Reputation Systems and Secure Communication in Vehicular Networks
Thesis

Author: David Antolino Rivas
Supervisor: Manel Guerrero Zapata

Ph.D. on Computer Architecture

Computer Architecture Department (DAC)
Universitat Politècnica de Catalunya - BarcelonaTech

Barcelona, March 2013

A Thesis Submitted for the Degree of Doctor of Philosophy

Abstract

This thesis presents *Chains of Trust*, a reputation system for Vehicular Ad-hoc Networks (VANETs) where users share information about Points of Interest (POIs) (restaurants, hotels, etc.), which relies on the use of asymmetric cryptography and requires no roadside infrastructure. Then it introduces *poiSim*, a new simulation tool completely developed in this thesis that will allow us to accurately simulate VANET scenarios: *poiSim* realistically simulates the interaction between almost 260,000 vehicles distributed over a map of Switzerland with a high level of detail. In addition, this thesis proposes *Anonymous Chains of Trust*, a protocol that goes one step further in the protection of user privacy by allowing users that trust each other to exchange their identities. Finally, it explores the future of VANET communication with the use of Visual Light Communication (VLC) to provide a secure link between nodes since VLC has the remarkable property of being resilient to jamming and Denial of Service (DoS) attacks.

Keywords: Security, Chains, Trust, POI, Reputation, Vehicular Ad hoc Networks, VANETs, Privacy, Certificates, Pseudonyms, Anonymity, Data Aggregation, Simulation, VLC, Visual, Light, Communication.

Preface

In the next few years Vehicular Ad-hoc Networks (VANETs) will revolutionize our driving experience, possibly to the point where our driving skills are not required anymore. Vehicles will be able to communicate with each other and with other networks, i.e., the Internet, thus laying the foundations for vehicular applications: intelligent driving systems, safety related applications, parking spot finders, peer to peer content and advertisements distribution, etc. Implementing security measures to protect users and their privacy will become of paramount importance.

A thorough review of the state of the art will reveal that most VANET applications rely on Public Key Infrastructure (PKI), which uses user certificates managed by a Certification Authority (CA) to handle security. By doing so, they constrain the ad-hoc nature of the VANET imposing a frequent connection to the CA to retrieve the Certificate Revocation List (CRL) and requiring some degree of roadside infrastructure to achieve that connection. Other solutions propose the usage of group signatures where users organize in groups and elect a group manager. The group manager will need to ensure that group members do not misbehave, i.e., do not spread false information, and if they do punish them, evict them from the group and report them to the CA; thus suffering from the same CRL retrieval problem.

In this thesis we present a fourfold contribution to improve security in VANETs. First and foremost, *Chains of Trust* describes a reputation system where users disseminate Points of Interest (POIs) information over the network while their privacy remains protected. It uses asymmetric cryptography and users are responsible for the generation of their own pair of public and private keys. There is no central entity which stores the information users input into the system; instead, that information is kept distributed among the vehicles that make up the network. On top of that, this system requires no roadside infrastructure. Precisely, our main objective with *Chains of Trust* was to show that just by relying on people's driving habits and the sporadic nature of their encounters with other drivers a successful reputation system could be built.

The second contribution of this thesis is the application simulator *poiSim*. Many's the time a new VANET application is presented and its authors back their findings using simulation results from renowned networks simulators like *ns-2*. The major issue with network simulators is that they were not designed with that purpose in mind and handling simulations with hundreds of nodes requires a massive processing power. As a result, authors run small simulations (between 50 and 100 nodes) with vehicles that move randomly in a squared area instead of using

real maps, which rend unrealistic results. We show that by building tailored application simulators we can obtain more realistic results. The application simulator *poiSim* processes a realistic mobility trace produced by a *Multi-agent Microscopic Traffic Simulator* developed at ETH Zurich, which accurately describes the mobility patterns of 259,977 vehicles over regional maps of Switzerland for 24 hours. This simulation runs on a desktop PC and lasts approximately 120 minutes.

In our third contribution we took *Chains of Trust* one step further in the protection of user privacy to develop *Anonymous Chains of Trust*. In this system users can temporarily exchange their identity with other users they trust, thus making it impossible for an attacker to know in all certainty who input a particular piece of information into the system. To the best of our knowledge, this is the first time this technique has been used in a reputation system.

Finally, in our last contribution we explore a different form of communication for VANETs. The vast majority of VANET applications rely on the IEEE 802.11p/ Wireless Access in Vehicular Environments (WAVE) standard or some other form of radio communication. This poses a security risk if we consider how vulnerable radio transmission is to intentional jamming and natural interferences: an attacker could easily block all radio communication in a certain area if his transmitter is powerful enough. Visual Light Communication (VLC), on the other hand, is resilient to jamming over a wide area because it relies on visible light to transmit information and ,unlike WAVE, it has no scalability problems. Consider a traffic jam, where vehicle density is higher than in any other situation, in WAVE vehicles will struggle to get their information across because they will all be competing for the transmission medium, whereas in VLC they will all be able to transmit continuously. In this thesis we show that VLC is a secure and valuable form of communication in VANETs, and we are the firsts to provide realistic results that back this theory.

To my parents, thank you.

Contents

List of Tables

List of Figures

Frequently Used Acronyms

VANET Vehicular Ad-hoc Network

VLC Visual Light Communication

PKI Public Key Infrastructure

CA Certification Authority

CRL Certificate Revocation List

WAVE Wireless Access in Vehicular Environments

DSRC Direct Short Range Communication

TPD Tamper Proof Device

POI Point of Interest

V2V Vehicle to Vehicle

V2I Vehicle to Infrastructure

OBU On Board Unit

RSU Road Side Unit

SDA Secure Data Aggregation

DoS Denial of Service

Chapter 1

Introduction

With the massive deployment of wireless technologies on motorized vehicles, automotive industries have opened a wide variety of possibilities for drivers and their passengers. Theoretically, anything from finding out the road conditions ahead to watching a movie through streaming is possible. Different kinds of applications will need different requirements. As mentioned in [1] and in [2] applications can be categorized as follows:

1. **Safety related:**
   (a) *Traffic information messages*: used to disseminate traffic conditions in a region and thus affect public safety only indirectly - they are not time-critical.
   (b) *General safety-related messages*: used by public safety applications such as cooperative driving and collision avoidance - they should satisfy an upper bound delay.
   (c) *Liability-related messages*: they are only exchanged in liability-related situations such as accidents - time is not an issue, but the messages should be able to reveal the senders' id to the law authorities.
2. **Others:**
   (a) *Toll applications*: electronic toll collection systems like *AutoPASS* in Norway allow drivers to continue driving without having to stop at tolls.
   (b) *TV and other multimedia content*: used to provide users with entertainment and information (movies, newspapers, etc.).

(c) *Advertisements*: businesses along the road (such as gas-stations and restaurants) could advertise themselves to drivers before they reached the businesses location, giving them enough time to compare different offers.

As far as safety applications requirements are concerned, the integrity and the non-repudiation of the messages has to be ensured, albeit maintaining at the same time the user's privacy, as will be discussed in section 2.1. Other applications, e.g., multimedia content distribution, may also need to encrypt their traffic to avoid eavesdropping from non-registered users. The use of Certification Authorities (CAs) and public key cryptography to protect Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communication fulfills most security requirements.

Architecture wise, applications can also be divided in two groups. On one hand, there are *Zero-infrastructure applications* where the only hardware requirement is the installation of On Board Units (OBUs) in the vehicles. OBUs provide the vehicles with sensing, processing and wireless communication capabilities for V2V communications, like in [3]. On the other hand, there are applications that also need Road Side Units (RSUs) to provide a V2I link, generally because they use Public Key Infrastructure (PKI) and they require access to a CA outside the network or to an Internet Service Provider ([4], [5], [6], [7], [8], [9], [10], [11]). However, with the recent development of cellular technologies like GPRS and UMTS the V2I link could by provided by the OBU itself, minimizing the dependency on road side infrastructure.

The vast majority of applications in Vehicular Ad-hoc Networks (VANETs) use PKI, because it provides confidentiality, integrity, authentication and non-repudiation and because it is a well known and reliable system. However, VANETs have their own peculiarities and if PKI does not adapt to them security issues arise. For instance, a vehicle continuously sending messages signed with the driver's private key becomes traceable, and thus the user's privacy is violated. As explained in section 2.1.5, another major issue comes from managing Certificate Revocation Lists (CRLs). CAs include revoked certificates in CRLs, which have to be distributed across the network. This poses a difficult challenge, particularly in the early stages of a VANET deployment, if the vehicles do not have permanent (or frequent enough) access to a CA. Furthermore, with millions of users in the system the potential size of the CRLs is huge.

This thesis presents *Chains of Trust*, a secure *Zero-infrastructure* reputation system focused on the distribution of Points of Interest (POIs) information. This reputation system relies on human driving patterns, i.e., the sporadic encounters

between vehicles, to transmit information. One user will trust another if they both give POIs similar reviews, and those POIs could be anything from road conditions to museums or restaurants. The main objective is to take advantage of those patterns and build a system, whose knowledge is distributed among the users' vehicles, which they can query for POIs information.

Our work is strongly focused on reputation systems because we believe that they can help people in their everyday decision making process and therefore improve their quality of life. We live in a world that produces massive amounts of information every day and in order to thrive we need to process them and make the best decisions we can. We rely on friends and family to deal with this complex problem, i.e., whether we are trying to decide where to go for dinner or making a career choice we rely on the experience of other people to help us make a good decision.

This concept lies at the foundation of reputation systems. Since it is not possible to experience everything first hand, a user of a reputation system shares his own knowledge with other system users and relies on some of them, preferably ones with a good reputation, to help him make decisions.

A user's reputation will grow with every good decision he helps others make. Naturally, people have different tastes so what may be a good recommendation for somebody may not be so good for somebody else. This leads to the creation of groups of users that trust each other because they have a similar taste, what is called a *Web of Trust*. On the other hand, entities with too different views will recognize each other as not trustable and disregard each other's recommendations. Since what is being shared is subjective information, two people may trust each other today and have different views tomorrow. In addition, they may not trust each other in one area of expertise and at the same time they may share similar views on others.

Reputation systems are increasingly being used nowadays. They are a very good way to bring some order into the chaos that can be a network of users sharing information.

They can be found almost everywhere, in P2P networks, in movie rating websites, in sites like eBay or YouTube, etc. They can be as simple as the one used by eBay -in which after each pair of users conducts a transaction they rate each other and a user's reputation is the count of positive and negative ratings- or they can be extremely complex ones.

Reputation systems, however, are vulnerable to several kinds of attacks [12, 13], one of the most serious being the breach of users privacy. By definition, in a reputation system every user has an identity to which all the opinions he makes

public can be traced to. For this reason, an attacker with the appropriate tools should be able to profile all the users in the system: knowing which restaurants they go to, the books they like, having an accurate idea of the area the users live in and even mapping their online identity to their real one.

This thesis presents *Anonymous Chains of Trust*, a solution to preserve users privacy in reputation systems. In particular, we apply this solution to the reputation system for VANETs *Chains of Trust*, although it may well apply to any reputation system. In a nutshell, users that trust each other are allowed to borrow each other identities to disseminate information over the network, thus making it impossible for an attacker to determine with all certainty who created a particular piece of information.

Finally, this thesis looks into VANET communication. Vehicular communication technologies comprise cellular (GPRS/UMTS), Direct Short Range Communication (DSRC) and the IEEE 802.11 technology family. Cellular communications can be used as a basis for long-range communications at low data rates (i.e., less than 2 Mb/s), mainly for V2I communication. Alternatively, WIFI IEEE 802.11a,b,g may provide short-range access (i.e., less than 100 m) to RSUs at medium-high data rates (i.e., between 1-54 Mb/s). Finally, Wireless Access in Vehicular Environments (WAVE) standards allow short-range communications (i.e., less than 1000 m) at data rates between 3-27 Mb/s. IEEE 802.11p WAVE, [14], is defined to allow both V2V and V2I communications. WAVE comprises IEEE 802.11p and IEEE 1609.x standards. WAVE units support multichannel operation: primary management frames and *WAVE Short Messages* (WSM) use a fixed *Control Channel* (CCH) while other management frames and data frames (e.g., IP datagrams) use a *Service Channel* (SCH). SCH exchanges require the devices to be members of the *WAVE Basic Services* (WBS) that act as the corresponding service sets in IEEE 802.11. At higher layers, the WAVE stack allows the transport of TCP/UDP using IPv6 datagrams. In this way, legacy TCP/IP connectivity is ensured. Besides, WAVE also defines a *WAVE Short Message Protocol* (WSMP) to accommodate high-priority, time-sensitive traffic. It should also be considered that the WAVE 1609.2 standard defines security services for the WAVE stack, which include confidentiality, authenticity, integrity and anonymity services.

Radio communication, however, is inherently vulnerable to jamming attacks: anyone with a powerful enough radio device can transmit in the same channel used by vehicles and distort communication over a wide area (the radius of which depends on the power of the radio device), thus causing a *Denial of Service* (DoS). The impact of such an attack ranges from a minor inconvenience for content distribution applications, like *Chains of Trust*, to a potential car accident for safety

applications. In addition, WAVE does not scale well, in high density environments vehicles have to compete for the transmission medium.

Recent research has begun to focus on Visual Light Communication (VLC) [15, 16] as an alternative form of communication. In VLC, the communication takes places between a *Light Emitting Diode* (LED) used as a transmitter and photodiode that acts as a receiver. In the past few years, there has been significant progress in this area, e.g., in [15] the authors were able to reach a transmission speed of a 100Mbps in indoor conditions. Extensive research still needs to be conducted before the technology becomes available to the general public. Efforts in that direction are backed by the recently created IEEE 802.15.7 Visible Light Communication Task Group [17] and the Visible Light Communications Consortium [18].

LED illumination is becoming widespread for indoor lightning due to its lower power consumption compared to the regular light bulb. In addition, it is also becoming increasingly popular in the automotive industry for indicator, tail and even headlights, as well as being used in traffic lights and signs. By the time VLC technology is mature enough to be used outdoors, LED illumination will be widespread and a great range of possibilities will open for VANETs.

In this thesis experiments for *Chains of Trust*, *Anonymous Chains of Trust* and VLC, we use our application simulation tool *poiSim* to process the mobility trace produced by the realistic *Multi-Agent Traffic Simulator* (MMTS) developed by K.Nagel at ETH Zurich [19]. This trace defines realistic mobility patterns for a 24 hour scenario with 259,977 vehicles distributed over regional maps of Switzerland. Our objective is to show that using a customized application simulator yields more realistic results than using a general purpose network simulator like many researchers do.

In order to study the behavior of applications in VANETs extensive research has been performed in mobility and network simulation fields. Vehicular traffic simulators can be classified in macroscopic and microscopic simulators. The macroscopic perspective considers system parameters as traffic density (number of vehicles per km per lane) or traffic flow (e.g., number of vehicles per hour crossing an intersection) to compute road capacity and the traffic distribution in the road net. In contrast, microscopic simulators determine the movement of each vehicle that participates in the road traffic.

As far as network simulators are concerned, there is a wide variety of available options: ns-2 [20, 21, 22, 23], GloMoSim [24, 25, 26, 27], OPNET [28, 29], etc. They are essential tools to simulate network aspects like communications, routing protocols and wireless propagation models. However, as far as we know, they are

not able to handle the simulation of hundreds of thousands of nodes, unlike our application simulation tool *poiSim*.

In most research articles [30, 31, 32, 33], the authors are aware of network simulators limitations and simulate a low number of vehicles (in the order of a hundred), moving randomly or following a statistical distribution. In this thesis we show that VANET applications simulation should be divided in two layers: the first will deal with network specific aspects such as the *Medium Access Control* (MAC) layer, which can be simulated by network simulators like ns-2 with a comparatively small number of nodes (in the order of a hundred) without affecting the general results, and the second will be application specific, which can be simulated by *poiSim* with a large number of nodes (in the order of hundreds of thousands) while using a realistic mobility trace. We believe that this approach will yield more accurate and realistic results than directly using a network simulator to simulate the application and the network specific behavior.

The remainder of this work is organized as follows. Section 2 gives the required background in the topics related to the thesis: security, reputation systems, network simulators, etc. Section 3 describes the problem being addressed by *Chains of Trust*, followed by a description of the simulation tool *poiSim* in section 4. Section 5 explains in detail *Anonymous Chains of Trust*, followed by our VLC proposal in section 6. Finally, the thesis closes with its conclusions and a list of references and publications.

Chapter 2

Background in Security, Information Dissemination, Reputation Systems, Vehicular Traffic Simulators, Networks Simulators and Visual Light Communication

2.1 Techniques to Achieve Privacy

In the near future, Vehicular Ad-hoc Networks (VANETs) are going to change the way people drive and it will solely depend on the security measures that are implemented if they do it for the better or for the worse. The creation of VANETs can help improve traffic management and roadside safety. Unfortunately, a VANET also comes with its own set of challenges, particularly in security and privacy. As a special implementation of mobile *ad hoc* networks, a VANET is subject to many security threats, which can lead to attacks and service abuses. For instance, an attacker could tamper with traffic applications and make its users believe there is a traffic jam in a particular road making them take an alternative way, thus freeing the original road for the attacker's benefit. A more dangerous example would be for an attacker to sign liability messages with a fake identity so that he could not be linked to a car accident scene. Furthermore, network applications could also be

used for more subtle and equally illegal objectives such as tracking people on their vehicles. Therefore, there is a real demand for security mechanisms, specially for those that protect the user's privacy.

The security architecture developed by the *Vehicle Safety Communications Consortium (VSCC)* and subsequently submitted to *IEEE P1609.2*, [34], defines a Public Key Infrastructure (PKI)-based approach for securing messages sent in Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communication. The standard, however, does not address privacy issues. In [35], the authors propose different mechanisms for certificate revocation and discuss privacy issues in vehicular networks. Conditional privacy preservation must be achieved in the sense that user-related private information, e.g., driver's name, license plate, position, etc., has to be protected, while at the same time authorities have to be able to reveal the identity of message senders in case of a traffic event dispute, such as a car accident. Therefore, it is critical to develop a suite of elaborate and carefully designed security mechanisms to achieve security and conditional privacy preservation in VANETs before they can be deployed.

Among the proposals to achieve privacy, different techniques can be identified:

- Anonymous Certificates
- Group Signatures
- Pseudonyms and Pseudonyms Certificates

Table 2.1 summarizes the privacy schemes and classifies them according to whether a scheme uses (i) anonymous certificates, (ii) group signatures or (iii) pseudonyms to achieve privacy. Table 2.1 also indicates if a work considers problems as group formation, traceability, revocation or message linkability. The *dynamic* column shows if the scheme dynamically changes the message signature keys.

Although the problem of certificate revocation is commented when needed throughout the whole section, we add at the end a specific subsection to point out other references in that field and discuss the most relevant mechanisms to reduce the size of Certificate Revocation Lists (CRLs).

2.1.1 Achieving Privacy through Anonymous Certificates

One solution to the privacy problem is to use a list of anonymous certificates for message authentication, where the relationship of the list of anonymous certificates

Table 2.1: Taxonomy of privacy and certificate revocation schemes.

	Anonymous Certificates	Group Signatures	Pseudonyms	Group Formation
[2]	X		X	
[36]			X	
[37]			X	
[38]	X			
[8]		X	X	X
[39]		X		
[40]			X	
[34]			X	
[41]			X	
[6]		X	X	
[42]	X			
[43]			X	
[44]			X	
[45]				
[46]	X			
[47]			X	
[48]				
[49]	X		X	

Table 2.2: Taxonomy of privacy and certificate revocation schemes (continued).

	Revocation	Traceability	Dynamic	Linkability
[2]	X	X	X	
[36]			X	
[37]			X	
[38]	X	X	X	
[8]			X	
[39]	X	X		
[40]	X		X	X
[34]				
[41]				
[6]	X	X	X	
[42]	X			
[43]			X	
[44]			X	
[45]	X			
[46]	X			
[47]	X			
[48]	X			
[49]	X			X

with a vehicle's driver is stored in a *Transportation Regulation Center* (TRC). For instance, in [2], the authors introduce a security protocol based on anonymous certificates. With a pool of approximately 43 800 certificates, every time a vehicle wants to communicate with the network it randomly chooses one of the available certificates to sign a particular message and then discards it. In this way, the driver's privacy is guaranteed, since there is no way for an attacker to tell if two messages were sent by the same user.

