of the malfunctioning sensor or demand of liabilities). In this context, though anonymity is a must, conventional entities for authentication do exist to avoid attackers sending fraudulent messages without a punishment (Papadimitratos et al., 2008b).

Thus, our approach is complementary to existing contributions that could be referred to as a-posteriori mechanisms (Armknecht et al., 2007), since false data can still be created. Although the POW makes it more difficult for attackers to spread a significant amount of forged messages, it does not avoid them but does limit the attackers' capability to flood the VANET with false warnings. Moreover, unintentional sensorial errors (e.g. those accidentally produced, not caused by malicious manipulation) can happen. Hence, once a warning message is received, recipients should verify all the EWC fields in order to be sure that they are not being cheated on. Plausibility checks are still necessary, although they are not the only existing countermeasure.

Finally, static road-side units (RSUs) can (i) work on-line and off-line to, when available, store and transmit EWMs as well as recollecting every evidence aimed at consequently penalizing misbehavior for threatening road security, (ii) produce a series of POWs within the nearby area in case a non-interactive POW scheme is deployed, or new POWs on-demand in case of the interactive POW scheme, (iii) inform vehicles of their own penalties when illegal actions have been committed, and also (iv) perform aggregation tasks in order to validate a subset of EWMs received from different nodes. Note that, for this purpose, the more EWMs collected, the more confidence the authorities will have of the observed event.

1.2. Article organization

The remainder of the article is organized as follows. First, an overview of the fundamental building blocks of our proposal, i.e. the main POW approaches found in the literature as well as the application of digital signatures in VANETs, are presented in Section 2. In Section 3 we propose a secure event dissemination scheme by creating non-repudiable evidences based on the use of certificates and POW mechanisms. Performance and security analysis are presented in Section 4 and 5, respectively. Section 6 presents a brief comparative discussion of our proposal and some related work. Finally, in Section 7 we establish the main conclusions.

2. Background

2.1. Proof-of-work mechanisms

The idea of demonstrating a computational cost performed in a specified interval of time, i.e. the well-known *Proof-of-Work* (POW) system introduced by Dwork and Naor in Dwork and Naor (1992), is still being the basis of a number of recent security protocols (Borisov, 2006). Basically, two entities are involved in such a process, most in the way of a *challenge–response* protocol, in which usually one party (the *verifier*) asks the other (the *prover*) to complete a simple test or puzzle before granting access to a resource or providing a certain service. Provers cannot obtain the requested material without expending a minimal amount of computational resource, and showing the expected evidence.

There are two classes of POW protocols namely non-interactive and interactive. In the non-interactive POW approach, a number of challenges are first computed in bulk and then centrally stored together. Provers select their own challenges or, in other cases, a random start value. On the other hand, the interactive scheme involves a traditional challenge–response mechanism, as follows:

$$\begin{array}{ll} Prover \rightarrow Verifier: & Request \\ Verifier \rightarrow Prover: & Challenge(\cdot) \\ Prover \rightarrow Verifier: & Challenge's\ solution \end{array} \tag{1}$$

Regarding the underlying functions generally used by these schemes, different primitives have been applied as a defense against spam and denial of service attacks, among others (Jakobsson and Juels, 1999). By definition, the function is expensive to solve, while staying comparatively cheap on the verifier side. The global aim is to limit the capabilities—resources and time—of adversaries since spammers (even using botnets) cannot compile unlimited amounts of processing time at their disposal. Thus, challenges are based on either CPU-bound cost-functions or memory-bound functions (MBFs).

On one hand, the Client-Puzzle Protocol introduced by Juels and Brainard in Juels and Brainard (1999) uses cryptographic puzzles for preventing a communication protocol such as TCP and SSL from connection depletion by rate limiting TCP connections. Client puzzles apply a probabilistically bounded cost-function (based on the typical one-way hash-function inversion problem) which imposes an upper bound on the cost of finding a known solution within some key space 2^k.

Therefore, by definition, the functions described above are expensive to solve, while staying comparatively cheap on the verifier side. Similarly, it may be the case that POWs are built using functions that present some sort of secret *shortcut* or *trapdoor* that makes the computation or verification easy, e.g. a shortcut to decryption. The underlying idea of the application of shortcuts is to define in one way or another the computational effort required to provers, whilst establishing boundaries to puzzles, namely the hardness or difficulty of puzzles. To this regard, various types of trapdoor functions have been proposed.

Please cite this article in press as: Palomar, E., et al. Hindering false event dissemination in VANETs with proof-of-work mechanisms. Transport. Res. Part C (2011), doi:10.1016/j.trc.2011.08.002

We say F is a trapdoor function if there exists some secret information ω, such that given $F(x)$ and ω it is easy to compute x and otherwise not (Syverson, 1998). In some cases, verifiers maintain a secret, e.g. some bits of the secret key, which cannot be revealed for a predictable time delay or until a predictable amount of computation has occurred. In other cases, the verifier reveals some piece of the secret along with the cryptographic puzzle.

On the other hand, Abadi et al.'s work (Abadi et al., 2003) presents an alternative computational approach based on memory latency, namely MBFs, originally introduced to deal with heterogeneous hardware. Before MBFs, the more powerful participants may be able to solve puzzles faster than others. Now, the computation time is dominated by the time spent accessing memory, and can be used to ensure that every node will spend approximately the same amount of the critical resource. The analysis and evaluation of several MBF potential approaches (Dwork et al., 2003) result in a constant performance across different machines.

2.2. Digital signatures in VANETs

By definition, a valid digital signature gives a recipient reasons to believe that a message was created by a *known* source, and that it was not altered in transit. Node authentication in VANET can be supported by a Public Key Infrastructure (PKI) that assists in identifying nodes (vehicles or drivers) providing them with digital certificates. In a VANET PKI (VPKI), a Certificate Authority (CA) issues keys and certificates to vehicles, as follows. The standard IEEE 1609.2-2006, namely Wireless Access in Vehicular Environments (WAVE), defines a series of mechanisms to assure a secure access to the VANET. This standard establishes the appropriate message format and also defines a basic certificate structure whilst supporting different certificate types. A certificate, namely WAVE certificate, consists of a digitally signed document binding a public key to an identity.