To achieve conditional traceability, a unique electronic ID is assigned to each vehicle by which the police and authorities can verify the identity of the owner in case of any dispute. Although this scheme can effectively meet the conditional privacy requirement, it is far from efficient and can hardly become a scalable and reliable approach. Since the ID management authority stores all the anonymous certificates for each vehicle in its administrative region (province or country), once a malicious node is detected, the authority has to exhaustively search in a large database (probably 43 800 certificates $\times$ millions of cars) to find the ID related to the misbehaving anonymous public key. Besides, if a node needs to be revoked all its anonymous certificates have to be included in the CRL, which will then grow very fast.

In [38] a similar solution is proposed. They also use short lived certificates, although they are blindly signed by the Certification Authority (CA). The *Escrow Authority* (EA) is responsible for maintaining the link between the anonymous certificates and the vehicle's real identity using a linkage marker, in order to deal with the "insider" attack. Still, they suffer from the same problems, because in order to revoke a vehicle all of its non-expired anonymous certificates have to be included in the CRL.

In [42] the authors devise a scheme following a very different approach from the ones described above. In a nutshell, all the nodes share a *Network Authorization Key* (AK), which grants the privilege of broadcasting messages in the VANET. In addition, every vehicle has a secret key (SK) only known by the CA and itself. Whenever a node wants to broadcast a message it needs to ask the CA for the AK, which as we will see below needs to be a short lived key. In order to enable the revocation of rogue vehicles their identifier is included in the message, although for privacy concerns it is encrypted with the CA's public key. Let us define the OBU_{id} of an anonymous node A as:

$$\{Id_A, H_{SK_A}(Id_A|H_{AK}(M))\}_{CA} \tag{2.1}$$

The OBU_{id} is added to any message A wants to broadcast to prove its authorization to transmit a broadcast message M by hashing it with the network

authorization key to produce a message digest $H_{AK}(M)$.

$$\{M, H_{AK}(M), CA, OBU_{id}\} \tag{2.2}$$

It should also be noted that the scheme relies on CRLs to revoke nodes from the network and the CA is the only one qualified to include them in the list. However, the AK is not updated until it has expired. Hence the need for a short lived AK, since nothing keeps the rogue node from broadcasting bogus messages until the AK expires (vulnerability window). On the other hand, if we consider a scheme where information messages are transmitted from On Board Units (OBUs) to Road Side Units (RSUs), validated at the CA and then issued back from the RSUs as trusted messages to the vehicles (to which they would respond diminishing speed or stopping) the vulnerability window disappears, because the CA has permanent access to the CRL and can discard any message coming from a revoked node. However, safety message applications would suffer a great delay in comparison to schemes where the information is actually collected and delivered directly by the vehicle's neighbors. Therefore, this solution is not the best suited for these kind of applications.

2.1.2 Achieving Privacy through Group Signatures

The main feature of the group signature scheme is that it provides anonymity to the group members, because any node inside the group can verify if a certain message was sent by a group member without knowing the sender's real identity inside the group.

In [39], the authors integrate the techniques of Group Signature [50] and Identity-based Signature [51] to solve the issues on security and conditional privacy preservation. They divide that problem in two parts: communication coming from an OBU and communication coming from a RSU. The main idea is to use group signatures to address the first part of the problem, so that messages are anonymously signed, while the identities of the senders can still be recovered by the authorities. In order to address the second part of the privacy problem they introduce a signature scheme that uses Identity-based Cryptography [52] to digitally sign each message sent by an RSU to ensure its authenticity.

1. **Communication from an OBU**: the main issue is how to solve the contradiction between making the messages anonymous and at the same time traceable by the authorities. A secure group signature must be correct (honestly generated signatures can be verified), anonymous and unlinkable to the

original identity although traceable under some circumstances [39]. By using a group signature scheme such as the one described in [50] a verifier can judge whether the signer belongs to a group without actually knowing the signer's real identity in the group. Besides, if the situation ever requires it, the CA, which serves as a group manager, can reveal the signer's true identity. In [39], the authors propose a role separation between the authority that provides the keys for the group and the law authorities that may need to trace a group member's real identity. Therefore, the role of the group manager is divided into a *Membership Manager* (MM), whose task is to assign private and group public keys to the vehicles, and a *Tracing Manager* (TM), i.e., the law authorities.

2. **Communication from a RSU**: messages sent from RSUs do not need to remain anonymous. Therefore, the identifier string of each RSU can be used as the public key to sign its messages. The provably-secure identity-based signature scheme described in [53] is the one chosen in [39], since the length of the signature is greatly reduced thanks to the use of bilinear pairing.

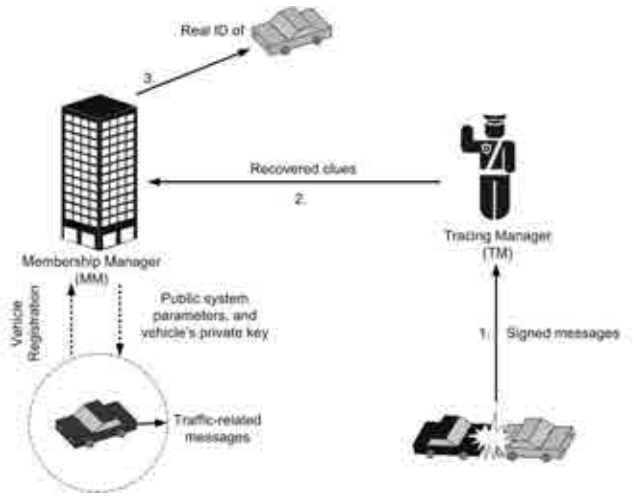


Figure 2.1: Secure Communication System

In Fig. 2.1 we can see depicted how the system works. Three types of network entities are identified: the TM, the MM and the mobile OBUs. The main idea is that all vehicles need to be registered with the MM and pre-loaded with the group public key and their own private key before they can join the network. When the

vehicles are on the road, they regularly broadcast routine traffic related messages (position, speed, etc.). Should an accident occur (or any other kind of event that required the vehicles' real identities to be revealed) police officers would submit the messages collected at the time of the accident to the TM, who is responsible for the authorization of revealing the real identities of the wanted vehicles. The TM would then forward recovered clues and evidences to the MM which would search the real identity in its membership database.

In the article, the authors emphasize the need for a system that has the ability to selectively revoke the group membership of a compromised vehicle either by updating the group keys or by releasing a customized version of the *Revocation Lists* (RLs). If the group keys are updated, the private keys of the revoked vehicles are distributed in a RL so that unrevoked vehicles can locally update their private and group public keys, whereas the revoked vehicles cannot due to the signature scheme being used (Strong Diffie Hellman in groups with a bilinear map) [54]. However, this option introduces significant overhead due to the periodic changes of keys. Alternatively, a *Verifier-Local Revocation* (VLR) scheme [55, 56, 57], similar to the traditional CRL, is very efficient (as long as the number of compromised vehicles is low) since only message verifiers are involved in the revocation check-up operation. In [39], a hybrid scheme is proposed, which in general terms consists in using VLR until the number of revoked vehicles reaches a certain threshold T and then switching to key updating.

Some aspects remain unclear in [39]. For instance, the authors do not cover how the groups are formed, or if there is communication among them, so that if a node is revoked from a group it is revoked from all groups. Besides, if VAR relies on the fact that only the verifiers deal with revoked nodes, that means that most of the group nodes are just dummy nodes (they do not interpret the message information) or even all if the verifier is the MM, which makes the whole scheme unsuitable for safety information applications. In our view, the authors should specify what VANETs applications can take advantage of their scheme.

In [8] the authors present a technique for secure group formation. Although the paper is centered on secure data aggregation it provides some insights in group formation techniques that could be used to increase privacy.

2.1.3 Achieving Privacy through Group Signatures: How Groups Are Formed

There are many ways to form groups in VANET applications. For example, all public transport buses can be members of a *preset group*. This is the easiest and most efficient way of group formation, but it requires prior knowledge of all group members, as well as a common authority over them. This is not the case when individual drivers on a highway decide to join a platoon in order to improve their driving experience. This requires *on-the-fly group* formation where a group leader is elected and group membership is managed dynamically. This latter category of groups is the most useful due to its flexibility, but it is also the most difficult to implement due to several issues, such as group leader election, group overlap, and the related security hurdles.

[8] introduces the concept of location-based groups, where the roads are divided into small area cells that define the groups. In this fashion, a vehicle will automatically know to which group it belongs, the group leader will be by definition the closest vehicle to the center of the cell and naturally, it will be elected dynamically. It should be noted that, in the leader election process, vehicles do not broadcast their real identities but rather pseudonyms for privacy purposes, so the authors combine the use of groups with the use of pseudonyms for intra-cluster privacy.

On the plus side of this proposal, the group formation process is simplified and when using geographic routing determining which groups should relay messages is straightforward. However, for an attacker to always be elected group leader will suffice to place himself in the center of the cell permanently.

Vehicles periodically broadcast their public keys, so upon the formation of the group or whenever a new vehicle A joins the group, the leader L broadcasts the group key encrypted with the node's public key followed by its signature.

$$L \rightarrow A : \{K\}_{PuK_A} Sig_{PrK_L}[\{K\}_{PuK_A}] \tag{2.3}$$

This technique leaves room for improvement if the vehicles travel together in platoon formation, since the platoon may span over more than one cell.

Also in [8] the authors propose another solution named *Dynamic Group Key Creation*. The key idea is that once the leader and members of the group are identified, the leader creates a key request message that transmits to the CA. The CA will use that information to generate an asymmetric group key pair and broadcast it to all the group members. The key pair will be encrypted with the symmetric group key included in the key request message. In addition, the CA

assigns to each group member a unique ID for non-repudiation purposes. Finally, once the asymmetric group key is established, any group member can send a message signed on behalf of the group (although accompanied by its certificate issued by the CA to allow the receivers to verify the signature). The message also includes the unique ID assigned by the CA to the group member that sent the message, which implies that the privacy of the individual vehicle is broken. Note, however, that the objective of the work reported in [8] is to reduce the overhead with *data aggregation* and does not explicitly address the problem of privacy.

2.1.4 Achieving Privacy through Pseudonyms

Pseudonymous authentication is widely accepted in the VANET community [40, 34], [6, 41], specially as an alternative to anonymous authentication, which can incur in additional overhead [2, 38].

The work reported in [40] presents a security architecture organized in layers. While the lowest layer is concerned with vehicle application registration and identification, higher layers are concerned with proper system operation, appropriate security measures and user privacy protection. In this group of higher layers we can find the *pseudonym* and the *revocation layer*.

The *pseudonym* layer provides a basic level of anonymity by introducing the possibility to use changing pseudonyms that cannot be linked by unauthorized parties. As pointed out by the authors, pseudonyms shall perform the same roles as the certificate issued for the node. This scheme uses dynamic pseudonyms to provide privacy, while at the same time an *Escrow Authority* (EA) is responsible for revoking and uncovering the user's real identity, if required.

The *revocation* layer is responsible for excluding nodes from the system. It contains a database of revoked pseudonyms and distributes this data to all nodes in the system if necessary, depending on the scale of the revocation decision, which can range from only node-local to system-wide.

We should note that when a node is revoked, all its pseudonyms are included in the revocation data. The authors do not specify how frequently pseudonyms should be changed or how large the pool of pseudonyms should be, however it is clear that there is a scalability problem.

From the system architecture perspective, the following entities are required:

- the vehicle manufacturer and the registration authority for the registration of nodes.
- the inspection site for test and certification of nodes.

- the "Escrow Authorities", entities with the authoritative power (e.g., police and courts) to identify and revoke nodes.
- the communication security infrastructure, which includes the communication systems, processing and databases necessary to carry out online testing, pseudonym provision for nodes, revocation of nodes and infrastructure based data assessment and intrusion handling.

As far as operation is concerned, vehicles use the certificate issued at the inspection site to request pseudonyms, which will be used to sign application messages. It is important to note that the scheme assumes sporadic access to the infrastructure. Some modules, such as the pseudonym provider may need reliable and on-demand connectivity, which could be provided by cellular technologies. As discussed in [35], distributing revocation information can also be achieved by simple terrestrial broadcast.

The authors in [6] go a step further and combine the use of *pseudonyms* and *group signatures*. They describe a scheme which relies on the concept of pseudonymous authentication, which they name *Baseline Pseudonyms* (BP). The novelty with respect to previous works presented in this section is that it allows on-the-fly generation of the nodes own pseudonyms using *Group Signatures*, which in combination with the BP approach they term *Hybrid Scheme*.

By BP we understand a system where each node (vehicle) V is equipped with a set of pseudonyms, that is, public keys certified by the CA without any information identifying V, where each pseudonym is used at most for a period τ and then discarded. For the i-th pseudonym K_v^i for node V, the CA provides a certificate $Cert_{CA}(K_v^i)$, which is simply a CA signature on the public key K_v^i. The private key k_v^i is used by the node to digitally sign messages. To enable message validation, the pseudonym and certificate of the signer are attached in each message. With $\sigma_{k_v^i}()$ denoting V's signature under its i-th pseudonym and m the signed message payload, the message format is:

$$m, \sigma_{k_v^i}(m), K_v^i, Cert_{CA}(K_v^i) \tag{2.4}$$

The CA maintains a map of the long-term identity of V to the K_v^i set of pseudonyms provided to a node. When required, the CA can extract the signer's identity from a message.

Assuming the general availability of the public key of the CA, upon the reception of Msg.(2.4) a node validates $Cert_{CA}(K_v^i)$. It makes use of a CRL, assumed to be distributed to vehicles via the infrastructure, as described in [58]. If K_v^i is

not included in the CRL and the CA signature on K_v^i is valid the node validates $\sigma_{k_v^i}(m)$.

The main idea behind the *Hybrid Scheme* mentioned above is that each node V is equipped with a group signing key gsk_v and a group public key gpk_{CA}. Instead of protecting messages with the group signature, a node generates its own set of pseudonyms K_v^i (and corresponding private keys k_v^i), and uses gsk_v to generate a group signature $\Sigma_{CA,V}()$ on each pseudonym K_v^i.

Basically, the nodes generate and "self-certify" K_v^i using $\Sigma_{CA,V}()$, hence producing $Cert_{CA}^H(K_v^i)$. The H denotes the Hybrid scheme differentiating it from the BP certificate and the CA subscript confirms that the certificate was generated by a legitimate node registered with the CA. Similarly to Msg.(2.4) we have:

$$m, \sigma_{k_v^i}(m), K_v^i, Cert_{CA}^H(K_v^i) \tag{2.5}$$

Upon the reception of a Msg.(2.5) the group signature is validated using the gpk_{CA} and the CRL. In this case, in order to disclose the identity of a message sender an *open* operation on the $Cert_{CA}^H(K_v^i)$ group signature is necessary ([59, 60]).

In the article, the pseudonym lifetime τ is also considered. On one hand, it makes the vehicles less traceable as it decreases. On the other, it negatively impacts on the size of *Revocation Lists* (RLs) and the revocation process performance. Varying τ from 60 down to 3 seconds the signing and verification costs are 4.6e-3 and 2.3e-3 s/msg respectively. Even though those timings may seem low at first glance, in a densely populated area with over 100 nodes within range it may be a problem for a safety messaging application, as they themselves remark.

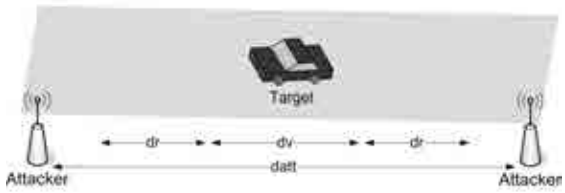


Figure 2.2: Attack scenario

In [2] the authors present an intuitive method to compute how often should an anonymous key or pseudonym be changed, adapting to the vehicle speed. Considering a tracking scenario where an attacker controls stationary base stations separated by a distance d_{att} and captures all the received safety messages. Assuming that the attacker can correlate two keys if the sender moves at a constant speed in the same direction on the same lane between two observation points.

Assuming the speed of the target V is v_t, its transmission range d_r and d_v is the distance over which a vehicle does not change its speed and lane (hence, the vulnerability window). As illustrated in Fig. 2.2, the vehicle's anonymity is vulnerable over a distance equal to $d_v + 2d_r$. Which means that it is not worth changing the key over smaller distances since an observer can correlate keys with high probability. This defines the lower bound on the key changing interval T_{key} when $d_{att} \leq d_v + 2d_r$:

$$min(T_{key}) = \frac{d_v + 2d_r}{v_t} seconds \tag{2.6}$$

However, if $d_{att} > d_v + 2d_r$, V can avoid being tracked by changing its key as long as it does not use the same key for a distance equal or longer than d_{att}. This in its turn defines the upper bound on the key changing interval:

$$max(T_{key}) = \frac{d_{att}}{v_t} seconds \tag{2.7}$$

Since V does not know d_{att}, but knows d_r and d_v, it can choose a value of T_{key} that is a slightly larger than $min(T_{key})$. If we denote by r_m the message rate, one key should be used for at most:

$$N_{msg} = \lceil r_m \times T_{key} \rceil messages \tag{2.8}$$

For instance, assume d_{att} = 2km, r_m = 3.33 msg/sec (1 message every 300 ms), d_v = 30 sec $\times v_t$ (i.e. V does not change its lane and speed over 30 sec), d_r = 10 sec $\times v_t$ (according to Direct Short Range Communication (DSRC), the transmission range is equal to the distance travelled in 10 sec at the current speed), and v_t = 100 km/h. Then $min(T_{key})$ = 50 sec and $max(T_{key})$ = 72 sec. V can choose T_{key} to be 55 seconds; as a result, N_{msg} = 184 messages.

In [43] the authors elaborate on the idea of using a pseudonym for a trip and then deriving several pseudonyms from it to use in the messages (sample identifier). They explicitly want the sample identifiers to be relatable to the trip identifiers, and at the same time different trip identifiers should also be relatable among themselves if a trip becomes interrupted by events like pauses or leaving and entering the highways with rural roads in between.

In [36, 37, 44] the authors introduce the idea of a silent period between key changes, although each one with their own particular approach.

For instance, in [44] the authors claim that in order to maximize anonymity, a moving vehicle V needs to continually observe the number of neighbors that are

communicating in its vicinity. Then, after a pseudonym update a vehicle does not actually change its pseudonym and start sending messages with it for a short fixed period of time. After that period V observes the number k of communicating neighbors and only if k is greater than a predefined threshold τ V transmits with the updated pseudonym. Otherwise, it remains silent.

The approach above is not suited for safety message applications. If the vehicles in the VANET need to periodically broadcast safety messages for cooperative navigation, then the period between those broadcasts will be the maximum time a vehicle can remain silent, which needs to be quite small (order of hundred milliseconds [36]) regardless of the number of neighbors. In [36] the authors introduce the use of a random silent period between the update of pseudonyms. They propose that vehicles form groups and that a group leader is elected. That group leader acts as a proxy for the rest of vehicles in the group for V2I communications, so that the rest of nodes in the group can remain silent for a longer periods of time. Nevertheless, they direct this scheme to *Location Based Services* (LBS)[1] and not to safety message applications.

Opposed to the use of silent periods between pseudonyms update are the *Mix-Zones* (MZs) described in [61]. Basically, in a MZ all the vehicles in a certain zone agree to change their pseudonyms at the same time, which according to the author makes any attempt to trace a certain vehicle V nearly impossible (provided that enough nodes are in that particular zone). However, this technique is also faulted for safety message applications for the very same reasons described for the previous technique.

Similarly, [40] introduces *Context Mixes*, where vehicles only change their pseudonym if they consider it is safe, i.e., they have enough neighbors.

Contrary to the widespread belief that changing pseudonyms protects vehicles privacy, in [49], the authors conclude that use of multiple pseudonyms may not be enough. Using *Multiple Hypothesis Tracking* (MHT) [62] and considering an attacker model where the attacker has the capability to capture all beacons sent to the network, they conclude that in a scenario with vehicles sending beacon messages at 1 Hz, changing their pseudonyms every 10 seconds and considering a equipment rate of 20% (rate of vehicles equipped with OBUs) an attacker can effectively track vehicles with an accuracy of almost 100%.

[1]LBS make use of the vehicle position to provide a service, for instance finding the nearest hospital

2.1.5 Achieving Privacy through PKI: Managing Certificate Revocation

PKI is a widely accepted solution [35, 39, 6, 45] as stated by the IEEE 1609 family of standards for Wireless Access in Vehicular Environments (WAVE) [63]. Vehicles in the network need the appropriate certificates in order to participate in the system operation. Nevertheless, the certificates should only be valid for limited periods of time after their generation and the CA should reserve the right to revoke any nodes' certificates, essentially evicting them from the network. In several articles, [2, 35, 64], it is accepted that vehicles will carry a trusted component or Tamper Proof Device (TPD) where the keys and certificates for network operation are stored and protected.

One of the main concerns of using PKI systems is managing the CRLs, with millions of users in the system, the potential size of the CRL is huge. In [35, 64] the authors present a way to compress CRLs using Bloom filters [65]. The main characteristic of Bloom filters is that they return a configurable rate of false positives, but there are no false negatives (if the Bloom filter claims that an element is not in the set, we can be sure it is not). A Bloom filter (Fig. 2.3) consists of

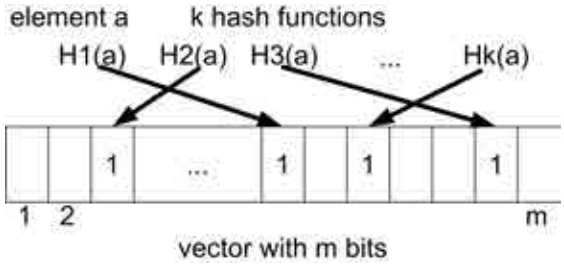


Figure 2.3: Bloom filter.

a sequence of m bits, initially all set to zero. A key or element can be included in the filter by hashing it with a specific number k of independent hash-functions (each ranging from 1 to m) and by setting to 1 the vector bits that are set to 1 in the result. After having added several keys to the filter, it is certainly possible that one bit is set to 1 multiple times. To check if an element is contained in the filter, the element is hashed and the status of the corresponding bits are checked. If at least one bit that should be one is not, one can surely affirm that the element is not contained in the filter. On the other hand, if all necessary bits equal 1, with high probability the element is included. However, it may also be possible that the bits were set to 1 by a combination of several other keys, as explained

before. Therefore, the more elements added to the set, the larger the probability of false positives. Alternatively, in [45] the authors take advantage of the multi-tier (regional) CAs set-up to decrease the size of the CRLs. Regional CAs will only manage the certificates of vehicles in their region.

Authors in [46] propose a scheme based on *Temporary Anonymous Certified Keys* (TACK), used to authenticate messages sent by the vehicles, whose CRL size is linear in terms of the number of revoked vehicles and unrelated to the size of the vehicle anonymous certificate set. There are three main entities:

- M: managing authority that acts as the root of trust.
- R: set of valid *Regional Authorities* (RA). RAs act as intermediary authorities and can grant vehicles temporary region-specific certificates. M issues certificates to RAs and certifies them to be valid intermediary authorities.
- V: set of valid vehicles or OBUs. Any vehicle with a valid certificate from M or a region-specific short-lived certificate from R (while in the proper region) is considered part of V.
- $\neg V$: set of expired or revoked vehicles.

The main idea is to apply group signatures considering a group which comprises all of the above described entities. M is defined as the group manager. It initializes the group signature scheme to generate a group public key gpk and a group master key gmk. It publishes gpk and retains gmk for itself. Each valid OBU has a group user key guk_i, issued by M, which is installed during annual vehicle inspections. It should be noted that M maintains a history of all key/OBU pairs it has issued, so that it can later trace misbehaving vehicles. When a vehicle enters a new region it needs to update its TACK following these steps:

1. Randomly select new short-lived public and private keys from the key space (K_S^+, K_S^-).
2. Use the group user key guk_i to sign K_S^+ and send it to the RA.
3. RA verifies that the user is not in the RL. If it is not, the RA signs a certificate for the OBU's TACK public key K_S^+ using the RA's secret signing key K_{RA}^{-1}.
4. RA waits for δ seconds to queue up all certificate requests for that region and broadcasts the certificates.

Whenever a user wants to send a message it signs it with its TACK private key K_S^{-1} and periodically broadcasts the RA signed certificate of its TACK public key K_S^{-1}. Whenever a user misbehaves, to determine which OBU generated a signature ψ the group manager tests ψ against the group user keys of OBUs in V. Once M identifies V_i it is added to the RL and distributed to the RAs.

Similarly, in [47] the authors try to achieve the same small CRL size with a pseudonymous authentication scheme. The network architecture is composed by a *Trusted Authority* (TA), RSUs and vehicles or OBUs. The TA issues a certificate $Cert_{TA,R_x}$ for a certain RSU R_x, and a series of pseudonymous certificates for a vehicle V_i to be installed during periodic vehicle inspections. It should be noted that the identities in the pseudonyms certificates are derived from two random seeds using a one-way hash function. The TA divides the maximum time between vehicle inspections into time windows. For every window, the TA chooses a random secret key to sign the vehicle's pseudonymous certificates, so that in every window the vehicle has to request R_x to re-sign the pseudonymous certificate for that window. In this scenario, a RSU can be revoked by including its only certificate in a CRL. To revoke a vehicle it would suffice for the TA to release the random seeds from which V_i's pseudonymous identities are computed, so that the RSUs do not issue the re-signature key to V_i in following windows. At the same time the valid pseudonymous certificate of V_i should be revoked.

In [48], the authors define *Most Pieces Broadcast* (MPB) technique to distribute CRLs. The first step is to break the large CRL file down into small pieces, taking into consideration the coding rate (rate of pieces generated from a file) and the code overhead (number of pieces needed to recover the original file). MPB ensures that only the node with the largest number of pieces broadcasts in a certain area to maximize the use of the wireless channel. It should be noted that RSUs will always be selected as the node with most pieces. The authors show that MPB is more effective than letting all OBUs broadcast their CRL pieces without control, which results in a broadcast storm of unneeded CRL pieces that slows down the CRL distribution.

2.2 Detection and Eviction of Misbehaving and Faulty Nodes

In the previous section we have focused on schemes that provide a secure and reliable network and try to keep attackers from disrupting its normal operation.

However, due to the attackers ability or just to the devices aging process at some point in time there will be misbehaving or faulty nodes in the VANET. That is why in this section we outline several techniques to detect and evict them from the network.

Table 2.3: Taxonomy of misbehavior protection schemes.

	Tamper Proof Device	Requires Certification Authority	Honest Majority	Sybil Attack Protection
[66]			X	X
[67]		X	X	X
[64]	X	X	X	
[9]		X	X	
[68]				X

In [66] the authors develop an heuristic called *adversarial parsimony*, which informally means finding the best explanation for corrupted data. The first step is to enhance the vehicles sensing capabilities giving them physical means to distinguish its neighbors, for instance with cameras or exchanging information in the infra-red light spectrum to verify that a vehicle is where it claims to be, thus preventing sybil attacks. That information needs to be exchanged between vehicles, and once enough evidence has been collected the heuristic will find inconsistencies, if any. For instance, if there is a group of nodes that are linked to the rest of the network by only one node then that link node is probably impersonating all the others.

In [67] the authors present a solution to reliably detect sybil attacks based on radio signal strength analysis and on the fact that a vehicle cannot be on different places at the same time. For clarity of description, they define three categories or roles:

1. **Claimer**: each node periodically broadcasts a beacon message at beacon intervals t_b for the purpose of neighbor discovery. In the beacon message, it claims its identity and position. The goal of the scheme is to verify its claimed position.

2. **Witness**: all neighboring nodes, within the signal range of the claimer, would receive the previous beacon message. They measure the signal strength and save the corresponding neighbor information in their memory. Next time they broadcast a beacon message, they will attach their neighbor list including the signal strength measurements.

3. **Verifier**: after receiving a beacon message, a node waits for a verifying interval t_v during which it collects enough signal strength measurements concerning the previous beacon message from neighboring witnesses. With the collected measurements, the node can locally compute an estimated position for the claimer, for instance, by performing *Minimum Mean-Square Error* (MMSE). However, to be as accurate as possible, before actually making the computations to locate the sender of a message the node needs to discard all the signal strength information that comes from sybil nodes.

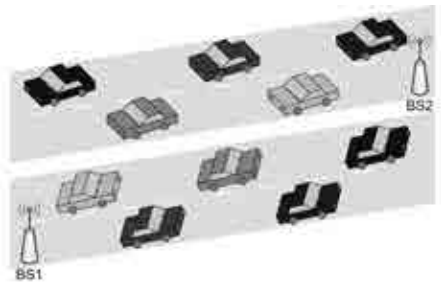


Figure 2.4: A scenario with roadside infrastructures.

In order to discard sybil nodes information they rely on two principles or rules.

1. **Rule 1:** a RSU or *Base Station* (BS) issues a position certification for each vehicle passing by. The position certification contains a time stamp and a location information of the BS and therefore can prove the presence of the vehicle near the base station at a certain time.

2. **Rule 2:** all witnesses for a claimer should consist of vehicles in the opposite traffic flow.

With Rule 1, we can ensure where a certain vehicle comes from. Looking at Fig. 2.4 node a can get a position certification from *BS2*, when passing by *BS2*, and node b also get one from *BS1*. When a and b meet each other, it is easy for them to prove that they come from opposite directions by exchanging certificates. With rule 2, we can ensure that each witness in the opposite traffic flow is a physical vehicle instead of a sybil one. For instance, suppose that a malicious node m fabricates seven sybil nodes, in which s_7 is traveling in the opposite direction and the rest in the same. When trying to verify the positions of $s_1, ...s_6$, node s_7 would be ignored because it cannot prove that it comes from the upstream of the road.

On the whole, with the help of roadside infrastructure, dishonest sybil nodes can be detected through position verification.

In [64] the authors rely on the vehicle's TPD to execute their protocol and even revoke itself if it detects it has been tampered with. They also assume the existence of a honest majority in the attacker's neighborhood. Unfortunately, TPDs usually end up becoming just Tampered Devices, as shown in [69, 70, 71]. Therefore, an attacker could just modify their programming to impersonate several vehicles (Sybil attack) [72], rendering the honest majority hypothesis invalid. And even if the TPD remained tamper-proof nothing can stop an attacker from actually stealing the physical device from another car and once again mount a Sybil attack. Nevertheless, the authors devise a *Misbehavior Detection System* (MDS) as well as a *Local Eviction of Attackers by Voting Evaluators* (LEAVE) protocol to detect and exclude misbehaving nodes.

MDS basically consists in each node using its own sensory inputs, messages received from its neighbors and a set of evaluation rules to classify the received safety messages from a given node as faulty or correct. Messages that are outdated, received beyond their theoretical area of propagation or contradictory to the node's own state are considered false. Their senders, as long as they are neighbors of the node running MDS are tagged as misbehaving and their identity is passed on to LEAVE.

The main principle of LEAVE is simple: the neighbors of a misbehaving vehicle should temporarily evict it. It should be noted that the system does not require a permanent connection to the CA to work, as we will see below. It is not a revocation protocol, but rather a collective warning system against misbehaving nodes. Upon detecting an attacker, vehicles broadcast warning messages to all vehicles in range, so that the sharing of information improves the effectiveness of the standalone detection system. Besides, those warnings can be very valuable when vehicles receive them even before being able to observe the misbehaving node themselves. Any vehicle receiving a warning message adds the warned device to an accusation list, and once enough warning votes against a node are collected, its identifier is added to a local blacklist. After entering the blacklist, *disregard* messages are repeatedly broadcasted to the local neighborhood to ignore the attacker's messages. The eviction is temporarily limited to the duration of the contact between the attacker and its neighbors running LEAVE. However, once the connection to the CA is re-established a global-scale revocation protocol can be initiated.

In [9] the authors devise another scheme based on *suicide attacks* ([73]) called Stinger, which also relies on a honest majority. In a nutshell, should a node believe another one has misbehaved it will send a message that will evict them both from

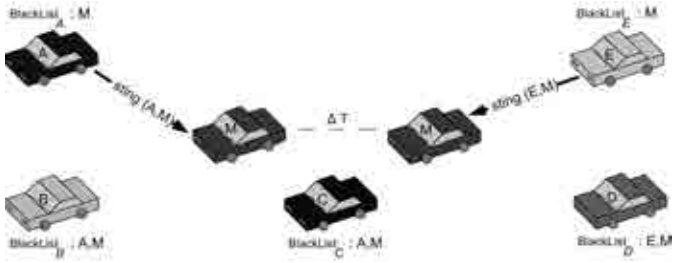


Figure 2.5: Multiple stings for misbehaving node M as it moves over time.

the network. The idea is to make the sacrifice of future participation so costly that discourages false accusations. Stinger deviates from a suicide attack in the following aspects:

1. **Stinger temporarily prohibits devices from transmitting messages, but allows them to continue to receive and forward messages**. Temporary removal could be used to rapidly ignore an errant transmitter. The authors assume that most interactions are short-lived and therefore temporary removal is as effective as permanent removal in tackling misbehavior. While the sting instruction prevents the bad and the good device from sending out additional warnings, both will still receive safety instructions from other cars. The authors claim that this solution minimizes the noticeable impact on the sacrificing vehicle while still penalizing a malicious device. However, in our view, when considering safety message applications the noticeable impact is indeed noticeable since the accusing nodes will not be able to send the information collected by their own sensors.

2. **Stinger does not allow more than one node to sacrifice itself for a misbehaving one (in a local context).** Fig. 2.5 illustrates how the protocol works as the cars move. Misbehaving node M is detected by A, which broadcasts $sting_{A,M}$ to indicate vehicles near A to ignore M. Hence, nodes B and C add both A and M to their local blacklists, while D and E do not because they did not receive the sting message. As M moves into range of D and E, E issues a new removal for M, $sting_{E,M}$. D adds E and M to its local blacklist, but C does not because it has already ignored M from A's sting.

3. **Stinger permits good devices to continue to accuse bad ones even after having issued one sting.** The authors claim this last condition to be

necessary to prevent the so-called *motorway attacker* who widely broadcasts misbehavior and moves around quickly to attract many stings and prevent good nodes from excluding subsequent attackers. However, we think the *motorway attack* is still possible just by doing the exact opposite. Since good nodes are always allowed to accuse misbehaving ones, it would suffice for the attacker to move around accusing good nodes instead of misbehaving himself. By using scheme like MDS (described at the beginning of the section) which had only local visibility (a node only gathers information about its neighbors), there would be no possible way for a group of nodes that encountered the attacker for the first time to identify him as such because of something he had done in the previous group. This could be solved by a misbehavior detection system which had a global vision on all the groups in the network.

[68] presents a system, which just like MDS, uses its sensory input to detect misbehavior. After receiving an alert message, a vehicle V compares the sensed behavior of the surrounding cars with a model of expected behavior under that kind of alert and analyzes how it deviates. For example, if the vehicles ahead of V start slowing down after he has received an alert message claiming there has been accident that will match the expected behavior. This kind of techniques could help an OBU determine whether alert messages are true or not, but they require fine tuned models of expected behaviors for each of the possible alerts. Something we believe to be unfeasible given the large number of possible alert situations.

2.3 Techniques for Secure Data Aggregation

One way to use available bandwidth more efficiently is to aggregate the information of several vehicles into a single message or record, as done in the V2V traffic information system described in [74], where vehicles share information about each other. Data aggregation shall be able to aggregate events according to temporal and spatial dimensions. Moreover, filtering old reports is an essential part of any aggregation scheme. Thus, any aggregated record has to include an expiration time after which the information is no longer valid. More difficult is the definition of spatiality. In terms of aggregation, the key question is how far a primary record (i.e., an original record) can participate in an aggregation process.

Authors in [75] prove that any successful aggregation scheme must reduce the bandwidth at which information about an area at distance d is provided to the cars asymptotically faster than d^2. In their scheme, data aggregation is originated

at *measurement points*, [75], and goes to *destinations* (i.e., set of vehicles that are interested in information from *a measurement point*). Many data aggregation schemes consider *measurement points* as specific areas that can be fixed (e.g., a road segment) or dynamic (e.g., based on the location of a set of vehicles). Other schemes consider groups of vehicles called *clusters* with a specific vehicle, *the cluster-head*, in charge of aggregating the primary reports. Clusters can be organized based on their fixed geographical area or can be dynamically formed by mobile vehicles. Furthermore, according to [7] data aggregation in VANETs can be classified as syntactic and semantic. *Syntactic aggregation* compresses data from multiple vehicles in order to fit the data in an unique record or frame. For example, an application that extracts a subset of each individual record and adds it to a single record is reducing the original information. *Semantic aggregation* means that the data from individual vehicles is summarized. For example, an application that instead of sending the location of each vehicle, only reports the number of vehicles in a given area.

Aggregation however, aggravates the security problem. A malicious aggregator may send aggregated records that do not correspond to real data. For instance, it may falsely report a congested road by pretending to have aggregated more records than it has actually received from cars ahead of it. Secure Data Aggregation (SDA) aims to ensure the integrity of the data aggregation mechanisms in the presence of malicious nodes that can alter the result of the aggregation. Forging or suppressing a single record can have low impact in both syntactic and semantic aggregation. Thus, the main threat, [8], in SDA is the generation of false aggregation information. Secure data aggregation is a topic well studied in sensor networks. However, due to the mobility nature of vehicular ad hoc networks and the fact that nodes move following specific paths, the re-use of wireless sensor network SDA mechanisms is not possible in VANETs.

Authors in [76] propose the following generic aggregate structure for SDA schemes:

$$A = [\underbrace{(a_1, b_1), ..., (a_n, b_n)}_{index-dimensions} | \underbrace{(v_1, ..., v_p)}_{values} | \underbrace{(m_1, ..., m_p)}_{meta-inform.}] \tag{2.9}$$

where the index dimensions indicate the area and time about which an aggregate contains information. The values are the information and the meta-information contains that additional information used in security mechanisms. In general, most of the SDA proposals found in the literature follow similar structures, although there is not a consensus in a well defined aggregated structure.

Table 2.4 summarizes the SDA schemes covered in this section and classifies them according to whether a scheme (i) performs syntactic or semantic aggregation, (ii) is cluster-based (cluster-head responsible for aggregating reports and (iii) is defined for fixed or dynamic geographical areas.

Table 2.4: Taxonomy of Secure Data Aggregation (SDA) schemes.

	Syntactic	Semantic	Cluster based	Fixed/Dynamic Areas
[7]	X	X		D
[8]		X	X	F
[77]		X		F
[78]	X		X	F
[79]	X			F
[76]	X			D
[80]		X		D

The authors of [7] propose a technique to probabilistically detect malicious vehicles that generate false aggregated information. In particular, they focus on validating speed and location information using syntactic aggregation although their solution is also applicable to certain cases of semantic aggregation. The proposal targets aggregated information from a measurement point to a destination, without the need of creating groups or clusters of vehicles. The main idea behind this scheme is to challenge the aggregator to provide a proof that can be used to probabilistically validate the aggregated record. An aggregated record is created by combining and compressing information contained inside several individual records. To validate the aggregated record the aggregator is asked to provide a randomly-chosen original signed record (whose information was included in the aggregated record) after the aggregated record has been sent. If the corresponding record was made up it will not be possible for the aggregator to produce the original signed record, and he will be caught. It should be noted that the probability of a misbehaving node being caught is directly proportional to the amount of bogus information it includes in the aggregated record and that for the system to work the penalty needs to be severe enough to discourage misbehavior (e.g. permanent eviction from the network).