In this context, each node possesses a public key certificate establishing a public/private key pair. Thus, given a signing algorithm (note that the standard establishes ECDSA), messages can be signed by originators using their own private material, whilst recipients verify signed messages using the source's public key certificate.

Usually, a range of *revocable* pseudonyms is assigned to VANET nodes as identifiers ensure concealment of its real identifier, and therefore protect the privacy of its drivers (Lu et al., 2008). Although pseudonym systems suffer from several drawbacks (Calandriello et al., 2007), they are an effective countermeasure against traceability, i.e. the indiscriminate disclosure of the whole activity of a certain entity, but allow at the same time a trusted third party (TTP) to re-construct the true identity when needed (Papadimitratos et al., 2008b). Indeed, the existence of such an identification option is mandatory if non-repudiation of origin needs to be provided.

3. Specification of the proposed method components

3.1. Principles

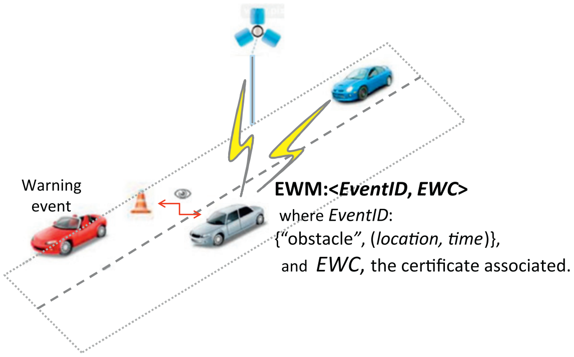
Before presenting the details of this proposal, we assume the following six working hypotheses:

1. Wireless routing and data forwarding are assured by a broadcast-based routing protocol. Thus, EWMs will be broadcasted to other nearby nodes (vehicles or road-side stations).
2. Identification of all participants is required through the corresponding public key certificate, i.e. nodes are authenticated by valid IEEE 1609.2 WAVE certificates, which are digitally signed and issued by the CA. Thus, each node n_i has a public key certificate PKC_{n_i} (containing n_i's public key, K_{n_i}) and the private key denoted by $K_{n_i}^{-1}$. However, to assure pseudo-anonymity and avoid traceability, such user certificates should be temporary and anonymous and then contain a pseudo-id of the real identity.
3. Each event is described by the tuple:

   $$< EventID := Type\text{-}of\text{-}event, (Location, Time) >$$

   where the *Type-of-event* element is chosen among a set of possible tags (e.g. accident, traffic congestion, road closing, obstacle and so on). For instance, the International Traveler Information Systems (ITIS) defined elements can be considered for describing any event (SAE, 2009b). *Time* and *Location* represent the time given by the HSM's built-in clock and the GPS-supported place (i.e. a mile/kilometer marker post) in which the event occurred, respectively. Data above is also used for indexing events.
4. Each EWM transmitted consists of the aforementioned information related to a certain event observed, along with the associated EWC that assures data authentication and confirms non-alteration and source authentication of the message. The EWC structure is described in sections below. Therefore, exploiting the fact that legitimate vehicles' HSM make it easier to assure a nearly unforgeable summary of the hardware and software configuration, among many capabilities (Guette and Bryce, 2008; Papadimitratos et al., 2008a), we assume that the vehicles' OBU and HSM are responsible for securely generating such a digital evidence. In particular, several works-in-progress, such as the EC FP7-ICT-2007 EVITA Project (EVITA, 2007), elaborate on the development of a trusted in-vehicle environment as a *black box* based on the application of trusted platforms.

Please cite this article in press as: Palomar, E., et al. Hindering false event dissemination in VANETs with proof-of-work mechanisms. Transport. Res. Part C (2011), doi:10.1016/j.trc.2011.08.002


Warning event
EWM:<EventID, EWC>
where EventID:
{“obstacle”, (location, time)},
and EWC, the certificate associated.

Event Warning Certificate EWC

Certificate C_e:	
Serial-Number:	I_c
Source:	PKC_A
Message:	$h(EventID)$
Movement:	$< pos, dir, vel >$
POW:	$\mathcal{POW}(\omega, seed)$
Validity Period:	(ts_i, ts_f)
Crypto. Algorithms:	$Sign.Algorithm$ $Sign.Hash.Algorithm$ $m.Hash.Algorithm$
Signature:	$s_A(C_e)$

Fig. 2. The event warning message and certificate’s structure.

5. Similarly, we assume that the used cryptographic primitives, such as the digital signature scheme and the cryptographic hash functions, are secure and cannot be compromised by an attacker. Hence, we also assume that the on-board HSM controls the access to the private keys so avoiding the forgery of digital signatures by attackers.
6. Anyone can verify the authenticity of a sender’s signature with the help of the sender’s certificate PKC_{n_i} enclosed in the EWC. To this regard, we assume that a reliable CA hierarchy and revocation information distribution scheme exists, such as the security architecture for vehicular communication presented in Papadimitratos et al. (2008b).

3.2. Structure of event warning certificates

The event warning certificate *EWC* (see Fig. 2) is structured in two parts, as follows:

- The contents of the certificate, denoted by C_e, with the following fields:
  1. A serial number I_c of the EWC issued by the originator A much in the way of an additional auditory feedback for the public authorities.
  2. The public key certificate of the originator, PKC_A, which ultimately establishes who has directly observed the event.
  3. The digest, $h(EventID)$, of applying a cryptographic hash function to the event warning data *EventID*, assuring its integrity. Note that *EventID* is represented by the tuple ⟨Type-of-event, (Location, Time)⟩.
  4. A register containing the originator vehicle’s movement information such as position, direction, and speed. Hence, vehicles traveling in the opposite direction can easily discard non-relevant EWMs.
  5. The evidence of computing the solution of a certain POW function through a unique, fresh seed generated for each event warning data and a certain ω computational cost, $\mathscr{POW}(\omega, seed)$, thus preventing huge amounts of messages from being sent in a short period.
  6. The validity period[1] (ts_1, ts_2) for *EWC*, establishing that the certificate is valid from ts_1 until ts_2. Likewise the time-to-live (TTL) field defines for a message in a VANET, this validity period (and the signature attached) prevents attackers from reviving an outdated EWC and also prevents EWMs from being everlasting.
  7. Descriptions of the hash and signature functions which have been used.
- The signature, $s_A(C_E)$, generated over the previous fields by the originator of the EWC.