In order to avoid a two-phase protocol, vehicles are equipped with a TPD which acts as a proxy for the receiver. As a proxy, it first provides a transmit buffer (data placed on this buffer cannot be tampered with and will be transmitted) and second it challenges the application (aggregator) to provide a randomly chosen original

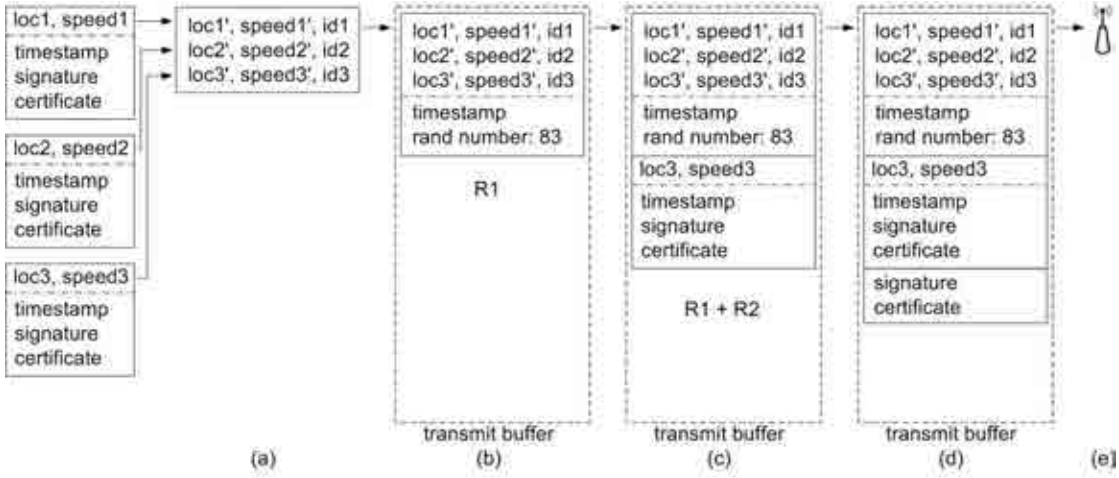


Figure 2.6: Secure aggregation using the TPD as a proxy for the receiver.

signed record to be sent with the aggregated data. The whole process can be observed in Fig. 2.6. The application extracts the data and car ID. from each regular record (a) and places it in the transmit buffer where the TPD appends a secure time-stamp and the randomly generated number 83 (b). The application takes the regular record corresponding to entry i=(83 mod 3)=2 (i.e., the third entry) (c) and appends it to the transmit buffer. Finally, the TPD signs R1+R2 (d) and broadcasts the contents of the transmit buffer (e).

Even though this method may indeed prove itself to be effective against malicious aggregators who try to insert false information in the network, it leaves the vehicles unprotected from malicious aggregators that leave out information from the aggregated records. In our view, the TPD could also serve as the entry point for received records and it should keep track that the vehicle identities in the received messages at some point before an upper bound τ are included in an aggregated message to be broadcasted.

The authors of [8] claim that bandwidth efficiency can be achieved using combined signature techniques. The authors address secure group formation, where each group is composed by those vehicles in a specified geographical area or *cell*. The group leader is chosen as that one closest to the center of the cell. Thus, the group leader is in charge of aggregating and disseminating data. Group leaders receive signed reports from vehicles creating a new message with a combined signature. Therefore, combined signatures is a semantic SDA mechanism since there is only one message m signed by the combination of all vehicles that participate

in the event detection. The following combined signatures are proposed:

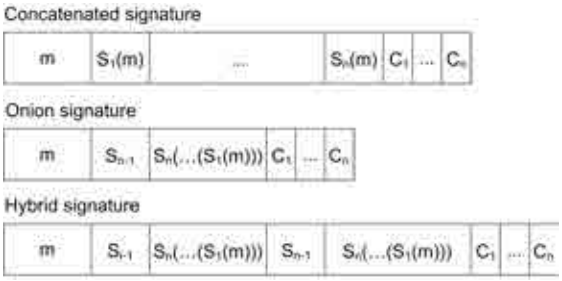


Figure 2.7: Three different types of combined signatures. n is the total number of signers. C_i is the certificate of i-th user

1. **Concatenated Signatures:** the idea behind this scheme is that whenever a vehicle receives a message if it agrees with the message information (based on its own sensors input) it appends its signature. This form of source aggregation results in a smaller data verification delay than destination aggregations where the receiver collects messages from different sources and then cross-checks them. Another advantage is that an invalid signature does not affect the whole message, in contrast to the next scheme.

2. **Onion Signatures:** the signature sizes are constant, since each message is hashed before being signed. Instead of simply appending a new signature, a vehicle signs the signature of the previous transmitter, although before retransmitting the new message, it should also include the last signature, i.e., the one it received, so that the vehicle at the next hope can verify the previous signature. The improvement in signature size comes at a cost. In this case, a single invalid signature will affect the whole message and the message needs to be verified at each hop, increasing the overall verification time. In our view, this last feature if correctly exploited could lead to a denial of service attack.

3. **Hybrid Signatures:** consists of several concatenated onion signatures, each of a given depth. The signature depth representing the number of layers it includes. This solution looks for a compromise between the previous two, both on their advantages and their drawbacks.

We find that *Hybrid Signatures* are a very interesting possibility to explore when considering safety message applications. We propose that the different kinds

of safety messages of the application be assigned a degree of time criticality and needed trust and depending on those values the appropriate depth of the *Hybrid Signature* be chosen. For example, if a vehicle is on a crossing with no visibility on the right side of the road we can safely assume its driver will not mind waiting a few seconds before it can safely traverse. Therefore, in that case the better suited solution would probably be a *Hybrid Signature* with depth 0.

In Catch-up [77], the authors propose an aggregation scheme for applications where a delay of tens of seconds is acceptable, not suited for safety messaging applications but perfectly valid for general traffic information. Aggregation is performed in road sections for the same frame interval of time. Their objective is to perform semantic aggregation by generating a single secure report with aggregation functions such as MAX, MIN, AVG. Any vehicle can aggregate the data and thus there is not any cluster structure created. The basic idea in this scheme is to insert a delay before forwarding a report to the next hop. However, their scheme makes this delay more controllable in order to increase the probability that a report can be merged with reports ahead or reports behind. Intelligent delay control policies are made based on local observations of individual vehicles. They also design a future reward model to define the benefits of different delay-control policies, and then establish a decision tree to help a vehicle choose an optimal policy from the perspective of long-term rewards.

CASCADE, [78], is a cluster-based syntactic SDA scheme. Each vehicle presents location information based on its difference from the location of the cluster's center and its speed based on its difference with the median speed of those vehicles in the cluster. The primary record is signed by the vehicle using ECDSA and includes a timestamp to prevent replay attacks and the vehicle's public key. Each vehicle, then builds its own local view from primary records. Records are grouped based on their distance from the receiving vehicle. First each data record is compressed using differential encoding. Second, an aggregated cluster record is built which is the concatenation of compact data records (syntactic aggregation). The signature is calculated by the aggregating vehicle over all fields of the aggregated frame except the certificate which is signed by the CA and the sender's location that represents the last location of the last vehicle that broadcasted the record.

The authors in [79] argue against fixed segmentation of roads because it contradicts the real situation. They propose a completely structure-free aggregation mechanism, which enables to aggregate data purely based on their correlation. On a conceptual level, all aggregation systems have the following basic components:

- Decision criteria: decide if two pieces of information are similar enough to

be aggregated.

- Information fusion: once the decision to aggregate two data items has been reached, a defined method is require to combine them.

- Dissemination mechanism: having aggregated to data items, the new information is only available to the aggregator. Thus, the node needs to disseminate the new data into the network.

The authors propose a fuzzy logic scheme to be used for the decision criteria, which allows a dynamic fragmentation of the road. First, all influences on the aggregation decision, i.e. location difference of two aggregates or a maximum standard tolerable deviation of the average speed values, are fuzzyfied by applying fuzzy set theory. Then, they use fuzzy logic operations to reason about the influences and reach a decision.

In [76], the authors present a syntactic SDA scheme. The mechanism chooses a subset of all atomic primary reports to generate an aggregate report. The authors employ a list of criteria to selectively choose which primary reports contribute to the aggregate report. The criteria includes the identification of those primary reports that led to an aggregate current maximum and minimum in time and space, defining a specific location area. The scheme is a cluster-based mechanism where the cluster borders are defined by the location of a subset of primary reports and those reports corresponding to vehicles inside the borders of the area will be selected to produce an evenly distribution that represents the whole area.

In [80], the authors introduce the concept of soft-state sketches for probabilistic hierarchical data aggregation, which derive from Flajolet-Martin sketches (FM sketches) defined in [81]. A FM sketch is a data structure for probabilistic counting of distinct elements. It represents an approximation of a positive integer by a bit field $S = s_1, ..., s_w$ of length $w \geq 1$. The bit field is initialized to zero at all positions. To add an element x to the sketch (an observation), it is hashed by a hash function h with geometrically distributed positive integer output, where $P(h(x) = i) = 2^{-i}$. The entry $s_{h(x)}$ is then set to one. In soft-state sketches, the authors use small counters of n bits instead of single bits at each index position. These counters represent a time to live (TTL) for a certain bit. Therefore, the operation of setting a bit to one after an observation is replaced by setting the corresponding counter to the maximum TTL. An approximation $C(S)$ of the number of distinct elements added to the sketch can be obtained from the length of the initial, uninterrupted sequence of ones, given by

$$Z(S) := min(i \in \mathbb{N}_0 | i < w \wedge s_{i+1} = 0 \cup \{w\}) \tag{2.10}$$

by calculating

$$C(S_1, ..., S_m) := m \cdot \frac{2^{\sum_{i=1}^{m} Z(S_i)/m} - 2^{-K \cdot \sum_{i=1}^{m} Z(S_i)/m}}{\varphi} \tag{2.11}$$

with $\varphi \approx 0.77351$, $K \approx 1{,}75$ and using m sketches.

Locally stored sketches are periodically broadcasted to the vehicle's one-hop neighbors, which upon reception merges them with its own. For example, consider an application where the number of free parking spots on a road segment is disseminated in the network. Two cars, A and B, make independent observations on the same road segment (with ID 7). A observes four free parking places and thus hashes the tuples (7,1), ..., (7,4) into its sketch for road 7. B observes five free parking places, and consequently adds (7,1), ..., (7,5). If A and B meet they will exchange sketches, as depicted in Fig. 2.8 and perform a position-wise maximum operation. Previously inserted elements die out after their TTL has expired, unless they are refreshed by a newer observation.

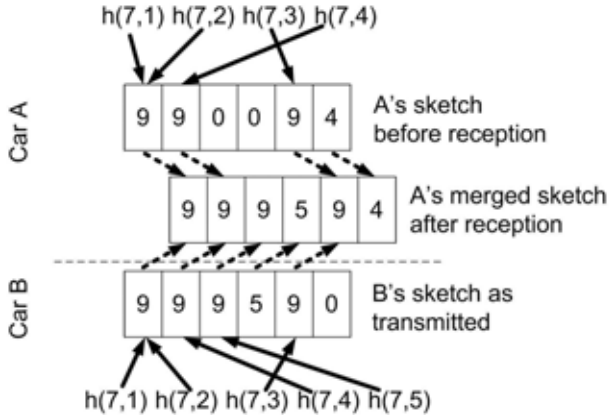


Figure 2.8: Aggregation of soft-state sketches

2.4 Information Dissemination Techniques

This thesis introduces an information dissemination technique, which to the best of our knowledge is the first one to build a reputation scheme using user signatures to distribute Points of Interest (POIs) information in a VANET.

Nevertheless, there are other works that consider the distribution of content in VANETs. For instance, in [82] the authors describe a protocol for the distribution of advertisements. They propose a virtual cash scheme where the following actors are involved:

- *CA*: every vehicle is loaded with a pair of keys (public and private) issued by a CA and with the CA's public key.
- *Vehicular Authority*: entity that approves every advertisement to be loaded in an *Ad Distribution Point*.
- *Ad Distribution Point*: broadcasts advertisements to the vehicles passing by.
- *Virtual Cashiers*: users are rewarded with virtual cash for forwarding advertisements. They sign each other receipts to prove the message forwarding. Later on, that cash can be exchanged for other services at the *Cashiers*.
- *RSUs*: provide a link to the CA for keys revocation purposes.

In [83] the authors present Roadcast, a popularity aware P2P content sharing scheme. Their technique relies on the idea that by ensuring that popular data is widely shared with other vehicles the overall query delay can be improved. If users request popular data, which is densely disseminated in the network, their queries can be answered in much shorter time than a request for rare data, because the chance of meeting another vehicle with that particular piece of information is much higher. In the opportunistic and unreliable VANET, the authors expect users to be more willing to receive data which approximately matches their request with a short delay than waiting for a longer time to receive exactly what they requested. Thus the need to forward the popular information with higher priority.

Chains of Trust distinguishes itself from the other solutions presented in this section by using broadcast in a completely ad-hoc network as a mean of information dissemination, by not using a CA and therefore not having to deal with the distribution of CRLs and by performing data aggregation through the use of concatenated signatures.

2.5 Reputation Systems

In this section we intend to analyze the current state of the art in reputation systems and point out the different schemes benefits and drawbacks.

The authors in [30] propose a reputation system to manage traffic warning events while preventing the spread of false messages. In their proposal, users/vehicles are divided into different categories according to their proximity to a traffic event and play different roles: (i) *Event Reporter* (ER) is the vehicle that witnesses an event, (ii) *Event Observer* (EO) is any node within one hop distance of an ER and (iii) *Event Participant* (EP) is any node beyond the one hop distance from the ER. Whenever an ER witnesses an event he assigns a *local trust* to it based on the information gathered by the vehicle's sensors. If that value is greater than a certain threshold then he transmits that information to all neighbors in one hop (EOs). When an EO receives a traffic event from an ER he stores it and observes the behavior of the ER. If the ER's behavior matches a model related to the traffic event reported the EO sends this message withing a certain ΔT time, which is enough for him to receive information from other EOs and EPs. At the end of the process every event is assigned a *global trust* based on the ER's behavior and on the *global trust* information sent by other nodes weighted by their role in the event. It should be noted that EPs will base their *global trust* solely on the information gathered from EOs and other EPs since they cannot directly observe ER.

The authors, however, do not take security into account. In their simulation scenario they consider a single vehicle forwarding false messages, which is not realistic since an attacker could easily report the same false event several times with different identities and successfully spread false messages. In addition, their system considers events reputation but not ER's; every role has a fixed reputation or weight assigned to it, which is what is taken into consideration when computing the event's *global trust*. As a result, there is no way to decrease the trust deserved by an ER who always reports false events.

Similarly, the authors in [84] introduce a scheme to report traffic events in VANETs that respects user privacy by using groups and offers security through trust and reputation. Their idea is to use group membership to provide individual users with privacy outside of the group while the *Group Manager* (GM) is responsible for adding new vehicles and evicting attackers or misbehaving members. A GM is identified by a certificate issued by a CA. Every group has a reputation (as a whole) in the network and every user contributes to it by sending group messages reporting traffic events. The GM has to be able to identify the real identity of the sender of a group message in order to protect the group's reputation against repudiation attacks. The regular flow of events is as follows:

1. Users periodically exchange messages with information of the state of the road.

2. Each receiver verifies that the message has a valid signature from the sender's group.

3. Each receiver computes how much he can trust the message based on the group's reputation and act accordingly, i.e., if he receives a trusted traffic jam alert he will take another route.

4. After taking a decision, the receiver vehicle may be able to know the real state of the road through direct observation. In that case he will update his level of trust on the group or groups that sent him information about this event. False messages are collected and eventually reported to the CA, which forwards them to the responsible GM to take appropriate measures.

Even if the authors do not mention it in their article, we believe that users require access to the CA every time they receive a group message because if a GM is revoked they need to be able to check if his identity is in the *Revocation List* (RL). In addition, the authors do not mention any mechanism for the group members to see if their GM is misbehaving by not evicting misbehaving nodes. We believe a lot of trust is placed on GMs, which could disclose the group member's identity to third-parties. Moreover, in the event of a traffic jam, only those vehicles which do not heed the warning and have the opportunity to make a direct observation will know the truth. If all users believed an attacker's warning he would be able to completely redirect the traffic on a road and he would not be punished because no other vehicles would be able to directly observe the event. Finally, it should be noted that there is no security mechanism to prevent a user A from lying about an event reported by another user B, which would make B's group manager punish B.

In [31], the authors propose a general information scheme (not necessarily directed towards traffic events) where every user is not only responsible for the events that he reports but also for the information that he forwards. In this scheme every vehicle is uniquely identified through the use of cryptographically self-generated addresses [85] and the authors assume that their scheme is immune to Sybil attacks. Information can be sent by anchored sources (trusted by default) and by mobile devices (whose level of trust is determined by a reputation system). Mobile devices are accountable for verifying data before propagating it. Therefore, whenever a user receives a message or segment he checks if it was originated at one of his trusted sources, if so that information is automatically trusted. If the segment is received from a source classified as malicious (by reputation) it is immediately discarded. Every time a segment from an unknown source is received

a *verification session* starts. If the number of received segments from unknown nodes reporting the same event reaches a threshold value all reporting nodes are promoted to trusted. Similarly, if an unknown user reports the same event than a trusted user, he becomes trusted as well.

We believe that this scheme fails to protect the users' privacy since they always use the same identity (an attacker could easily profile their routes). The authors do not take into consideration that even a trusted originator of an event may be interested in spreading false information at some point. In addition, their proposal heavily relies on anchored resources that only distribute reliable information, which may not be realistic. Finally, the idea of only forwarding information after it has been verified is not without its risks considering the ephemeral nature of a VANET.

The authors in [32] present another solution to distribute safety related information by broadcasting events (traffic jams, accidents and vehicles braking) which uses a reputation system to detect and isolate malicious nodes. Their algorithm is divided into the following phases:

1. *Neighbor discovery*: whenever a node S needs to forward an event received from one of its neighbors, it sends a neighbor discovery request to which its surrounding nodes reply with their identities. Each of the receivers R of that discovery request will check in its trusted nodes table if it trusts S and respond to the discovery request only if it does. If the identity of S is unknown to R then R adds S to its trusted nodes table with a trust level $(MAX\_TRUST - MIN\_TRUST)/2$. Similarly, when S receives the discovery responses it will update its trusted nodes table following the same criteria.

2. *Data dispatching*: once a node has discovered its neighbors it broadcasts the event information.

3. *Decision making and trust updating*: packets reporting events beyond a certain distance d are discarded (far away events are considered irrelevant). The next step is to see if the node itself is in the detection range of the event: if it is, the node will be able to judge if this event is true or false and update his trust on the reporting node accordingly; if it is not in range, it collects information from other neighbors for a time t and only if the number of reporters exceeds a certain threshold the event is considered true (either way, after t expires the level of trust on the reporters will be updated accordingly).

4. *Neighbor monitoring*: the authors assume that a genuine packet will always be broadcast, whereas false information will be unicast towards a certain

node. Based on that, nodes should monitor the network observing its neighbors behavior.

The authors are not clear on whether they use *Public Key Infrastructure* with a CA. If they assign to unknown nodes levels of trust greater than what misbehaving nodes have, it will always be easy for attackers to change their identity once they are discovered. A CA would be able to prevent that by linking the identity if the vehicle's license plate, for example. However, if they use a CA vehicles need to be in permanent connection with it to receive updates on the CRL, which requires a heavy road-side infrastructure. In addition, using always the same identity introduces a tracking vulnerability for the users.

In [33] the authors present a scheme to distribute traffic events information. They define a two tier approach: vehicle sensors first have to detect an event a certain number of times T_S before reporting it to the driver and if they have not detected the event for themselves, they need to receive the event warning from T_V vehicles before trusting it. Every time an event is detected T_S times a message including how many times the vehicle's sensors have detected the event and the identity of vehicles detecting it is send to the vehicle's neighbors. The receiving nodes will use this value and the number of vehicles that detected the event to determine if it is true or not.