[1] Though the EWC’s validity period (which does not exceed the PKC’s lifetime) should be determined according to the average vehicle density ratio and speed for the target area, we can determine it by assuming a fixed value of hops (Naumov et al., 2006).

3.3. Event warning message generation

Let *A* be an honest originator of a given *EWM*. Before disseminating the *EWM*,*A* must:

- Complete a given computational problem. As described in Section 2, the way in which this task is carried out can be defined by the methods below:
  1. Following a non-interactive POW approach: We have also identified two alternatives to create non-interactive fresh puzzles, as follows:
     – An especial RSU within the target area dumps a limited number of pre-computed puzzles (called *pre-puzzles*) to surrounding vehicles' on-board units which securely backup the puzzles received. In this context, for every EWM to be distributed the HSM will pick up a random element in the set to compute that pre-puzzle's solution. This pre-puzzle list is periodically flushed by the corresponding RSU and re-distributed to nearby vehicles in order to avoid pre-computations.
     – The sender's HSM is responsible of computing a series of puzzles by utilizing a secure time base provided by the HSM's built-in clock (Papadimitratos et al., 2008a) as the source for fresh seeds. This pre-puzzle list is securely stored and periodically flushed by the HSM.
  2. Following an interactive POW approach, in which the RSU nodes will be responsible of a new puzzle creation and its distribution in response to the specific senders' requests. These requests will present the observed event-related data together with the HSM time as the fresh seed in order to avoid pre-computations. This method may introduce a very low extra communication overhead whilst contributing robustness to the process of puzzles creation.

Therefore, the EWC compulsorily contains a POW together with a trapdoor function to supply receivers with an advantage for easily verifying the POW (see Section 4 for further details).

- Complete all other EWC's fields.
- Attach her signature.

Finally, *A* sends by broadcast the message *EWM* to the nearby nodes, along with the *EWC*:

$$EWM := (EventID, EWC)$$

3.4. Event warning message verification

Let *B* be a recipient of a *EWM*. We can assume that *B* is still located far away from the road incident and therefore has no visibility of it. *B* has to perform a *local verification* stage in order to ensure that she is not being cheated on. This consists in the following steps:

1. *EWC verification*. The correctness of the EWC generation should be verified to ensure the authentication and integrity of the *EWM*, so *B* must verify at least the correctness of the following items:
   - *B* computes $h(EventID)$ from *EWM* and compares the result with that included in the certificate. If both values differ, then either *EWM* has been altered or *EWC* is not an authentic certificate for *EventID*.
   - *B* validates the POW, evaluating the puzzle's solution. We further elaborate on POW evaluation in the Section below.
   - *B* verifies the sender's signature, $s_A(C_e)$, by utilizing the public key material in PKC_A, included in the EWC, and the validity of the sender's certificate PKC_A as well.
2. *Local management*. Recipients dump received EWMs on a local database. Plausibility checks should be performed here in order to verify the liability of the received message (Golle et al., 2004).
3. *Valid EWC management*. If previous verifications succeed, the node will relay the same message if and only if both, the *EWC*'s and PKC_A's validity periods, are not expired.
4. *Invalid EWC management*. In case that the previous verifications fail, the received message is not trustworthy and a reactive mechanism must be put in practice to avoid the sender continue disseminating such unreliable data. On one hand, if there is no valid POW evidence within the EWC, either the sender's HSM is not working correctly and could be considered as tampered or the POW is intentionally fake and then should be classified as a punishable attack. On the contrary, if there is a valid POW, a fail in the plausibility checks could also be due to sensor malfunctioning,[2] apart from intentional manipulation. On the other hand, if the signature verifications fails, either the message has been corrupted in transit and could be then considered as a communication failure e.g. a noisy channel, or maliciously manipulated in order to launch, for example, an impersonation attack (see the security analysis presented in Section 5 for further details).
   In any case, the authority must have knowledge of this situation to take the appropriate measures, like appointing a revision date for the vehicle and sending a notification to the vehicle's responsible entity or even revoking all the credentials (public

[2] This represents a major challenge since, in theory, truthful warning events cannot be entirely validated until reaching the obstacle indeed. Several works such as Ostermaier et al. (2007) use *revocation messages* to inform others that the reported event has possibly disappeared. However, we believe that disadvantages involved, specially when attackers collude and then provoke false positives, limit their applicability.

key certificates) associated to that vehicle. However, the information submitted by a single vehicle is not enough to take such decision. Previous proposals, like the Local Eviction of Attackers by Voting Evaluators (LEAVE) protocols, have addressed this issue by involving several vehicles in the process (Raya et al., 2007).

3.5. A note on the role of road side units

In this section we discuss about the functionality provided by the RSUs deployed over the VANET. First, infrastructure nodes would act as vehicles to deliver warning information to all nearby nodes within the network, i.e. they are capable of either creating or retrieving EWMs as well. To this regard, although our model was primarily conceived to work in an ad hoc manner, we should consider that even a very limited number of RSUs can largely improve the dissemination process by an aggregation scheme, as authors state in Lochert et al. (2008). Though this aspect and others regarding the placement of the roadside infrastructure are out of the scope of this work, we stress here the importance of these supporting units over our proposal. In fact, RSUs are particularly useful in low traffic density area to extend the EWMs' visibility far away from the observed event to some extent. Therefore, as the EWC contains its expiration date, the corresponding EWM will be relayed not only by vehicles but also by adjacent RSUs to reach vehicles in close proximity.

Secondly, RSUs would act as evidence collectors especially justified if there is a fine penalty and accountability service behind to punish the incorrect behavior. This enables the legal accountability and auditing agency to correlate together all the EWMs that a certain vehicle unit has reported as potential offenses. For instance, as we stated before, the appropriate CA can disclose node identities, when necessary, to the extent permitted by applicable law.

Finally, RSUs can produce POWs to be sent to the vehicles within the nearby area. As mentioned before, a series of pre-puzzles is distributed to all vehicles located within the target area in case of a non-interactive POW scheme is deployed. In this scenario, there is an especial RSU per geographic region in charge of the dissemination of the pre-puzzle list. Pre-puzzles are created in advance and periodically flushed. Also, in case of interactive POW scenarios, RSUs can produce fresh puzzles on-demand in response to the senders' requests. In this case, the puzzle creation is performed once receiving the event-related data and time from the sender. As we discuss in the section below, the creation of the puzzle is straightforward (as the verification) and takes a negligible time contrary to the computation of its solution.

4. Analysis of the temporal and computational impacts of EWC

4.1. Costs of the event warning message generation

In this section, we estimate the computational effort EG required by senders to create an EWM. In particular, our analysis focuses on the cost of providing the POW evidence which represents the hardest operation in the entire EWM generation process, i.e. $EG = POW^{weak/hard} + H + S$ (see the estimation for the cryptographic operations used in Table 2).

As mentioned before, two main techniques exist for establishing the POW: metering the computational cost or the memory access delay. For the experiments conducted, we have implemented the former method although we also give estimations for common MBFs presented so far (Dwork et al., 2003; Abadi et al., 2003; Coelho, 2005) in which the limiting factor of reaching the solution is the memory access speed. As described in Section 3, our scheme distinguishes between two different approaches for a non-interactive POW scheme and only one interactive approach. Hence, to measure the impact of the POW generation, we assume the following interactive scenario where the cryptographic puzzle is delivered on-demand by a nearby RSU. Thus, a straightforward and efficient way of implementing such a construction is by using a block cipher[3] (e.g. the AES-CCM). More specifically, the puzzle consists of the cryptogram obtained from applying the block cipher and the key K_S to the plaintext to be encrypted, together with the trapdoor $\omega(l)$—a number l bits revealed out of k bits of the key K_S. To solve the puzzle for every EWC, it is required a different K_S to be discovered by brute force.[4] Hence, trapdoor values can be used to adjust the POW from a very hard to a moderately hard problem according to the number of l bits correctly revealed, as shown in Fig. 3. In this scenario, secret keys may be randomly generated by RSUs whereas plaintexts will consist of the event-related data sent by the requester. However, in our experiments, plaintexts are randomly generated and securely accessible at each round of the cryptanalysis process.

Note that, fresh seeds in non-interactive POW models are however more problematic to manage. Consider, for example, pre-puzzles based on the time source inside the sender's HSM which have been pre-computed various time periods before. This makes freshness verifications difficult to receivers. In this case, a simple way to provide freshness for puzzle constructions starts by applying hash-chain-reversal puzzles where both the seed and the hash chain are generated by the sender's HSM during the EWM generation. Thus, before transmitting the EWM, the sender is expected to spend her own resources to reverse the hash chain (i.e. solve the puzzle), and attach the valid HSM timestamp as the shortcut password to receivers.

For our experiments, we considered two block ciphers as the basis for such cryptographic puzzles, AES-128 (CCM mode) and TEA (Daemen and Rijmen, 2002; Russell, 2004). Both were coded in C, compiled with Microsoft Visual C++, and run on a

[3] Apart from these, there exist other possibilities to use cryptographic primitives for building up similar puzzles (e.g. hash functions in which a preimage of a given value has to be found).

[4] The effort required to recover the puzzle's solution can be adjusted from a very hard to a moderately hard problem according to the number of bits revealed in the trapdoor value. For instance, if a 256-bits key is used and the user is provided with a trapdoor value that reveals 250 bits of the key, then she has to perform 2^{6-1} decryptions on average to find the correct solution.

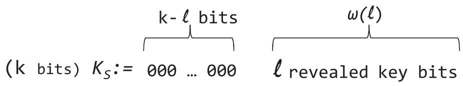


Fig. 3. A number of l trapdoor bits for a k-bits encryption key K_S.

standard platform: AMD ATHLON(tm)2600 2.09 GHz processor,[5] with 1GB RAM under Windows XP SP2. We have carried out 1000 experiments for different values of $k - l$ hidden key-bits, for a key length of $k = 128$, and also randomly varying the cryptograms and key used. We consider that more than 32 hidden key-bits would be impractical. For instance, with 2^{64} large number of potential keys to test, it literally takes a matter of years. Results[6] of the exploration effort using both algorithms and trapdoors with 20, 24, 28 and 32 hidden bits out of 128 bits are shown in Fig. 4. Experiments conducted with trapdoors of 20 bit and 24 bits give us weak puzzles, e.g. TEA with 20-bits trapdoors takes 0.01 s long on average. On the contrary, AES takes an unacceptable worst case of 5400 s with 32-trapdoor bits, while TEA takes 1980 s at worst. Thought we consider previous values non-admissible, however, this time decreases to less than 100 s at worst, for both, AES and TEA, when providing 28-bits trapdoors.

For each case, as the amount of the revealed l key-bits increases, the number of candidate keys obviously decreases, so the exploration time too. A factor contributing to the complexity of the puzzle is the cost of executing several decryptions, for testing each candidate key. For practical considerations, the requirements of the application in which the POW mechanism is used will determine the choice of the hidden material.

Finally, we use the same examples described in Abadi et al. (2003) to introduce the underlying concept behind MBFs and show its feasibility for our proposal. Let $F()$ be a function whose domain and range are integers in $\{0 \ldots (2^n - 1)\}$, where $2n$ is the number of entries in the array (Abadi et al., 2003). Suppose that $F()$'s inverse, $F^{-1}()$, cannot be evaluated in less time than a memory access. Suppose also that the challenge required from a prover consists of computing $F^{-1}()$ many times. Then it becomes very useful for the prover to build a table for $F^{-1}()$ and to rely on the table thereafter. In this context, experiments conducted in Abadi et al. (2003) estimate the time taken to build the table for $F^{-1}()$. More specifically, experiments show that costs of MBFs are about four times slower than the CPU-bound functions simulated on an standard platform.

In any case, computational overhead on nodes is within acceptable limits and sufficiently fast at each event occurrence.