We believe that the major problem with this scheme is that it does not address security at all. The authors do not consider the possibility of misbehaving nodes (intentionally or just due to the usual degradation of components). In addition, this solution is an event reputation system, but not a user reputation system, which means that the system has no memory over previous events recommended by a certain user and therefore all users can be equally trusted, which is a unrealistic assumption.

[86] presents a solution to manage a reputation system in the early stages of VANETs. The authors consider a scenario where the density of *smart* vehicles equipped with wireless communications is too low to allow for V2V communication. As a result, their scheme relies on the distribution of RSUs to handle the reputation scheme. Ideally, vehicles will always follow the same route (to work places, schools, superstores, etc.) and therefore be periodically in contact with the same RSUs. Depending on the desired deployment cost, the authors distinguish between two different designs:

- Isolated RSUs: if RSUs are not directly connected to each other, they need smart vehicles to forward their messages. This format of communication is called *Delay Tolerant Network* (DTN) [87]. In a nutshell, every vehicle

is assigned an *Agent RSU* which keeps track of its reputation and provides the vehicle with a certificate with its updated reputation. The other RSUs will monitor the vehicle behavior, i.e., forwards messages between RSUs, correctly reports traffic accidents to the RSUs, etc. Each RSU will use smart vehicles to forward this information to the vehicle's *Agent RSU* so that it can update the vehicle's reputation.

- Internet-accessible RSUs: in this scenario there is no need to distinguish between the *Agent RSU* and the others. Since they are all communicated, a vehicle can obtain its reputation update from any of them.

The authors also take into account the possibility that a user might take a different route which does not pass by any of his usual RSUs, e.g., he goes to work from Monday to Friday but Saturday and Sunday he drives to a different location. The solution they propose is to increment the validity period of the reputation certificate, so that on Friday the user receives a certificate valid until Monday.

We believe this is an interesting approach to the initial stage of a VANET. However, there are several drawbacks. For instance, road condition alerts will not be delivered immediately upon detection because there is no V2V communication. Secondly, the authors consider a scenario where a user takes an alternative route, although they need to plan ahead so that the RSU can give him an extended reputation certificate. In our opinion, this is not realistic since people are only predictable up to a certain point. Finally, the Internet-accessible RSUs model brings out the problem of having a network of connected devices which register every move made by every user, which poses a threat to user privacy.

2.6 Vehicular Traffic Simulators

The main goal of these simulators is to generate a trace which accurately portrays how vehicles behave on the road.

The *Multi-agent Microscopic Traffic Simulator* (MMTS) developed at ETH Zurich [88] is capable of simulating public and private traffic over real regional road maps of Switzerland with a high level of realism[2]. MMTS models the behavior of people living in the area, reproducing their movement (using vehicles) within a period of 24 hours. The decision of each individual depends on the area it lives in. The individuals in the simulation are distributed over the cities and villages according to statistical data gathered by a census. Within the 24 hours of simulation, all individuals choose a time to travel and the mean of transportation

according to their needs and environment. For example, one person may take a car early in the morning to go to work while another goes shopping using public transportation in the afternoon.

The street network that is used in MMTS was originally developed for the Swiss regional planning authority. The major attributes of each road segment are type, length, speed, and capacity. The simulation's result is a 24 hour detailed car traffic trace with almost 260.000 vehicles involved, with more than 25.000.000 recorded vehicles direction/speed changes in an area of around 250 km x 260 km.

Simulation of Urban Mobility (SUMO) [89, 90], is another example of microscopic traffic simulation. It simulates how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows to address a large set of traffic management topics. It is purely microscopic: each vehicle is modeled explicitly, has its own route, and moves individually through the network. It should also be noted that it can extract road topologies from maps obtained from the TIGER database [91, 29].

In VanetMobiSim [92] the authors take into consideration multiple factors to produce detailed vehicular movement traces, e.g., obstacles, vehicles characteristics, human driving patterns, intersection management, etc. According to the authors, VanetMobiSim combines a macroscopic and microscopic approach to produce more realistic results. It should also be noted that, like SUMO, it can extract road topologies from maps obtained from the TIGER database. The authors include as well some interesting results regarding the execution time on a an average computer (Intel Core2 Duo at 2.2 GHz with 2 GB of RAM). VanetMobiSim can simulate 5.000 vehicles in a 2 km x 2 km area in over 30 minutes.

For *poiSim* it was decided to use the MMTS traces, mainly because they contain 24 hours of over 260.000 vehicles moving over a realistic map of Switzerland and because they are publicly available for download.

2.7 Network Simulators

The authors of [93] divide network simulators between commercial-based and open source. In the first group we may find OPNET, QualNet [94, 95] and OMNet++ [96, 97]. They all contain a large number of network protocols for wired and wireless networks.

In the second group we may find ns-2, its evolution ns-3 [98] and GloMoSim as the most representative network simulators. ns-2 and ns-3 are discrete event

[2]Vehicular traces are publicly available from http://www.lst.inf.ethz.ch/research/ad-hoc

simulators targeted at networking research, which provide substantial support for simulation of TCP, routing and multicast protocols over wired and wireless (local and satellite) networks. ns-3 is more efficient than ns-2 and offers new features to help program simulations, although there is still an ongoing effort to port all protocols supported by ns-2. GloMoSim has basically the same functionality as ns-2, although it simulates fewer protocols due to the smaller GloMoSim support community.

It should be noted that, to the best of our knowledge, none of these simulators are able to handle simulations in the order of hundreds of thousands of nodes, and therefore process the MMTS traces. In [88] the authors are aware of this limitation and select smaller regions from the trace to run their simulations with ns-2. That is precisely why we were inclined to design our own simulation tool, so that we were able to process the entire trace.

2.8 Visual Light Communication

One of the goals of this thesis is to study how Visual Light Communication (VLC) would impact VANETs. For that reason we first need to give a brief introduction.

The predecessor of modern VLC was the photophone invented by Alexander Graham Bell and Charles Sumner Tainter [99]. The device consisted of a transmitter which modulated a light beam with a person's voice and a parabolic receiver on the other end which converted the light back into sound. The transmitter used a mirror which vibrated with voice, thus alternating between convex and concave forms and dispersing and focusing the light. The receiver had selenium cells at its focal point, which made possible to convert the light back into voice due to its photovoltaic properties (its electrical resistance is higher when in the dark and lower when exposed to light). The invention was successfully tested over a distance of approximately 213m using plain sunlight as their light source.

VLC uses visible light, with a wavelength between ∼400nm (750THz) and ∼700nm (428THz), to transmit information. It is based on the usage of a white LED emitter and a photodiode as a receiver.

The authors in [16], classify white LEDs into two types: (i) devices that use separate red-green-blue (RGB) emitters and (ii) blue emitters used in combination with a phosphor that emits yellow light. The former has a greater bandwidth while the latter has lower complexity.

As far as data rate is concerned, in [16] the authors present their results building a VLC link between an emitter and a receiver using a pre-equalized 45MHz band-

width white LED, reaching a speed of 80Mbps with *On-Off Keying Nonreturn-to-zero* (NRZ-OOK) over a link of 10cm (a distance which could be extended by using an array of transmitters, according to the authors). Similarly, in [15] the authors present their experiment using post-equalization, which reaches 100Mbps over a 10cm link, although the range could also be extended by using an array of transmitters.

The Visible Light Communications Consortium shows in [100] a wide variety of applications for VLC:

- a prototype which transmits sounds through RGB lights, where each RGB light has the sound of a different instrument: guitar, keyboard, etc.
- usage in restricted areas like aircrafts and hospitals.
- in a supermarket, product information could be acquired by the visible light receiver installed on the shopping cart
- indoor navigation systems
- wireless LANs

As the technology matures it will be possible to extend optical wireless networks to the outdoors. For instance, in [101, 102, 103, 104] the authors use lasers to transmit information and, in particular, to solve a problem commonly referred to as the *first/last mile problem* [101, 104]. In the early days of optical fiber deployment, the fiber connected a telecommunication company's different switching stations while consumers connected to those stations through twisted-pair wiring, which in effect limited the network access rates. Optical wireless proposed to bridge this gap and connect consumers directly to their closest switching station with a laser link, thus improving data rates and minimizing deployment costs.

In our view, in the next decade we will see vehicles transmitting information with their headlights or receiving information from traffic signs, as envisioned in [100, 105, 106]. However, there are several aspects that need to be addressed first, like the low transmission speed over a long link (speed rapidly decreases as the distance increases, from 100Mbps in a 2 meters link to 115Kbps for approximately 5 meters [100]) and how to transmit in movement. In addition, in order to succeed in the open air it must overcome the interferences caused by meteorological conditions (e.g., fog, rain, etc.).

On the plus side, VLC has important advantages over radio communications such as: practically unlimited bandwidth (unlike the hyper-regulated radio spectrum), a relatively low power consumption and resilience against jamming and DoS attacks.

2.9 Conclusions

Three main techniques for achieving privacy have been discussed in this section: *anonymous certificates*, *group signatures* and *pseudonyms/pseudonymous certificates*. All of these techniques have been widely studied throughout the literature and from our point of view are mature enough. The use of these techniques (or a combination of them, as we have seen) in VANETs is generally justified by the fact that they contribute to the users' privacy. However, by taking a closer look at the methods described in this section we realize that in order to keep the users' identity traceable under some circumstances those methods need PKI. Therefore, the need for revoking certificates and managing large CRLs. It has been shown that applications may face that particular problem in different ways. Some may appoint certain nodes as message verifiers and they will be the only ones working with CRLs. In the global picture, that could give the impression of efficiency (since the amount of nodes repeating work decreases) although that is certainly not a good idea for safety message applications because most of the network works blindfolded. On the other hand, some other schemes may apply techniques to compress the CRLs like Bloom filters or to directly reduce the amount of certificates that need to be revoked, and thus included in the CRL. We believe extensive effort will be dedicated to reduce the CRL size as done in [47, 46] and to study the most efficient ways to distribute it ([48]).

Special mention deserves the work of [49] for considering the effectiveness of pseudonyms change. Privacy is a major concern in VANETs security, and so far the use of pseudonyms seemed to be a perfect solution for the traceability problem. We believe that extensive research should be performed to verify if the authors claim of complete traceability holds for equipment rates higher than 20%.

Several of the articles covered in this background section introduce schemes designed to evict nodes from a VANET while there is no direct connection to the CA, problem that could be easily solved using cellular technologies to establish that link (as described in section 1). As seen in [67, 64, 68], roadside infrastructure and enhancing the vehicles sensing capabilities are valuable assets to verify other vehicle's messages and prevent Sybil attacks. In our view, preventing mul-

tiple identity attacks is of paramount importance to protect the honest majority hypothesis on which so many protocols rely. However, we foresee that approaches following [68] will be very seldom used since the generation of models of expected sensed behaviors for each of the possible alerts with a reasonably low rate of false positives seems to be a daunting task if feasible at all.

As far as information aggregation is concerned, in our view it is a process of paramount importance in VANETs. Hundreds of vehicles transmitting information and relaying that very same information to other vehicles next to them in a multi-hop network. Besides, considering the number of samples a vehicle takes every minute is enough to make us realize of the large traffic load involved in VANET applications (particularly in safety messaging). Therefore, if there is any way to decrease the network traffic load it should be exploited.

We find particularly relevant the contribution of [75] where the authors give an analytical measure of scalable data aggregation schemes. We also consider *intelligent delay control* policies a field were extensive research needs to be performed, since they can help optimize the use of the wireless medium.

SDA schemes are defined according to whether the aggregation is syntactic or semantic and thus the proposed schemes are bounded on what kind of aggregation is performed. Furthermore, most of the schemes are bounded on whether the aggregation is performed in fixed or dynamic areas and who is the node that aggregates the information. A general framework for both semantic and syntactic aggregation would facilitate the definition of SDA for any kind of application and network topology. In this direction, papers [7] and [76] are the ones that contribute to more general specifications.

Some SDAs make use of TPDs, such as [7], and the whole aggregation process depends on their correct behavior. As already discussed in this section, *Tamper Proof Devices* are not always as tamper proof as they should be. Therefore, we consider protocols that place their security on TPDs to be inherently flawed.

Moving on to the reviewed information distribution schemes, they were reviewed as a representative sample of information distribution applications. Most content distribution applications rely on a heavy roadside infrastructure to handle CRLs and access to the network information, which may not be realistic during VANETs deployment and early stages.

As far as reputation systems are concerned, the presented solutions suffer from the following drawbacks: [30] fails to take security into account, [84] relies heavily on the CA and on the group manager, [31, 32] do not take into consideration users' privacy and [86] makes users request their certificates when they are planning their trips, which is inflexible.

Regarding network simulators, as already mentioned in section 2.7 none of the reviewed simulators are able to handle simulations in the order of a hundred thousand nodes. Together with the need of a realistic network simulation, these were the main motivations to design our own simulation tool in combination with the MMTS vehicular traffic traces.

Finally, VLC is an emerging research topic and promising results have already been shown in [100, 15]. Once data rates increase for long links ($\sim$ 5 meters) the potential of this technology will be fully exploited and VANETs will be resilient to jamming and DoS attacks.

Chapter 3

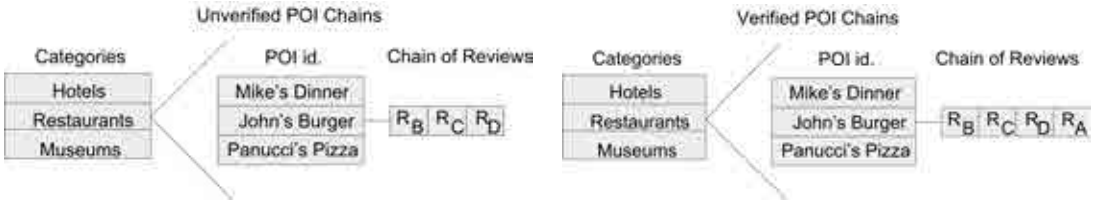
Chains of Trust: a Points of Interest Dissemination Strategy

Now that the background on security and reputation systems has been provided, we can introduce *Chains of Trust*, a reputation system that distributes Point of Interest (POI) information while preserving user privacy.

3.1 Scheme Overview

In the reputation system, every vehicle needs to store information about other vehicles and POIs (whether received from other users or reviewed by himself). Every node in the network shall store:

- POI chains: they are a series of reviews of the same POI from different users. As depicted in Fig.3.1 POI chains can be divided in:



(a) Unverified POI chains organization (b) Verified POI chains organization

Figure 3.1: POI chains organization.

  - Unverified POI chains: they contain POI reviews that the user has received from other users but which he has not yet been able to verify (by visiting and rating the POI himself), e.g., a traffic jam alert or the review of a new restaurant. Every unverified chain is rated based on the level of trust the user has in the known reviewers in the chain. When a user queries his vehicle, the POIs information is displayed ordered by that rate as defined in section 3.2.1.
  - Verified POI chains: once the user has a chance to check if there really is a traffic jam or how good that restaurant is, he evaluates the reviewers in the unverified chain and updates his level of trust in them depending on how truthful they were and marks the chain as verified. Verified chains are an essential part of the exchange of information between users, as will be explained in section 3.4.

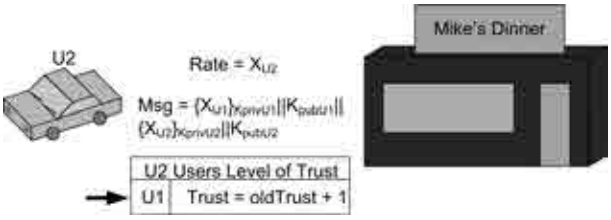
- Trust levels in other users (per category): every node needs to remember how much he trusts other users based on the verification of previous reviews. Besides, nodes not only share information about POIs, but also information about other nodes. For those recommended nodes several other properties will have to be stored, as will be detailed in section 3.3.

- Information about the latest messages from every user, both about POIs and nodes, should be stored for misbehavior detection, e.g., if the user is in a misbehaving strike. Further details will be given in section 3.6.

Since there is no Certification Authority (CA), every user or vehicle will create its own pair of public and private keys (of length L) and will be responsible for its securing. Notice that K_{pub} is the user identifier, therefore L should be long enough to ensure the statistical uniqueness of identities. That is why the scheme uses RSA with 1024 bits long keys. The private key will be used to sign information about POIs and about the levels of trust that a particular vehicle has in the others, while the public key will be attached to that information so that the rest of the network can verify the signatures correctness. For instance, consider the scenario depicted in Fig. 3.2. Imagine that a user U_1 goes to a restaurant A and he likes it. U_1 will broadcast a message to the other users in the network saying that restaurant A deserves a certain rate χ, signed with his K_{priv} and attaching his K_{pub}. All the other nodes that successfully receive the message store the unverified chain for future reference. When another user U_2 queries his own vehicle for a place to have lunch the vehicle returns a list of places recommended by other users (among

(a) U_1 reviews *Mike's Dinner* and broadcasts the message.

(b) U_2 queries his vehicle for a restaurant.



(c) U_2 follows U_1 recommendation, adds his own opinion to the chain and updates his level of trust in U_1 according to how similar both reviews were.

Figure 3.2: General behavior of the *Chains of Trust* protocol.

which is U_1's recommendation). If U_2 decides to go to A he will afterwards input his review into the system and if he liked it as much as U_1 his level of trust in U_1 will increase, or decrease otherwise. Regardless of how much he coincided with U_1's opinion, U_2 will append his signed review to the original, together with his K_{pub}, and broadcast the message. In this way, every time a user follows and verifies a recommendation he can update his level of trust in n other nodes (where n is the length of the chain of signatures), thus increasing the speed at which the reputation system develops.

In order to foster the development of *Chains of Trust* at an early stage vehicles could be pre-loaded with a set of POIs at the same time the application is being installed. In this way, users could benefit from the application since the very beginning, even compensating for a low initial adoption rate. In addition, those pre-loaded POIs could help users moving through a new area where they do not know anybody else, as will be described in section 3.5.

As mentioned above, this technique does not require a CA, or any road side infrastructure for that matter, since the network is completely ad-hoc and there is no certificate revocation process to manage. Every user generates his own pair of keys (the public key being his identity) and begins to play a part in the network by signing information. In the beginning, his identity is unknown to the rest of the users, therefore, he has to gain the others trust by telling the truth. That is how

the scheme protects itself against misbehaving. If an attacker misbehaves from the start he will not be able to inflict any real damage since all the nodes join the system with the lowest level of trust, and his reviews will be mostly unnoticed. If he tries to gain some credit and then misbehaves the *Rewards and Penalties* system will recognize a misbehaving strike and punish it. Although nothing prevents the attacker from creating a new identity he will not gain anything from it, since any new identity has no credit on the network. It should be noted that the level of trust of one user in another will decrease if the second either lies to him by misbehaving or if he rates a POI significantly different than the first would. Therefore, the terms lies and disagreements shall be used indistinctively throughout the thesis. More details on misbehavior can be found in section 3.7.

As far as the application's platform is concerned, we would like to elaborate on why *Chains of Trust* is specifically a Vehicular Ad-hoc Networks (VANETs) application and not appropriate for other mobile platforms, e.g., smartphones, PDAs, etc.. For starters, vehicles provide enough energy for the required periodic exchange information and for a fast enough processor to handle RSA encryption and decryption operations. Secondly, a larger amount of memory could be installed in order to store more data about the vehicles in the user's *Web of Trust*. Finally, vehicles allow for the installation of antennas with better gain, improving message reception and giving us the possibility to extend the transmission range. In addition, we believe that an application specifically conceived for smartphones would require a completely different solution. With 3g network access, users could connect to a remote server only when they needed to query for a POI category or submit their own POI reviews, which would mean that this remote server should have enough resources to store all the users' information. In addition, a CA would need to issue and distribute certificates to allow users to securely authenticate with the server. This a completely different scenario from our ad-hoc network proposal, which requires no infrastructure (remote server or CA) and where the system knowledge is distributed among its users.