4.2. Cost of the event warning message verification

Puzzles verification is straightforward mainly due to the fact that the POW evidence, included in the EWC, must contain both the puzzle and its solution, this is, in case of the scenario simulated, the cryptogram along with the encryption key used K_S. In this context, recent speed benchmarks for some of the most commonly used cryptographic algorithms establish a data transfer rate up to 61 millions of bytes (MiB) per second for the AES-CCM and 27 MiB/s for TEA in a standard platform. Hence, in order to analyze the cost EV of verifying an EWM, we have to consider the POW verification, hash comparison, verification of the signatures contained in the EWC and also the plausibility checks' cost, so $EV = VPOW + H + 2V + PCH$. From the estimations shown in Table 2, the entire EWM verification process takes less than 1ms long (see further details in the table).

4.3. Required distance between sender and a receiver

For the sake of completeness, it is interesting to evaluate the appropriate distance between senders and recipients which gives us the effective data rates and the consumption in the transmission of messages. We estimate the *EventID* size by an approximation to the attributes' length from SAE J2735 standard (SAE, 2009a). Thus, *EventID* just contains: (i) the field type-of-event which is two bytes in length (ii) the GPS-supported location is determined by a total of 14 bytes (more specifically, 4 bytes for latitude, 4 bytes for longitude, 2 bytes for elevation, and 4 bytes for positional accuracy), and (iii) two bytes field for the timestamp record. In summary, the *EventID* size is 18 bytes and, therefore, taking into account the size for EWC's attributes (see Table 1), we can estimate the *EWM* size at 260 bytes (18 B + 242 B).

Now, since the claimed rates for current generation of IEEE 802.11p VANET platforms range from 2.56Mbps in highly populated scenarios which are likely to use Dedicated Short Range Communication (DSRC), to a maximum transmission range of 6Mbps (Torrent-Moreno et al., 2005), the expected communication cost is 0.81ms. Thus, we now prove that our system, according to the 1000m range distance standard (SAE, 2009a), allows recipient vehicles to react to the EWMs described in this work. For instance, driving at a speed of 110 km per hour, the average estimated vehicle stopping distance is

[5] Further simulations in embedded platforms are convenient, but the results already obtained will give an adequate idea of the relative values, and are valid to perform comparisons. Several draft implementations of the DSRC IEEE 1609.2 Draft security protocol are currently being considered but are demanding more details in terms of hardware and software specifications.

[6] We provide a characterization of the distribution of our results. The purpose of this is to show the most extreme values in the experimental data set (maximum and minimum values, which we can consider them as outermost situations or potential outliers), the lower and upper quartiles, and the median. The results are displayed as boxplots, where the box represents the interquartile range from the 25th to the 75th percentiles, and whiskers extend until the minimum and maximum of all the simulation data. Around the median, we have an equal number of values above and below, forming the interquartile range upon which we can consider an associated probability of occurrence.

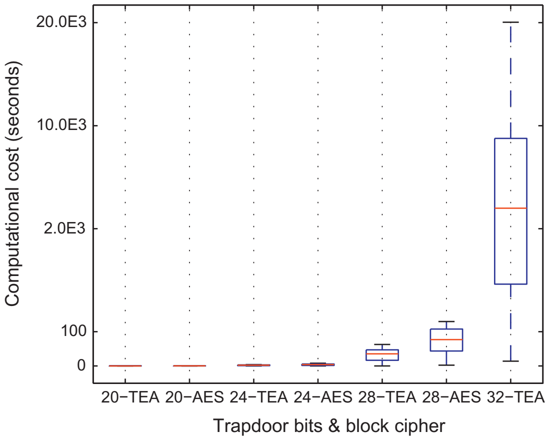


Fig. 4. Comparison of the computational effort spent (in 1000 experiments) by AES-CCM and TEA with 128 key-bits and randomly generated tokens, in terms of the amount of revealed information of the encryption key. The computational cost is expressed by the time (seconds, on a logarithmic scale) in reaching the entire key.

Table 1
Size (in bytes) for the EWC attributes.

EWC fields	Size (bytes)
Serial number	2 B
Public key certificate	100 B
Hash	32 B
Movement	16 B
POW (+K_S)	16 B (+16 B)
Validity	4 B
Crypto. algorithms	16 B
Signature (ECDSA)	40 B
Total	242 B

$\frac{110^2}{170} = 71.2$ m (Zhuang et al., 2011). At the same speed, the distance covered during the transmission time (0.81 ms) and the EWM verification time at receivers (0.16 ms) is 0.03 m, so receivers located 71.23 m farther from the sender will be able to stop or notice the event safely.

4.4. Point-of-collapse

For further analysis, we examine the feasibility of our scheme in different contexts, i.e. in complicated traffic settings and with high probability of an event warning to occur. In fact, these are the main factors which dominate the performance of our scheme, together with the POW generation as we analyzed in previous sections. Now, we provide analytical measurements of the processing overhead involved in our scheme's performance by linking the message arrival load (i.e. messages to be processed such as beacons and incoming EWMs) to the message processing capacity (messages that can be actually processed by the OBU/HSM). In other words, we say that a *point-of-collapse* is reached if a message delay becomes infinite due to the time spent in queue waiting to be processed.

To this regard, we first model a node's processing overhead time in terms of the operations which consume its resources, as the time for:

(a) validating the incoming beacons of surrounding vehicles (*BI*),
(b) generating its own beacons (*BG*),
(c) verifying the incoming EWM from precedent vehicles (*EV*),
(d) generating EWM when encountering events susceptible of warning (*EG*), and
(e) forwarding incoming EWMs when valid (*EF*).

Thus, assuming that the node processing supports a single queue and a single OBU/HSM, then the node service capacity *SC* can be expressed as

$$SC = \underbrace{r_{BI} \cdot BI}_{\text{(a)}} + \underbrace{r_{BG} \cdot BG}_{\text{(b)}} + \underbrace{r_{EV} \cdot EV}_{\text{(c)}} + \underbrace{r_{EG} \cdot EG}_{\text{(d)}} + \underbrace{r_{EF} \cdot EF}_{\text{(e)}}$$

Table 2
Processing time (schema, of a single queue and measurements for cryptographic operations on a standard platform) for a node in computing and verifying EWMs.

Number of crypto. operations	
EG	=$POW^{weak/hard} + H + S$
EV	=$VPOW + H + 2V + PCH$
Processing time for crypto. operations	
S	2.94 ms (ECDSA sign)
H	≈*negligible* (SHA-2 → 111 MiB/s)
POW^{weak}	10 ms (POW comput. TEA and 20 bits-trapdoor)
POW^{hard}	1×10^5 ms (Idem but 28 bits-trapdoor)
VPOW	≈*negligible* (POW verif. enc_{TEA} = 27 MiB/s)
V	3.52 ms (ECDSA verification)
PCH	≈*negligible* (Plausibility Checks)
Processing time for a secure beaconing	
	(Festag et al., 2010)
BG	2.9 ms/16.9 ms (standard/embedded platf.)
BI	7.7 ms/45.4 ms
BF	≈*negligible*

where r_X denotes the frequency or rate that the message type is generated or received. Moreover, we assume that the frequency of encountering warning situations, which is generally modeled as a Poisson distribution with mean $\lambda = p$, is equal to the expected number of occurrences during a given distance e.g. 1 event every 5 km. Similarly, the traffic density also follows a Poisson distribution with mean $\lambda = k$ vehicles per square kilometer. Additionally, we assume that each vehicle will generate a signed beacon every $r_b = 100$ ms as described in Festag et al. (2010). Thus, for the sake of completeness,

$$SC = \underbrace{k \cdot r_b \cdot BI}_{\text{(a)}} + \underbrace{r_b \cdot BG}_{\text{(b)}} + \underbrace{p \cdot \bar{k} \cdot EV}_{\text{(c)}} + \underbrace{p \cdot EG}_{\text{(d)}} + \underbrace{p \cdot \bar{k} \cdot p_{valid} \cdot EF}_{\text{(e)}} \tag{2}$$

where $\bar{k}$ refers to a sample of the precedent vehicles in a population of k, and p_{valid} represents the percentage of the incoming EWMs which result in successful verifications (see table in Table 2 for a detailed description of *EV* and *EG*). Note that we do not consider the computation of t_d and t_{tx}, i.e. detection time and transmission time respectively, in our analytical study. Hence, from Eq. (2), the POW computation probability linearly grows with event warning density, similarly the verification probability degrades as both the traffic and the event warning densities diminish.

For the sake of illustration we can analytically estimate the achievable service capacity of a node generating and verifying EWCs within the reference interval given by the beaconing period. Thus, we have

$$k \cdot r_b \cdot BI + r_b \cdot BG + p \cdot EG + p \cdot \bar{k} \cdot EV + p \cdot \bar{k} \cdot p_{valid} \cdot EF < 100ms \tag{3a}$$

Utilizing the measurements results from Table 2, and also assuming that the expected number of vehicles at a target area is 10 (vehicles per square km), and a fixed sample of the vehicles ahead as $\bar{k} = \frac{k}{2}$, Eq. (3a) gives us a point-of-collapse at $p = 0.4$ events in the target area within 100 ms time interval. In other words, a node is able to process up to 4 events per second in this scenario. On the other hand, if the probability of finding events is fixed e.g. 1 event every 5 km at 110 km per hour, then the number of warning occurrences within a 100ms interval is 0.0006. Hence, our scheme is applicable if the number of vehicles is less than 12.6 per square km.

Moreover, we say that a point-of-collapse is reached for different values of *p* and *k* when falling within the white region in Fig. 5. This figure shows how the beaconing process starves nodes of the service capacity when the number of surrounding vehicles is more than 13. In summary, we believe that analytical results present promising scenarios for our scheme's applicability.

5. Security analysis

We provide an informal analysis about the security of the proposed scheme. For this purpose, we discuss several attack scenarios and forms of malicious behavior which can occur concerning each step of the method execution.

The two main objectives of our scheme are to serve as a countermeasure against the indiscriminately dissemination of false event warning messages and to provide non-repudiation evidence of origin. Both goals are jointly reached by the use of digital signatures and the newly introduced POWs. Thus, senders' signature attached to event certificates prevents attackers to generate messages with false data and insert them in the channel without being detected at some time. In this context, vehicles will be asked responsibilities for their transmitted messages due to the authorities having the possibility of linking a given warning message to the corresponding sender thanks to the signature. Accountability is then effectively provided.

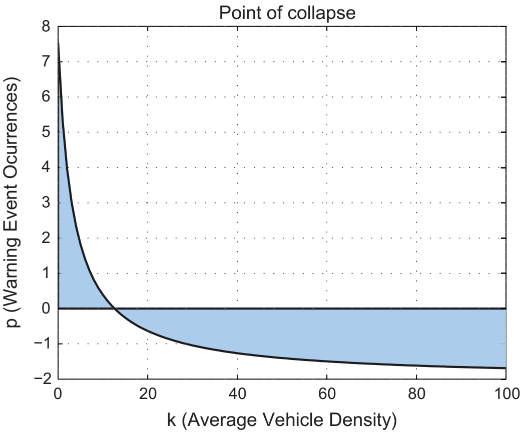


Fig. 5. Points of collapse assuming $\bar{k} = \frac{k}{2}$ and a beaconing interval of 100 ms.

For instance, the scheme is started by *A* and ultimately relies on her honesty, i.e. her on-board HSM has not been manipulated. In this case, we can assume that the original *EWM* released by her is authentic. We also assume that the cryptographic functions cannot be manipulated as well as the other fields in the event certificate. In this case, the initial event certificate, *EWC*, is correct. It is straightforward to see that any modification on the certificate performed by an attacker after its signing by the HSM or by a relaying node can be easily detected.

On the other hand, assume that an attacker can succeed in hijacking a session between any pair of participants. Even in this case, messages are safe from spoofing, because they are signed by authenticated nodes. In other words, the attacker cannot generate the correct digital signatures. Furthermore, an attacker can try to modify some messages with the hope of modifying any message or field in the event certificate. However, such a modification can be detected as it leads to an incorrect message authentication in the verification stages of our proposal.

Given enough control over the network infrastructure, an attacker can try to delete some of the messages exchanged. By doing so, the only result achieved is the unsatisfactory execution of the protocol (which can be viewed as a DoS), but it does not enable the attacker to gain any useful information. Thus, it is assumed that the infrastructure nodes placed along the road provide support for vehicles at some time.