3.2 POI Categories and Records

Several POI categories shall be considered, and a different level of trust for each category for each user shall be kept by each vehicle, i.e., a user may be a good hotel reviewer and a terrible restaurant critic. The following is an example list of what may be considered a POI category:

- Traffic conditions

- Gas stations
- Grocery stores
- Restaurants
- Hotels
- Bars
- Museums
- General entertainment

For each category a validity period is defined, e.g., a hotel review may be valid for months whereas a traffic jam alert may expire within hours or even minutes. That validity period is necessary to prevent unfair punishments. For instance, if a user identifies a traffic jam and sends a message alerting the network and several hours later another vehicle passes by and sees no trace of it he should not decrease his level of trust in all the users who signed the alert message.

Before POIs can be reviewed we first need to give them a unique identifier consisting of common knowledge information:

$$Id = \{Category||POI\_Name||Postal\_Address||GPSCoords\} \tag{3.1}$$

The Postal Address and the GPS Coordinates fields complement each other, since it is difficult to give the postal address of a traffic jam or the GPS coordinates of a restaurant (unless you position your vehicle right at the door). It should be noted that the GPS coordinates will admit a certain margin of error due to the devices positioning error.

Whenever a user wants to review a POI, he will assign a rate to it and assemble a record R with the following information:

$$R = \{Id||Rate||Timestamp\} \tag{3.2}$$

Each record has a timestamp so that users are able to keep track of the the validity period per category. In addition, it could also be used to remove old entries from the trusted nodes table.

Once the record has been prepared, the sender needs to sign it (by encrypting the record's hash with his private key) and attach his public key to it. At some point in the future the vehicle will broadcast M.

$$M = \{R_1||\{H(R_1)\}_{K_{priv_A}}||K_{pub_A}\} \tag{3.3}$$

Afterwards, when a vehicle receives a message it stores it for future use. When a user queries his vehicle for a recommendation on a POI category in a certain area the system answers with a list of received POIs, the ordering of which follows the criteria defined in section 3.2.1. If the user follows the recommendation he will be able to write another review about the recommended POI. The idea is to keep the previous reviews and attach the latest to the group, thus forming a chain of signatures that grows until a parameter n. By keeping a chain of size n every time that a user follows a recommendation he will be able to update his level of trust in n other users. It should be noted that the new added records are a slightly modified version of the first because they contain the hash of the original POI Id, instead of the complete identifier.

$$R' = \{H(Id)||Rate||Timestamp\} \tag{3.4}$$

The Id field (or its hash to be more precise) needs to be included in each of the added records to prevent a security vulnerability. Imagine that the messages were shortened by removing the Id to decrease the transmission time and to save storage space in the vehicles. Then, only the first record of the chain would be bound to the POI. As a result, it would suffice for a misbehaving node to replace that first record with another POI Id and broadcast that message over the network to ruin the reputation of the other signers. A good alternative would be to use Onion Signatures (as described in [8]) to preserve the message integrity every time a new record is added. However, Onion Signatures do not take into account that a message cannot grow indefinitely and at some point new records will replace old ones which deems this scheme unfeasible since in order to preserve its integrity not a single bit of information can be discarded.

A message containing a chain of length 2 is of the form:

$$M = \{R_1||\{H(R_1)\}_{K_{priv_A}}||\{R'_2\}_{K_{priv_B}}||K_{pub_A}||K_{pub_B}\} \tag{3.5}$$

It should be noted that the added records are not hashed and then signed, but directly encrypted with the user's private key. Since R' includes the hash of the POI identifier, it already is a short message. Therefore, the use of digital signatures on it would make the hash function redundant.

3.2.1 POI Chains Grading

When a user Q queries his vehicle for a certain POI in its vicinity, the system needs to display all received recommendations following a certain order. In the case of

verified chains that order is determined by the rate the user assigned to a POI the last time he was there. In the case of unverified chains the order is defined by the trusted (and in some cases by the most trusted) nodes in the chain. Let us define n as the number of reviews in a certain chain POI_1, $U_1, ..., U_n$ as the users whose POI reviews are in the chain and $\hat{U}_1, ..., \hat{U}_n$ as the subset of those nodes known by the user Q, χ_{POI_1,U_1} as the rate that U_1 gave to POI_1 and $\lambda_{\hat{U}_i}$ as the level of trust that Q has on U_i as a POI reviewer. Then the chain grade G is defined by:

$$G = \sum_{i=0}^{n} \left(\chi_{POI_1,\hat{U}_i} \cdot \frac{\lambda_{\hat{U}_i}}{\sum_{j=0}^{n} \lambda_{\hat{U}_j}} \right) \tag{3.6}$$

It should be noted that the rates assigned by unknown nodes are ignored as long as there is a known reviewer in the chain. Otherwise, the chain's rate is the arithmetic mean of the POI rates assigned by the unknown reviewers. Similarly, the reviews of the less trusted known nodes are ignored when there is a known node that belongs to the group of Q's k most trusted nodes. In order to prevent misbehavior only Q's most trusted users, i.e., the ones on the first k positions of the list, are considered for grading the chain. Otherwise, an attacker could create multiple low trusted identities and reduce the weight of legitimate reviews in Eq. 5.1 to obtain his desired result. By prioritizing the opinions of a small group of reviewers over the rest an attacker will first need to gain enough trust to belong into that group and once he starts misbehaving he will rapidly lose his influence, as described in section 3.6.

If k is too small some good and trustable reviewers' opinion will never reach the top of the list, and therefore their opinion will not count as much as it should (according to their good behavior). However, if k is too large an attacker could easily gain access to the top k reviewers group and start misbehaving. The idea behind Alg. 1 is to start with a low value and build-up. If the top k reviewers as a group gain more trust as the user reviews POIs the group can be expanded, which means more reliable information, and the user can prioritize their opinions over the rest of his trusted nodes. Otherwise, if one of the top k reviewers misbehaves then his own reputation will suffer, as described in section 3.6, and k will decrease to expel the misbehaving node and minimize the impact of his future reviews.

As a result, when a user queries his vehicle, the system replies with a series of recommendations starting with verified chains, followed by unverified chains with

Algorithm 1 How k is computed

```
//initial value
k := 1;
//every time a POI is reviewed by a user
for every POIReview
  //compute the mean of the level of trust of the k most trusted users
  previousMean := computeMean(trustedNodesList, k)
  //process the POI review, update level of trust in other users or create a new
chain if this is a
  //POI new
  processPOIReview(POIReview, trustedNodesList, k)
  //compute the mean of the level of trust of the new k most trusted users
  currentMean := computeMean(trustedNodesList, k)
  if currentMean > previousMean
    //the user's trust in the reviewers of the last POI has increased, therefore,
his list of k most
    //trusted users can expand
    k := k + 1
  else if currentMean < previousMean
    //one of the reviewers in the k-top has disagreed with the user, k needs to
be decreased to
    //prevent misbehavior
    k := k − 1
  end if
end for
```

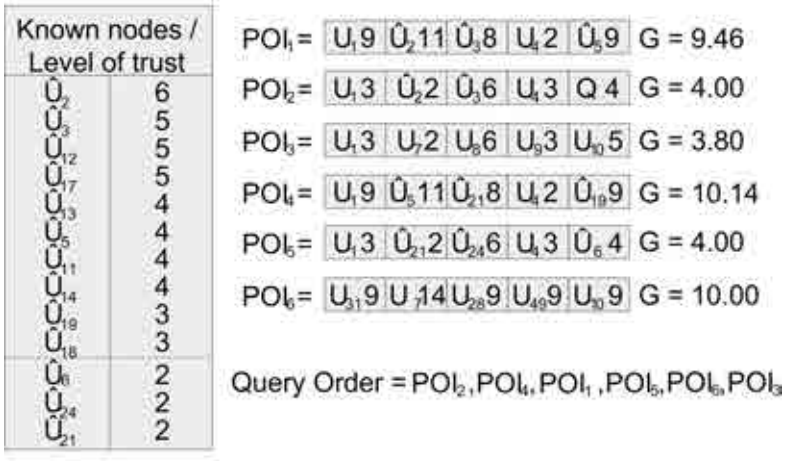


Figure 3.3: User Q chains grading process.

reviews by its k most trusted reviewers, followed by unverified chains with the rest of trusted reviewers and closing with unverified chains with unknown reviewers. The chain's rate establishes its position within its category. For example, Fig. 3.3 depicts the grading process of several chains by user Q and the order in which they are presented to the user: POI_2 (verified chain), POI_4, POI_1 (unverified chains with the most trusted known reviewers), POI_5 (unverified chain with the rest of known reviewers) and POI_6, POI_3 (unverified chains without known reviewers).

3.3 Nodes and Records

The use of user chains has to be carefully crafted in order to avoid abuse and misbehavior. Users in the network play two different roles: POI reviewers and other users reviewers. As POI reviewers, every vehicle has to store his level of trust in the other known users. As node reviewers, every vehicle needs to keep track of the nodes every other node recommends to him and their levels of trust as POI reviewer, because they impact on the level of trust the recommender deserves in that role. If a recommended node misbehaves (as POI reviewer) its recommender's reputation (as recommender) will suffer, or improve otherwise.

3.4 The Information Exchange

The application is designed to disseminate information about POIs among the vehicles in the network, thus the need for that information to flow from one vehicle to another. On one side there are POI chains (both verified and unverified) which represent the new information that comes into the system in the form of reviews of new POIs plus the re-evaluation of the already known. On the other, there are user chains, which are lists of known nodes and their level of trust. Basically, once two nodes know each other, besides exchanging information about POIs, they can exchange information about other users, thus increasing the speed with which the *Web of Trust* develops. The ideal way to exchange information would be for a user to issue a request for information on a certain category and its surrounding vehicles to answer it. However, it is not unusual that after having spent some time in a platoon formation a vehicle is alone or only has a few trusted vehicles in its vicinity at the time of sending the request. That is the reason why POI information should be exchanged periodically as well, and when the user needs a recommendation his vehicle still requests it to the nearby vehicles to complete what has already been gathered. As a result, the system provides the user with a satisfactory number of choices regardless of the trusted number of vehicles he has nearby when the request is sent.

Some would identify this periodic exchange of information as a tracking vulnerability. However, provided that the period between message exchanges is long enough (as explained in [2, 37, 36, 44]), if an attacker plans to track a user's movements he is going to need to physically follow him, since there is no road side infrastructure to collect the messages he is going to need to be in range. Further details on the period value are given in section 3.8.

Messages will include POI review chains from different categories. A smart exchange of information is also considered, where depending on external factors some categories will be more represented on the messages than others, i.e., gathering information about restaurants will be prioritized at lunch and dinner time, about gas stations when the vehicle is running low on gas, etc.

The following three types of message exchange are considered:

1. **Requests**: if a vehicle receives a review request he will reply with several POI chains for the requested category. Preferably, reviews that he has verified himself and which have the highest rate in the category. If not enough verified chains are available, he will reply with the highest rated unverified chains (following the rating criteria described in section 3.2.1). When the

requester receives the reply he considers all the chains in the message as unverified and stores them as such. Hence, the difference between verified and unverified chains (in the response) becomes subtle: only the verified chains have the sender's signature, whereas the unverified are just being forwarded. A user will not be penalized nor rewarded for forwarding unverified chains.

2. **Periodic Exchange**: vehicles should exchange POI chains periodically with the better rated POIs in each category. Our scheme prioritizes the recommendation of which POIs another user should visit over which POIs it should not. We would like to avoid a situation where a user knows many POIs with bad reviews and only a handful with good ones.

3. **Recognition Exchange**: if during a periodic exchange, one vehicle is recognized as a trusted user (from a previous encounter) then recognizer and recognized will exchange user chains and verified POI chains, although they will be marked as unverified by the receiver. Besides, the nodes and its level of trust included in the node chains will be added to the list of the previously known nodes, as explained below.

Requests and periodic exchanges of information are of vital importance for a user that is traveling or moving through a new area. They will both provide the user with unverified chains and once he reviews one POI in one of those chains he will be able to establish a level of trust for each of the reviewers, thus staring a new *Web of Trust*.

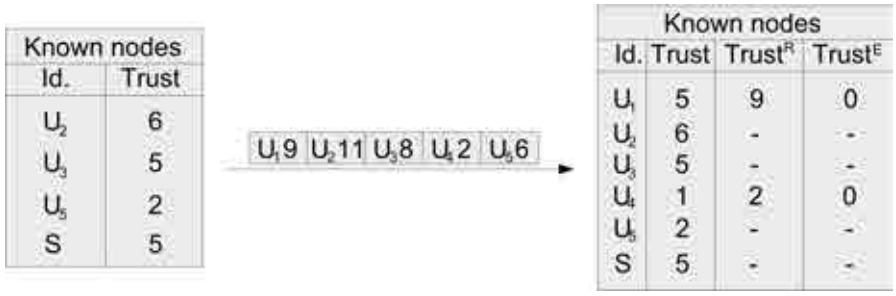


Figure 3.4: R's known nodes table before and after processing a Recognition Exchange message.

Fig. 3.4 depicts a *Recognition Exchange* between a user S and a user R, in which S sends a message M with his most trusted nodes.

$$R_U = K_{pub_U} || Level\_of\_Trust_U(as\_POI\_reviewer) \tag{3.7}$$

$$M = R_{U_1}||...||R_{U_5}||Timestamp||\{H(R_{U_1}||...||R_{U_5}||Timestamp)\}_{K_{priv_S}}||K_{pub_S} \tag{3.8}$$

User R adds U_1 and U_4 to the list of known nodes, with S as their recommender, λ_S^R as the level of trust R has on S as recommender and with an initial level of trust defined by the function:

$$T(U_i) = max(Trust_{U_i}^E, min(\lambda_S^R, Trust_{U_i}^R)) \tag{3.9}$$

After R has had a chance to receive several reviews from U_1 and U_4, S will be rewarded or penalized, depending on how similar is the level of trust that R has on them related to what S recommended. All nodes recommended by the same user are inextricably linked, i.e., the misbehavior of one may affect the others. In order to deal with misbehavior, trust on a certain node U_i is divided in the trust recommended by another user ($Trust_{U_i}^R$) and the level of trust result of R own experience with U_i ($Trust_{U_i}^E$). In this way, when a node misbehaves its recommender is punished and the λ factor (Eq. 3.9) is decreased for all nodes he recommended. However, $Trust_{U_i}^E$ will not be affected, and as a result nodes that have earned a reputation for themselves are no longer subject to the reputation of their recommender.

Finally, it should be noted that for every user in the recommended nodes message the receiver only processes those nodes he does not know: if a node knows another user, it means he has followed one of his recommendations and that is more important than a recommendation another user could make.

3.5 The Visitor Scenario

Whenever a user enters a new area and he requests a POI to the system, the system will send a request message and will present the received information together with the information received from periodic exchanges. If the user is in a completely new area it may be possible that he does not know any of the reviewers who have sent him POI recommendations for that specific region. Should that be the case, the system (when queried) will present the user a list of POIs with unverified reviews and a list of the pre-loaded POIs for that area. If the user chooses one of the POIs with unverified reviews, when he inputs his review afterwards he will update his level of trust on all the reviewers in the POI chain, thus gaining information on other users and the POIs they signed. If the user chooses one of the pre-loaded POIs he will start a new review chain with his review and he will not gain information on other users. The visitor situation needs to be considered

in detail, because it may closely match a tourist profile. On one hand, he will be completely new in the area and most or all POI reviewers will be unknown to him. On the other, precisely because he is a tourist he will input reviews more frequently than the average user and that will allow him to fast develop a new *Web of Trust*.

3.6 Rewards and Penalties

3.6.1 As POI Reviewers

Whenever a user U receives a recommendation and follows it, he can input his own opinion in the system. Based on that, his vehicle evaluates the recommendation chain updating the levels of trust in other users depending on the similarity of their rates to U's. If U has a positive impression of the recommended POI, all the other users in the chain that gave a positive review to the POI are rewarded; otherwise they are penalized. For this system to work, the penalty always has to be greater than the reward; otherwise, a user could cause as much damage to the system as much good he had previously done.

Even though it may seem like that the sole objective of this policy is the punishment of all those users that spread lies and misbehavior in the system, that is inaccurate. Misbehaving nodes is only a part of the problem, i.e., people tastes vary from individual to individual, thus so will their POI reviews. The main goal is not only to build a *Web of Trust*, but also a web of similar tastes, as previously stated.

There are several requirements the penalties system should comply with:

1. If a user A has received only a few messages from a user B and B *lies to or disagrees with* him, then his level of trust should be significantly decreased.

2. If a user A has received many messages from a user B and B *lies to or disagrees with* him, then his level of trust should be decreased, but not dramatically.

3. The system should be able to recognize misbehaving strikes, after several lies or disagreements in a row the level of trust in the misbehaving node should plummet.

A very good candidate for the penalties function is the exponential curve because it has a slow growth at the beginning and a steep increase as the rate of

lies or disagreements raises, which is appropriate to deal with misbehaving strikes. It was decided that the level of trust should range from 0 to 15 and that after 5 consecutive bad reviews the evaluator level of trust in the evaluated should be set to the minimum. Thus, e^x was discretized from 0 to 15 into 6 elements (as depicted in Fig. 3.5) to obtain the cumulative penalization function $f(x)$, where x is the number of lies. It should be noted that the rating system could easily be modified to operate in another range of values, e.g., from 0 to 10 (which might be more human friendly). The same can be said about the number of bad reviews. What is important is that the penalization function follows the requirements described above and every time a user misbehaves the penalization is greater.

$$\alpha = e^{\frac{1}{5}ln(15)-\beta} \tag{3.10}$$

where

$$\beta = \frac{\#good\_reviews_{evaluated}}{\#reviews_{evaluated}} \tag{3.11}$$

$$f(x) = (e^{\frac{1}{5}ln(15)-\beta})^x \tag{3.12}$$

The value that will be subtracted from the level of trust in the beginning of the misbehaving strike is $f(strike\_length)$. As described in requirements 1 and 2 the penalties function should take into account how many good reviews the evaluated user has sent over time, understanding by good reviews those whose rate difference with the evaluator's does not exceed a maximum value defined in the system, which is denoted by ΔOp. To that end β is included in the equation. It should be noted that for recommended nodes, as described in section 3.4, the level of trust to be decremented shall be both $Trust^E$ and $Trust^R$.

In Alg. 2 the pseudocode of the rewards and penalties function is presented. Consider $\chi_{U_1,A}$ as the rate user U_1 assigned to POI A. The first time that U_1 finds the difference between his rate and U_2's over a certain POI A is greater than ΔOp it marks node U_2 as misbehaving. The value of $Trust^E$ is stored as the rate at the beginning of the strike from which α^{length_strike} will be subtracted. If a user is in a misbehaving strike his level of trust will decrease faster. A misbehaving strike can be broken after the evaluator verifies $BREAK\_STRIKE$ good reviews from the evaluated. However, breaking the strike does not mean that the evaluated user goes back to its previous level of trust.

Algorithm 2 Rewards and penalties pseudocode

if $\neg misbehaving\_strike$ **then**
  if $|\chi_{U_1,A} - \chi_{U_2,A}| \leq \Delta Op$ **then**
    $Trust^E := Trust^E + 1$
  else
    $misbehaving\_strike := true$
    $Trust^E_{pre\_strike} := Trust^E$
    $\alpha := e^{\frac{1}{5}ln(15)-\beta_{U_2}}$
    $Trust^E := Trust^E_{pre\_strike} - \alpha$
    $Trust^R := Trust^R_{pre\_strike} - \alpha$
    $strike\_breakers := 0$
  end if
else
  if $|\chi_{U_1,A} - \chi_{U_2,A}| \leq \Delta Op$ **then**
    $Trust^E := Trust^E + 1$
    $strike\_breakers := strike\_breakers + 1$
    if $strike\_breakers = BREAK\_STRIKE$ **then**
      $misbehaving\_strike := false$
    end if
  else
    $\alpha := \alpha * e^{\frac{1}{5}ln(15)-\beta_{U_2}}$
    $Trust^E := Trust^E_{pre\_strike} - \alpha$
    $strike\_breakers := 0$
  end if
end if

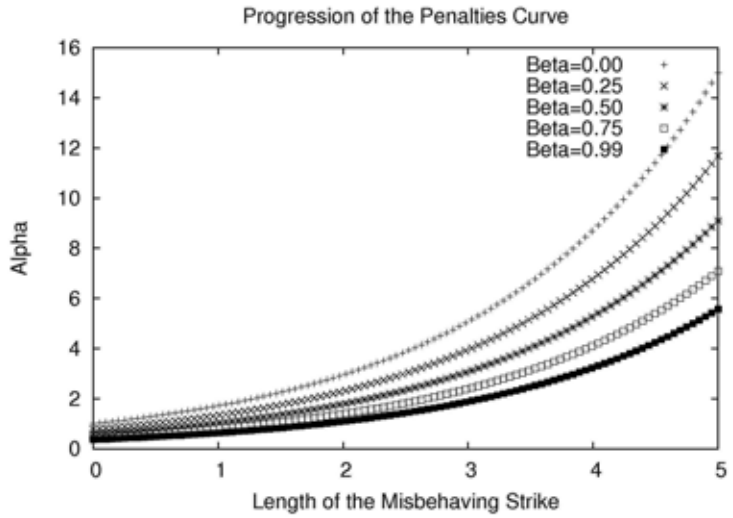


Figure 3.5: Progression of the function $f(x) = (e^{\frac{1}{5}ln(15)-\beta})^x$

3.6.2 As Node Reviewers

As node reviewers, a very similar system to the one described in the previous section will be used. In order to be considered a good recommender in our system, the proportion of good recommendations against bad needs to be at least 5 to 1. If it is, then the user's level of trust as recommender will be increased. If it is less, it will be decreased by α^n (Eq. 3.10 with $\beta = 0$) where

$$n = 5 \cdot \lfloor \frac{\#bad\_recommendations}{\#recommendations} \rfloor \tag{3.13}$$

It should be noted that the timestamp in both types of reviews (POI and other users) allows the system to discard old information and to avoid punishing the user for events that occurred a long time ago.

Finally, as detailed in section 3.4, by decreasing the level of trust on the user as recommender his recommended nodes level of trust becomes more dependent on $Trust^E$ and less on $Trust^R$.

3.7 Misbehavior

In this section we will elaborate on the different mechanism *Chains of Trust* implements to protect itself from the most common misbehavior or third party attacks.

- *False reviews spamming*: an attacker spreads good POI reviews (e.g., to promote his restaurant) or bad (e.g., to harm his competition). If the attacker is unknown to the rest of the users, then their level of trust in him will be 0 and as explained in section 3.2.1 his unverified POI chain will go mostly unnoticed. On the other hand, if the attacker has previously worked on gaining a certain reputation as POI reviewer, then the penalties system described in 3.6 will deal with the attack. As depicted in Fig.3.5, we can see that a few bad reviews are enough for a user to lose all his credit, e.g., after 3 bad reviews its reputation is decreased by 5.08 units. As a result, it can be concluded that the attack fails because the number of well intentioned reviews the attacker needs to send to build a reputation is much greater than the number of ill intentioned reviews he can send before he loses his reputation. It should also be noted that even if the attacker tried to use multiple identities to increase the length of the chain the same reasoning would apply and the attack would fail.

- *Nodes recommendation*: an attacker could create multiple identities, use one to recommend the others and use the latter to implement the *False reviews spamming* attack. As stated in section 3.4 only the nodes unknown to the nodes recommendation message receiver are added into his list of known nodes and they are added with a level of trust defined by Eq. 3.9. If the attacker gains a good reputation as recommender and then recommends a list of his own identities, thus constructing a web of misbehaving nodes, after several incorrect messages all of its recommended nodes' level of trust as POI reviewers will be based on $Trust^E$. Unless the recommended node has earned a reputation for himself, his level of trust as per Eq. 3.9 will be 0 and it would be as if it had never been recommended, rendering the attack unsuccessful.

In addition to what has been said above, the difficulty of launching an attack on a mobile target should also be considered. Due to the lack of road side infrastructure the attacker could not rely on compromised Road Side Units (RSUs) to help him launch a global scale attack and would have to use his own resources.

3.8 Analysis of *Chains of Trust* Scalability

The first step to determine if *Chains of Trust* can succeed in real life is to simulate the data transmission protocol for hundreds of nodes. To that end, a simulation in ns-3 [98] was implemented defining a vehicular scenario with 400 nodes arranged in 4 lanes as depicted in Fig.3.6, connected through a Wireless Access in Vehicular Environments (WAVE)-Direct Short Range Communication (DSRC) 27Mbps link with a 120 meters range. WAVE-DSRC has the mechanisms to provide different user applications with different channels while reserving certain channels for safety applications, others for control and others for public safety [14]. It should be noted that our simulation uses ns-3 *YansWifiPhyHelper* and *YansWifiChannelHelper* classes, as defined in [107].

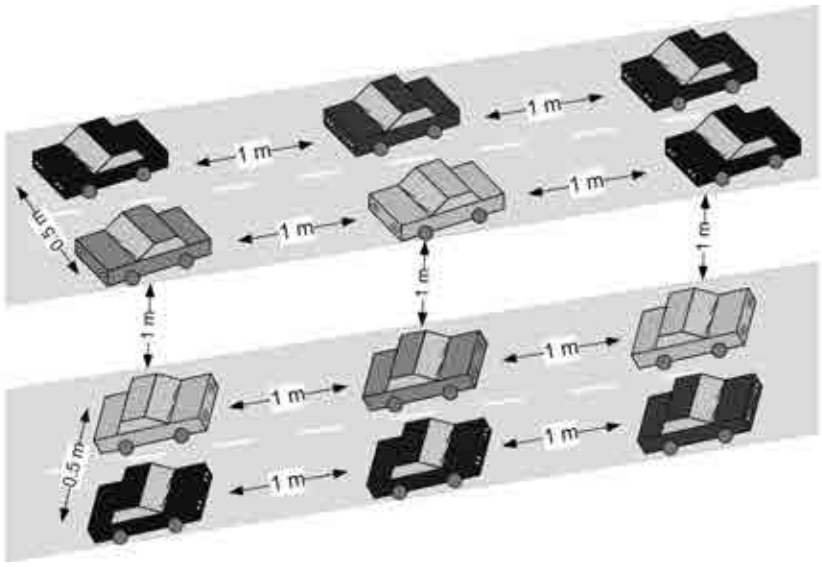


Figure 3.6: Vehicle layout for the 400 nodes simulated in ns-3.

In Alg. 3 we can see the simulation pseudo-code. Basically, the program schedules the broadcast of $numPackets$ 1000 bytes packets at a randomly chosen time between the start of the simulation and its ending point, defined as *period*. For every scenario ($numPackets/period$ combination) the number of broadcasts received by each of the 400 simulated nodes is computed ($results_{numPackets,period}$) and compared with how many broadcasts each of those nodes would have received without packets loss ($reference_{numPackets}$), considering the mean as the scenario's result:

Algorithm 3 Data transmission simulation pseudocode

```
period := 60, 120, 180, 240, 300 seconds
numPackets := 100, 200, 300, 400, 500, 600, 700, 800
for every element in numPackets
    for every element in period
        runSimulation (numPackets, period)
    end for
end for
function runSimulation (numPackets, period)
    setupWifi()
    setupVehiclesTopology()
    time := random(0.1, period)
    //schedules an event on time 'time' to send 'numPackets' packets of a 1000
bytes
    Simulator :: ScheduleEvent(time, numPackets, 1000)
end function
```

$$Received\ broadcasts\ \% = \sum_{node=1}^{400} \left(\frac{results^{node}_{numPackets,period}}{reference^{node}_{numPackets}} \right) \tag{3.14}$$

Looking at the results in table 6.1 it can be seen that to ensure a delivery rate over 90% while maximizing the amount of information being transmitted 400 packets is the best option for a 120 seconds period. For a larger number of packets there is a drop in reception due to the MAC collisions. If a shorter period is considered there is a slight drop in performance, although the major reason against transmitting every 60 seconds is limiting the amount of information an attacker can collect while following a target. We believe 120 seconds is more secure since the attacker has to be in range twice as long, while at the same time *Chains of Trust* can produce satisfactory results, as will be showed in the following sections.

3.9 How Will *Chains of Trust* Behave in a Realistic Scenario?

Chains of Trust is designed so that every vehicle is pre-loaded with a selection of a 100 POIs to provide information to users that have arrived to a new area

Percentage of received broadcasts					
Number of packets / Period	60	120	180	240	300
100	95.97	97.99	98.62	98.96	99.15
200	91.57	95.91	97.27	97.93	98.38
300	86.86	93.72	95.82	96.87	97.54
400	82.16	91.50	94.37	95.78	96.64
500	77.23	89.28	93.01	94.80	95.85
600	71.93	87.00	91.57	93.73	94.98
700	66.30	84.58	90.03	92.57	94.06
800	60.54	82.19	88.51	91.48	93.22

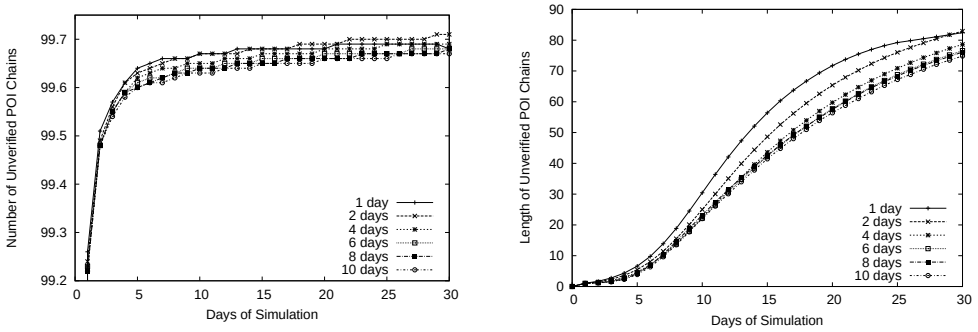
Table 3.1: Percentage of received broadcasts for every simulated scenario.

(as described in section 3.5). However, a user will not transmit a pre-loaded POI unless he visits it for himself, at which point he starts a new chain with his review and can be transmitted. Therefore, the pre-load will not impact in the results presented in this section.

The first test should reveal if the scheme is feasible in a realistic scenario. The application simulation tool *poiSim* (explained in section 4) will be executed for different reviewing rates, i.e., every user will input a review into the system once a day on average (1/1), once every two days (1/2), once every four days (1/4) and so on until a review is input once every 10 days (1/10).

In Figs. 5.3a - 5.3b the evolution of the number and length of unverified POI chains can be seen. After the first 5 days of simulation the number of unverified chains and its length (number of POIs it contains) is very similar regardless of the reviewing rate. The fact that the average number of unverified chains is over 90 and its length is almost 5 (considering a reviewing rate 1/6) means that there has been interaction between the users and some have already started to build a better reputation in the network. Considering the results after 30 days of simulation it can be seen that they do not differ significantly.

As far as verified chains are concerned in Fig. 5.3c the direct relation between the reviewing rate and the number of verified chains the nodes store can be observed, which is logical considering that every time a POI is reviewed its unverified chain moves on to the verified state. In the first 5 days of the simulation, the number of verified chains for reviewing rates 1/4 and 1/6 is 1.24 and 0.83 respectively. Similarly, Fig. 5.3d shows that the progression of the length of verified chains is



(a) Number of unverified POI chains: for every node the number of unverified POI chains is computed, their mean is the depicted result.

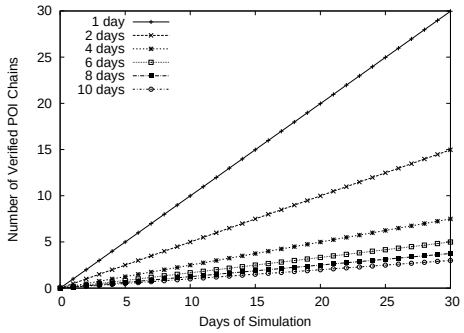
(b) Length of unverified POI chains: for every node the mean of its unverified POI chains length is computed, the mean of those means is the depicted result.

Figure 3.7: Evolution of the length and number of unverified and verified chains.

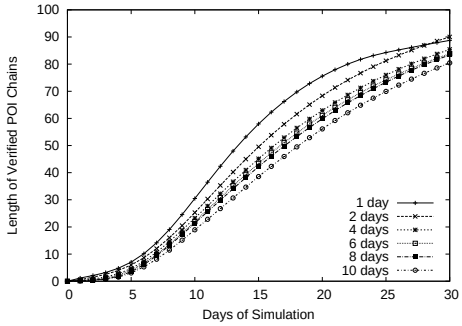
very close to the unverified depicted in Fig. 5.3b. Regarding the rate assigned to the POIs in the verified chains, in Fig. 5.3e it can be observed that the rate of the reviewed POIs varies until it stabilizes around 7, which is expected since the randomly chosen rates are distributed around that value, as described in section 4.1. The different simulated reviewing rates determine how fast the POI rate converges to that value.

The measure of the system success is given by how many users every user knows and what level of trust he has assigned to them. In Fig. 5.4a it can be observed that after the first 5 days of simulation every user has several other users in his known nodes list, going from 20.76 users on average for a review rate 1/1 to 2.11 users for a review rate 1/10. As expected, lower reviewing frequencies result in a lower number of known nodes. If a middle ground scenario is considered, review rates 1/4 and 1/6 yield 3.85 and 3.05 known users respectively. Results improve significantly after the first ten days of simulation, where reviewing rates 1/4 and 1/6 result in every node knowing on average 33.24 and 26.37 nodes, respectively.

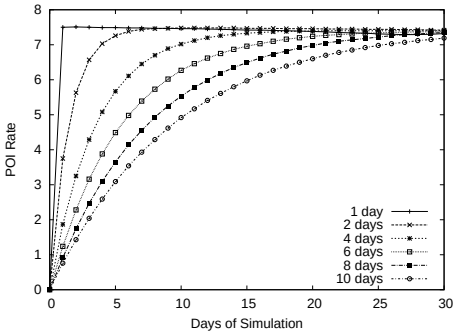
Regarding the rate or level of trust a user assigns to his known users, in Fig. 5.4b it can be seen that after the first 5 days of simulation for all reviewing rates the average level of trust is almost 1. As the simulation progresses, the level of trust may oscillate (as it can be seen for reviewing rate 1/1) due to the randomness of the simulation, although on the long run a larger number of chains are reviewed and the level of trust increases due to the higher proportion of good reviews. After the first 10 days, considering review rates 1/4 and 1/6 result in levels of trust of



(c) Number of verified POI chains: for every node the number of verified POI chains is computed, their mean is the depicted result.



(d) Length of verified POI chains: for every node the mean of its verified POI chains length is computed, the mean of those means is the depicted result.



(e) Rate in the verified POI chains: the mean of the rates users assign to POIs.

Mean of the deviation σ																					
Simulation day	1			5			10			15			20			25			30		
Rating freq. (review/days)	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10
Num. Unver. POI chains	59.86	63.26	63.69	33.09	36.27	37.08	31.01	33.70	34.61	30.06	32.30	33.24	29.76	31.34	32.30	29.62	30.70	31.64	29.60	29.99	31.12
Length Unver. POI chains	0.30	0.17	0.14	7.21	8.55	7.94	30.84	29.78	30.50	40.39	40.25	40.42	41.11	43.32	44.13	40.74	43.08	44.44	40.78	41.76	43.10
Num. Ver. POI chains	0.00	0.36	0.30	0.14	0.83	0.66	0.30	1.17	0.95	0.47	1.44	1.16	0.62	1.67	1.34	0.79	1.87	1.50	0.94	2.05	1.64
Length Ver. POI chains	0.00	0.00	0.00	7.81	6.21	3.97	32.05	27.09	20.65	44.28	42.12	34.37	46.98	49.37	43.25	47.57	51.74	47.99	48.01	51.87	49.93
Rate Ver. POI chains	0.00	0.00	0.00	4.35	1.91	1.24	4.35	3.11	2.23	4.34	3.73	2.91	4.34	4.03	3.38	4.34	4.19	3.69	4.32	4.26	3.91

(f) Mean of the deviation table for (a), (b), (c), (d) and (e).

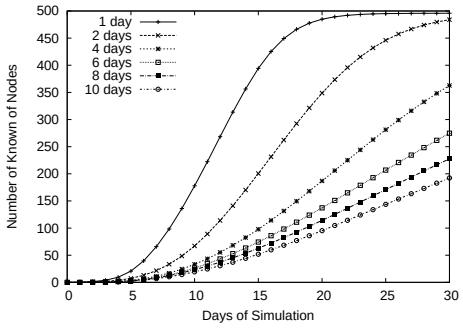
Figure 3.7: Evolution of the length and number of unverified and verified chains (continued).

1.31 and 1.45 respectively. In Fig. 3.8c we can see represented the level of trust in the 25% most trusted nodes each user has. After the first 15 days it can be seen how its progression differs from Fig. 5.4b, ending the simulation with a level of trust slightly over 4 for review rates 1/4 and 1/6.

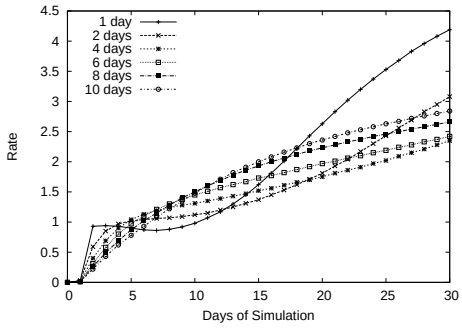
We believe this first experiment has proven that the system will in all likelihood succeed in effectively disseminating POIs information and building a *Web of Trust* among users in a real life scenario. Considering moderate reviewing rates of 1/4 and 1/6 we can see that just after the first 5 days of simulation every user has on average more than 90 unverified chains containing 5 user reviews, almost 1 verified chain with 5 reviews and more than 3 trusted nodes with trust levels over 1. It should also be noted that results significantly improve after 10 days of simulation. Therefore, it can be concluded that although the system will produce results from the very start, depending on the reviewing rate the it may need from 5 to 10 days (in the worst case scenario) to fully develop a *Web of Trust*. Users, however, will be able to take advantage of the application from the start by using as well the collection of pre-loaded POIs.

3.10 Chain Size Experiments

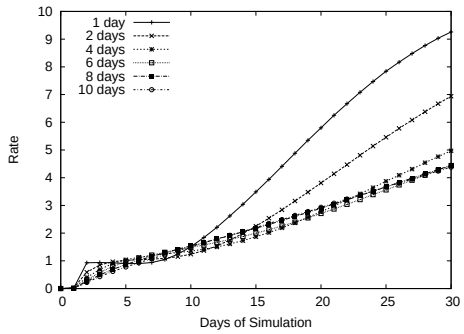
The length of POI chains is of paramount importance in the system because every time an unverified POI chain is reviewed the reviewer updates his level of trust in all its signers. Hence, the longer the chain the better the system should work. Naturally, the messages cannot be allowed to grow indefinitely because vehicles do not have an infinite amount of memory and the messages exchanged between vehicles should be relatively short due to wireless communication limitations. In section 3.9, *poiSim* was configured to allow chains up to a length of 225 reviews, but we would like to observe how does the system behave with shorter chains and verify if there is a certain frontier value where the benefits of increasing the length begin to decrease. Thus, the simulator was executed with POI chains of 75, 150 and 225 reviews. In addition, for this experiment every user will input a new review in the system every 1.800 seconds. Certainly, it is not very likely that users will input one new POI review every half an hour. However, once we have established the validity of the system, we would like to modify the reviewing rate to study the system in the long run. In Figs. 3.9a - 3.9b, for both unverified and verified POI chains there is a slight difference between using length 150 or 225 after 5 days of simulation. In a 5 days simulation both lengths are high enough to not be a limitation, but in larger runs we would definitely see a bigger difference



(a) Number of known nodes: mean of the number of known nodes by every node.



(b) Rate or level of trust of the known nodes: mean of the rates users assign to other users as POI reviewers.