The POW mechanisms serve as an extra deterrent to attackers due to the required computation needed to generate EWMs. We also analyze the effectiveness of using POW mechanisms to limit the indiscriminate sending of false messages and its resilience to some cheating attacks. For instance, an attacker controlling one node or a malfunctioning vehicle will be limited in the amount of event warning messages that they may send because the excessive time consumed in solving all the associated puzzles (note that each puzzle generation is unique and strictly depends on the event-related data and on some fresh information). Having said this, it is also easy to envisage that puzzles' pre-computation results unfeasible.

Moreover, suppose an attacker tries to gather a set of solved puzzles for using them thereafter. As in the previous case, there is not possibility of doing so due to unique data material related to the event is associated with the corresponding puzzle. That is, the freshness of puzzles relies on that material.

Unluckily, if an attacker controls several nodes, the POW is not as effective in limiting the number of false messages another vehicle may receive, as the attacker can then easily overcome the time and/or resource restrictions imposed by POWs to one vehicle by compromising and controlling the appropriate number of vehicles. However, we think that the inherent difficulty of compromising several vehicles (and not only one) and that the attacker still needs to control them (i.e. make them circulate in the targeted area) in order to count with correct POWs, makes this attack scenario less probable. However, even if it happens, accountability is still possible based on the proof provided by the digital signature in the EWC.

Finally, it may seem to readers of this article that honest vehicles would suffer from the same problem, i.e. every vehicle is required to perform a POW. By now, good behavior is not rewarded. Future research directions should focus on that, particularly by parameterizing the hardness of the required POW.

6. Related work

In Golle et al. (2004), authors propose that the vehicle evaluates the received data taking into account what has been obtained from its mounted sensors. Our work is complementary to that one, as we are currently focused on decreasing the chance of creating false messages, but we are not offering a total protection against this threat.

Social approaches have also been applied on this area. In particular, voting schemes have been studied in Ostermaier et al. (2007). However, their main problem is how to set the correct minimum threshold of voters that are required to rely on a given message (Daza et al., 2008). As the number of surrounding vehicles is greatly variable, no static value can be defined

(Wu et al., 2010). Moreover, though an honest majority is assumed and the threshold value can adaptively change in light of the message context (or vehicle density), malicious voters collusion is still possible, and so other protecting mechanisms (as it is the case of ours) should be employed as well.

An infrastructure for gaining trust in context information has been proposed in Fransen et al. (2006). In this work, two ways of establishing trust are identified, say by the sensor trust or by the sensorial data trust itself. Sensor trust can be achieved through five different mechanisms: because of a pre-established trust confidence, by a TTP, based on a web of trust, based on previous experiences and based on the sensor repudiation as seen by other entities. On the other hand, the sensorial data trust can be achieved by repetition (i.e. if the same message is received from different sources) or by plausibility evaluations, as explained before. Our work considers the sender vehicle as the sensor in that architecture, and our focus is on building a new way to establish sensor trust. Thus, our work can be easily seen as a new mechanism incorporated to the architecture in Fransen et al. (2006).

Finally, in Eichler (2006) a ticket-based approach is proposed to encourage vehicles sending correct messages. Using those tickets, they can get some benefits like tax discounts. However, in case a vehicle does not behave correctly, isolation and revocation mechanisms must be applied (Ostermaier et al., 2007). This approach does not alleviate the problem once it appears, but it prevents it to be repeated in the future. Our approach serves, however, as both a-priori and a-posteriori countermeasure, since it is focused on limiting the amount of false messages to be created whilst, as a result, preventing the amount of false information to be spread. From our point of view, such proactive approach is beneficial when road safety is at stake.

7. Conclusions

There is a great amount of different attacks that can be performed against data dissemination in the vehicular context. Although there have been several works addressing this issue, none of them has solved it completely. In this work, we have focused on minimizing vehicles sending false information. For this purpose, we have proposed that the sender must perform a non-negligible amount of computation prior to sending the warning message. This computation cost (known as *Proof-Of-Work*, POW) is intended to discourage dishonest sender vehicles as the required computational task consumes a non-negligible amount of the senders' resources. Now, warning messages will include a new structure called Event Warning Certificate (EWC) that contains the event data along with a non-repudiable proof attesting that the POW has been performed. Thus, EWCs are only considered as potentially useful if such proof is genuine. Our results show the suitability of the computational requirements of both the POW generation and its validation for the vehicular scenarios described.

Future research lines will be focused on extending the POW mechanism to provide a reward/punisment service by parameterizing the complexity level of POWs. This could be done by linking the POW's difficulty to the vehicle's recent trust as perceived by surrounding vehicles, for example by using a protocol such as LEAVE.

Role of the funding source

The funding source has implicitly participated in this work much in the way of encouraging our work's results to be submitted for publication in the best international journals.

In addition, our sincere thanks to the anonymous reviewers for their useful comments which assisted us in the improvement of this article.

Acknowledgment

This work is partially supported by Ministerio de Ciencia e Innovacion of Spain, Project E-SAVE, under Grant TIN2009-13461.

References

Abadi, M., Burrows, M., Manasse, M., Wobber, T., 2003. Moderately hard, memory-bound functions. In: Proc. of the 10th Annual Network and Distributed System Security Symposium, pp. 25–39.

Armknecht, F., Festag, A., Westhoff, D., Zeng, K., 2007. Cross-layer privacy enhancement and non-repudiation in vehicular communication. In: Proc. of the 4th Workshop on Mobile Ad-Hoc Networks (WMAN).

Blum, J., Eskandarian, A., 2004. The threat of intelligent collisions. IT Professional 6, 8–13.

Borisov, N., 2006. Computational puzzles as sybil defenses. In: Proc. of the 6th Int. Conf. on Peer-to-Peer Computing. IEEE, pp. 171–176.

Calandriello, G., Papadimitratos, P., Hubaux, J.P., Lioy, A., 2007. Efficient and robust pseudonymous authentication in vanet. In: Proc. of the Fourth ACM Int. workshop on Vehicular Ad Hoc Networks. ACM, pp. 19–28.

Coelho, F., 2005. Exponential Memory-Bound Functions for Proof of Work Protocols. Technical Report A/370/CRI version 2. Technical Report.

Daemen, J., Rijmen, V., 2002. The Design of Rijndael: AES The Advanced Encryption Standard. Springer-Verlag.

Daza, V., Domingo-Ferrer, J., Sebé, F., Viejo, A., 2008. Trustworthy privacy-preserving car-generated announcements in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology 58, 1876–1886.

Dwork, C., Goldberg, A., Naor, M., 2003. On memory-bound functions for fighting spam. In: Proc. of Advances in Cryptology. Springer, pp. 426–444.

Dwork, C., Naor, M., 1992. Pricing via processing or combatting junk mail. In: Proc. of the 12th Annual Int. Cryptology Conference on Advances in Cryptology. Springer-Verlag, pp. 139–147.

Eichler, S., 2006. Anonymous and authenticated data provisioning for floating car data systems. In: Proc. of the 10th IEEE Singapore Int. Conference on Communication Systems. IEEE, pp. 1–5.

EVITA, 2007. Project EVITA: E-safety Vehicle Intrusion Protected Applications. FP7-ICT-2007 of the European Community. <http://www.evita-project.org>.

Festag, A., Papadimitratos, P., Tielert, T., 2010. Design and performance of secure geocast for vehicular communication. IEEE Transactions on Vehicular Technology 59, 2456–2471.

Fransen, F., Lachmund, S., Olk, E., Bussard, L., 2006. An infrastructure for gaining trust in context information. In: Proc. of IEEE SECURECOMM Workshop on The Value of Security through Collaboration.

Golle, P., Greene, D., Staddon, J., 2004. Detecting and correcting malicious data in vanets. In: Proc. of the 1st ACM Int. Workshop on Vehicular Ad Hoc Networks. ACM New York, NY, USA, pp. 29–37.

Guette, G., Bryce, C., 2008. Using tpms to secure vehicular ad-hoc networks (vanets). In: Proc. of the Information Security Theory and Practices: Smart Devices, Convergence and Next Generation Networks. Springer, Berlin/Heidelberg, pp. 106–116.

Jakobsson, M., Juels, A., 1999. Proofs of work and bread pudding protocols. In: Proc. of the IFIP TC6/TC11 Joint Working Conference on Secure Information Networks. Kluwer, B.V, pp. 258–272.

Juels, A., Brainard, J., 1999. Client puzzles: a cryptographic defense against connection depletion attacks. In: Proc. of the Networks and Distributed Security Systems, California, USA, pp. 151–165.

Lo, N., Tsai, H., 2007. Illusion attack on vanet applications-a message plausibility problem. In: Proc. of the IEEE Globecom Workshops, pp. 1–8.

Lochert, C., Scheuermann, B., Wewetzer, C., Luebke, A., Mauve, M., 2008. Data aggregation and roadside unit placement for a vanet traffic information system. In: Proc. of the 5th ACM Int. Workshop on VehiculAr Inter-NETworking (VANET '08). ACM, pp. 58–65.

Lu, R., Lin, X., Zhu, H., Ho, P.H., Shen, X., 2008. Ecpp: efficient conditional privacy preservation protocol for secure vehicular communications. In: Proc. of the 27th Conference on Computer Communications. IEEE, pp. 1229–1237.

Naumov, V., Baumann, R., Gross, T., 2006. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In: Proc. of the 7th ACM Int. Symposium on Mobile Ad Hoc Networking and Computing. ACM, pp. 108–119.

Olariu, S., Weigle, M., 2009. Vehicular Networks from Theory to Practice. Chapman & Hall.

Ostermaier, B., Otzer, F.D., Strassberger, M., 2007. Enhancing the security of local danger warnings in vanets – a simulative analysis of voting schemes. In: Proc. of the 2nd Int. Conf. on Availability, Reliability and Security, pp. 422–431.

Papadimitratos, P., Buttyán, L., Holczer, T., Schoch, E., Freudiger, J., Raya, M., Ma, Z., Kargl, F., Kung, A., Hubaux, J.P., 2008a. Secure vehicular communication systems: design and architecture. IEEE Communications Magazine 46, 100–109.

Papadimitratos, P., Mezzour, G., Hubaux, J.P., 2008b. Certificate revocation list distribution in vehicular communication systems. In: Proc. of the Fifth ACM Int. Workshop on Vehicular Inter-Networking. ACM.

Raya, M., Hubaux, J.P., 2007. Securing vehicular ad hoc networks. Journal of Computer Security 15, 39–68.

Raya, M., Papadimitratos, P., Aad, I., Jungels, D., Hubaux, J.P., 2007. Eviction of misbehaving and faulty nodes in vehicular networks. IEEE Journal on Selected Areas in Communications, 25.

Russell, M., 2004. Tinyness: an overview of tea and related ciphers. <http://www-users.cs.york.ac.uk/matthew/TEA/TEA.html>.

SAE, I., 2009a. Draft SAE J2735. Dedicated Short Range Communications (DSRC) Message Set Dictionary rev 29.

SAE, I., 2009b. J2540-2. ITIS Phrase Lists.

Syverson, P., 1998. Weakly secret bit commitment: applications to lotteries and fair exchange. In: Proc. of the 1998 IEEE Computer Security Foundations Workshop (CSFW11). IEEE Computer Society, pp. 2–13.

Torrent-Moreno, M., Santi, P., Hartenstein, H., 2005. Fair sharing of bandwidth in vanets. In: Proc. of the 2nd ACM Int. Workshop on Vehicular Ad Hoc Networks. ACM, pp. 49–58.

Wolf, M., Weimerskirch, A., Paar, C., 2004. Security in automotive bus systems. In: Proc. of the Workshop on Embedded IT-Security in Cars (escar), pp. 1–13.

Wu, Q., Domingo-Ferrer, J., Gonzalez-Nicolas, U., 2010. Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications. IEEE Transactions on Vehicular Technology 59, 559–573.

Zhuang, Y., Pan, J., Luo, Y., Cai, L., 2011. Time and location-critical emergency message dissemination for vehicular ad-hoc networks. IEEE Journal on Selected Areas in Communications 29, 187–196.