(c) Rate or level of trust of the 25% most trusted nodes: mean of the rates users assign to other users as POI reviewers (only for the 25% highest rated nodes).

Mean of the deviation σ																					
Simulation day	1			5			10			15			20			25			30		
Rating freq. (review/days)	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10
Num. Nodes	0.17	0.00	0.00	26.91	9.43	8.11	127.91	43.84	37.66	119.57	84.33	68.95	52.82	118.10	100.82	34.07	140.27	126.49	32.49	148.91	142.56
Nodes Rate	0.00	0.00	0.00	0.87	1.53	1.03	1.42	1.53	1.68	2.17	1.92	2.17	2.99	2.32	2.55	3.76	2.71	2.85	4.31	3.03	3.10
Nodes First 25% Rate	0.00	0.00	0.00	0.87	1.04	1.03	1.48	1.50	1.65	2.08	1.79	2.02	2.52	2.03	2.23	2.80	2.25	2.35	2.97	2.41	2.43

(d) Mean of the deviation table for (a), (b) and (c).

Figure 3.8: Number of known nodes and their levels of trust progress.

in the length of chains the vehicles store. Therefore, from *Chains of Trust*'s point of view chains should be as long as the wireless communication between vehicles permits.

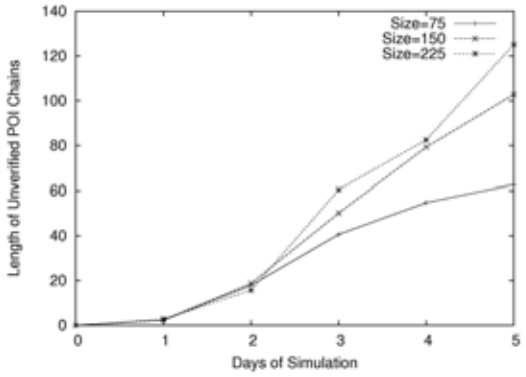
Fig. 3.10a shows a similar growth for the number of known nodes with the three simulated chain lengths. That is because after the first day of simulation most of this growth is a product of the exchange of recognition messages (which do not depend on the chain length). As far as the rates are concerned, in Figs. 3.10b - 3.10c there is a certain variation attributed to the randomness of the simulation, rather than to the chain length. It should be noted that user opinions of POIs are normally distributed with a mean μ that we termed its real rate. As a result, the mean of the rates of 75 reviewers should not differ much from the mean of the rates of 225.

In Figs. 3.10b - 3.10c the levels of trust progress as the simulation advances, although it provides conclusive evidence that longer chains do not lead to more trustworthy nodes. Therefore, in a scenario where recognition messages do not play such an important role on conveying nodes information, POI chains assume that responsibility. Mainly, because every time a chain is verified all the reviewers levels of trust are updated in the verifier. As a result, the length of a POI chain should only be limited by physical requirements such as the size of the message to be transmitted.

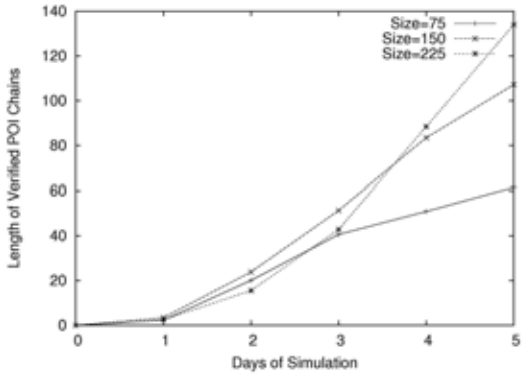
3.11 POI vs. Nodes Experiments

The purpose of this experiment is to discern how much of the system performance can be attributed to the exchange of recognition messages, or in other words, how is the system performance affected when node reviews are not exchanged. To that end, the simulation in *poiSim* was executed with 0, 25 and 50 node reviews per transmitted message and with a reviewing rate of a new review every 1.800 seconds.

In Fig. 3.11a the results of those simulations are plotted. As expected, the average number of users known by every user increases as the number of nodes in the message increases as well. However, it should be noted that the maximum number of nodes to be stored (500) is reached in the three cases during the third day of the simulation. Therefore, the exchange of node reviews does not represent a dramatic improvement in that aspect. On the other hand, Fig. 3.11b shows that the rates of the 25% highest rated nodes improve as a result of increasing the number of nodes in the message. This leads to the conclusion that recognition



(a) Length of unverified POI chains: for every node the mean of its unverified POI chains length is computed, the mean of those means is the depicted result.

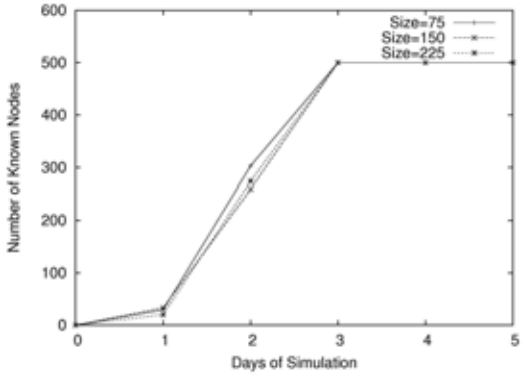


(b) Length of verified POI chains: for every node the mean of its verified POI chains length is computed, the mean of those means is the depicted result.

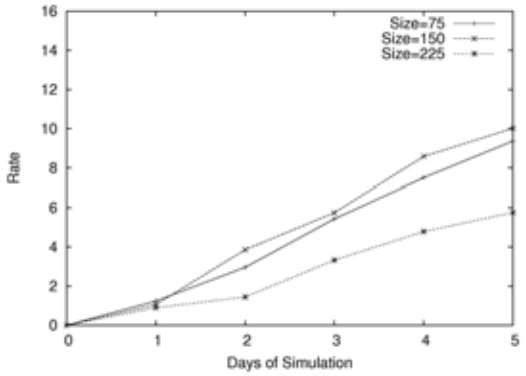
Mean of the deviation σ						
Chain type	Length	Day 1	Day 2	Day 3	Day 4	Day 5
Unverified POI chains	75	2.17	22.24	29.13	26.15	23.82
	150	1.59	25.90	47.16	59.93	54.01
	225	2.09	18.80	63.11	76.52	83.49
Verified POI chains	75	1.56	22.70	31.36	28.86	24.78
	150	1.64	30.11	47.73	49.21	51.95
	225	1.08	30.79	49.64	72.80	80.65

(c) Mean of the deviation table for (a) and (b)

Figure 3.9: Evolution of the lengths of unverified and verified chains.

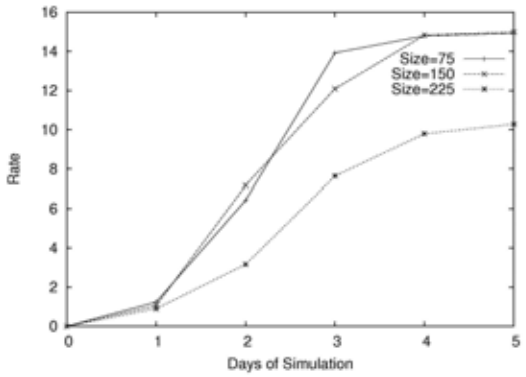


(a) Number of known nodes: mean of the number of known nodes by every node.



(b) Rate or level of trust of the known nodes: mean of the rates users assign to other users as POI reviewers.

Figure 3.10: Number of known nodes and their levels of trust progress.



(c) Rate or level of trust of the 25% most trusted nodes: mean of the rates users assign to other users as POI reviewers (only for the 25% highest rated nodes).

Mean of the deviation σ						
Data	Length	Day 1	Day 2	Day 3	Day 4	Day 5
Nodes rate	75	2.08	4.91	5.29	5.40	4.94
	150	2.11	3.94	4.51	5.02	4.14
	225	1.44	3.05	3.20	3.59	3.37
Nodes first 25% rate	75	2.08	6.14	1.32	0.41	0.26
	150	2.10	3.21	1.61	0.38	0.13
	225	1.44	3.88	3.40	2.91	2.59

(d) Mean of the deviation table for (a), (b) and (c)

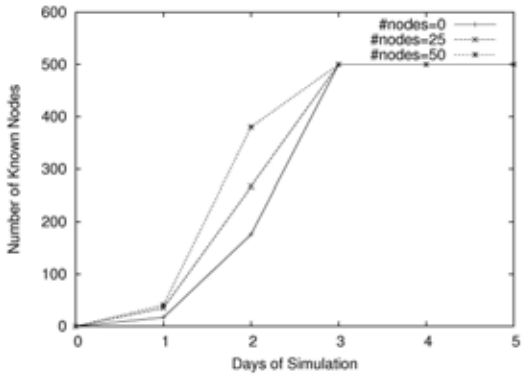
Figure 3.10: Number of known nodes and their levels of trust progress (continued).

messages are not critic to the system performance, although they do provide a considerable improvement.

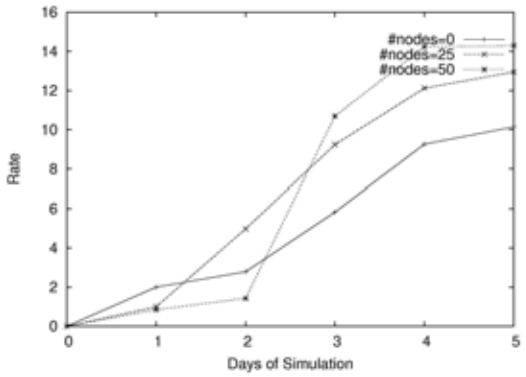
3.12 Conclusions

This thesis presents *Chains of Trust*, a new POI information dissemination scheme that builds a reputation system based on people's traffic patterns. Unlike other solutions presented in sections 2.4 and 2.5, *Chains of Trust* completely relies on the ad-hoc network to function and requires no roadside infrastructure, protects user privacy by allowing users to manage their own identities, it requires no CA, and by keeping users' information distributed among the vehicles in the network and not centralized in a single entity that could be compromised. In addition, it uses POI chains to accumulate POI reviews of the same POI, so that whenever a user follows a recommendation he can update his level of trust in all the reviewers in the chain, therefore increasing the speed at which the reputation system is built.

From the results presented in sections 3.9, 3.10, 3.11 several conclusions can be drawn. First and foremost, *Chains of Trust* performs satisfactorily in a realistic scenario by rapidly building a *Web of Trust* among its users, even for low reviewing frequencies. Secondly, the length of POI chains is relevant in terms of the number of nodes a user gains information of when verifying a POI review. However, regardless of the length, the mean of the known nodes level of trust remained similar, hence indicating that it does not help to improve the trustworthiness of those nodes. Finally, user chains do help improve the development of the *Web of Trust* once a primary structure of known nodes has been established.



(a) Number of known nodes: mean of the number of known nodes by every node.



(b) Rate or level of trust of the 25% most trusted nodes: mean of the rates users assign to other users as POI reviewers (only for the 25% highest rated nodes).

Mean of the deviation σ						
Category	#Nodes	Day 1	Day 2	Day 3	Day 4	Day 5
Nodes first 25% rate	0	0.94	1.90	3.49	4.46	4.21
	25	0.97	4.39	2.22	2.15	1.58
	50	1.78	1.99	3.89	0.98	0.94

(c) Mean of the deviation table for (b)

Figure 3.11: Number of known nodes and their levels of trust progress.

Chapter 4

poiSim: the Simulation Tool

Once *Chains of Trust* had been defined, it needed a realistic simulation tool to estimate its success in the real world. Simulation tools like Glomosim or ns-2 were discarded because in order to simulate hundreds of thousands of nodes they require a massive amount of memory. Thus, we were inclined to design our own simulation tool. Like in [88], it was decided to analyze the realistic vehicular trace produced by the *Multi-Agent Traffic Simulator* (MMTS) developed by K.Nagel at ETH Zurich. All in all, with over 260.000 simulated nodes or vehicles in an area of around 250 km x 260 km, this mobility trace suited the simulation needs.

According to the data in [108] and [109] there were 4.012.690 passenger vehicles registered in Switzerland in the year 2008, which means that poiSim simulates only 6,48% of them. Even if we took into account the number of registered vehicles which are not used, we believe the simulated adoption rate would still be considerably low.

4.1 General Description

In section 1 it was described how the scheme relies on people's habits in order to construct a *Web of Trust*. The main goal behind designing a specific simulator is to discover if those habits suffice to ensure the application success in a real life scenario. If so, passed the first several days each node in the network should have several Point of Interest (POI) reviews as well as known nodes. *poiSim* will also be used to analyze how the system behaves when modifying several parameters, e.g., the length of POI chains, or to study how it performs when user chains are not used. It should be noted that *poiSim* is a high level simulator, i.e., it simulates

Chains of Trust but it does not simulate a *Medium Access Control* protocol for example, it would be unfeasible to simulate wireless communication realistically for hundreds of thousands of nodes in a reasonable amount of time. That is why a separate experiment was conducted in section 3.8. In addition, the version of *Chains of Trust* simulated by *poiSim* will be slightly different from the original scheme designed in the previous section. The differences will allow the simulator to improve its performance and will not affect the results. They will be explained in this section.

These are several of *poiSim*'s features: it simulates 259.977 nodes and 15.000 POIs. Every node stores:

- Levels of trust on 500 other vehicles.
- 100 unverified POI chains with 225 POI reviews each.
- 150 verified POI chains with 225 POI reviews each.

And for every POI:

- 5000 reviews are stored in the system.

Every POI is assigned a random value ranging from 0 to 15 to be its real rate μ. The rates the users assign to those POIs will be normally distributed around μ with variance $\sigma^2 = 2$ (as depicted in Fig. 4.1).

Communication wise, a range of 120 meters of coverage is considered and every time a vehicle transmits all the vehicles within range receive the message. There are two kinds of simulated messages: periodic and recognition.

1. Periodic Messages: every 120 seconds a vehicle will broadcast a message with his 25 highest rated verified POI chains, adding unverified POI chains to complete the message if necessary.
2. Recognition Messages: every time a vehicle recognizes another as a trusted user it will send his 25 highest rated verified POI reviews and his 25 most trusted nodes, together with his level of trust in them. Unverified POI chains may be included as well to complete the message if necessary.

It should be noted that *poiSim* does not include request messages, as the original scheme did. The reason is that their implementation would not have changed the simulation experiments, since they are seldom used in respect with periodic messages (every 120 seconds). As a result, the quality of the system is measured by

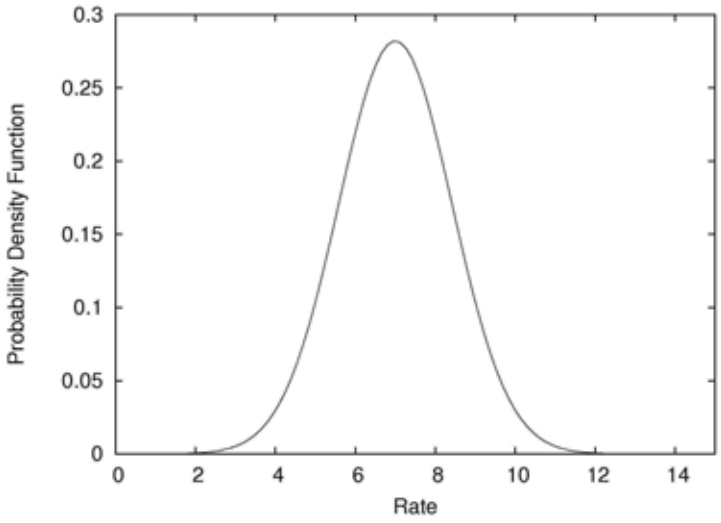


Figure 4.1: User's rate distribution for the real rate $\mu = 7$ and $\sigma^2 = 2$

the number of nodes known to every user and the number and length of verified POI chains every user stores at the end of every simulated day. Due to computational limitations, it was decided not to simulate user chains as explained in sections 3.3, 3.4 and 3.6.2, since their simulation would have required the execution of Alg. 4 for each of the 260.000 simulated nodes and each of the 225 reviewers that unverified and verified chains store. Had user chains not been removed, they would have added a large overhead on the simulation, thus extending the simulation execution time to weeks or even months. As a result, *poiSim* does not simulate misbehavior (since it would not be possible to penalize bad recommenders), hence there is no need to keep track of the recommender-recommended relation and its rewards and penalties policy. We believe misbehavior attempts will be dealt with the mechanisms described in section 3.7 and will not affect the overall performance. User chains still exist and are transmitted in recognition messages, but the recommender will not be rewarded nor penalized for it. In addition, recommended nodes will be added to the known nodes list with a trust level of 1. In this way, it can be established if the scheme performs satisfactorily or not, and if it does it can be safely assumed that the implementation of user chains will be an improvement, since recommended nodes will be added to the known nodes list with the level

of trust with which they were recommended (always greater or equal than 1). It should also be noted that our simulation only contains one POI category, which is enough for the desired testing purposes.

Algorithm 4 User chains algorithm

```
//every time a POI is reviewed by the user
for every POIReview
  //for every reviewer in the POI's unverified chain
  for every unverifiedChain.reviewer
    //if the user and the reviewer's opinion are too different
    if |unverifiedChain.reviewer.rate − user.rate| ≤ ΔOp
      //update the level of trust in all the nodes recommended by the reviewer's
recommender
      reviewLevelOfTrust(unverifiedChain.reviewer.recommender)
    end if
  end for
end for
```

In a nutshell, *poiSim* processes each line of the MMTS trace, which contains a $node_{ID}$ and its corresponding x, y, z, t coordinates and updates the vehicles position. On every update it ensures that the vehicles send a periodic message every 120 seconds, which is a long enough period to avoid causing a tracking vulnerability, and a recognition message when needed. In addition, once a day at most each user reviews a randomly chosen POI from his unverified POI chains, or a completely random POI if there are no unverified POI chains available, as described in Alg.5.

In order to better study the system, to observe how the POI reviews are exchanged between users, how users build a better reputation for themselves and the effect of several configuration parameters on the simulation, such as the chain length or the number of user reviews in a user reviews message, the 24 hours vehicular trace is replayed to obtain a multiple days scenario. It should be remarked that the only common element in every simulated day will be the MMTS trace, because the POIs being reviewed are randomized, and hence will be different in every run.

Algorithm 5 POI review algorithm

```
if node.reviewedPOIToday() = false then
  node.setReviewedPOIToday(true)
  if random(0, 1) = 1 then
    if node.unverifiedReviewsTable.isEmpty() = false then
      reviewPOIUnverifiedTable()
    else
      reviewPOIRandom()
    end if
  end if
end if
```

4.2 Design Overview

The simulator has been designed with efficiency in mind, with the emphasis placed on memory rather than on reducing the computing time. The reason for this order in priorities is simple if illustrated with an example. Every simulated node must have a unique identifier and a level of trust, the first ranging from 1 to 260.000 and the second from 0 to 15. In order to store the $node_{ID}$ the simulator is going to use a 4 bytes integer, since 2 bytes fall short, and to store the level of trust a single byte will suffice. However, when memory alignment is taken into consideration that single byte turns into 4 (or even 8, depending on the architecture). As a result, every node is now 8 bytes long. Looking at the bigger picture, every node stores the level of trust of 500 other nodes ($8 \times 500 = 4.000$) and over 260.000 nodes are simulated (260.000×4.000 =1.040.000.000). Had both fields been stored in the same 4 bytes integer it would have been possible to save half that space. It is of paramount importance to grasp the magnifying effect of changes deep into the structure of the simulator. Certainly, by using the same region of memory for both fields every time they are accessed an additional operation will need to be performed to separate them, which will increase the access time; the alternative, however, is not being able to run the simulation with average computational resources. Fig. 4.2 provides a clear depiction of *poiSim*'s logical components and processes. Basically, there is a thread that reads the mobility trace from a disk, block by block, and places it in a double buffer from which another thread feeds on. Those blocks are processed line by line, which are of the form $node_{ID}$, x, y, z, *time*, *command*. During that processing, the *command* dictates if a node is created, destroyed or updated. Besides, based on the *time* the simulator checks

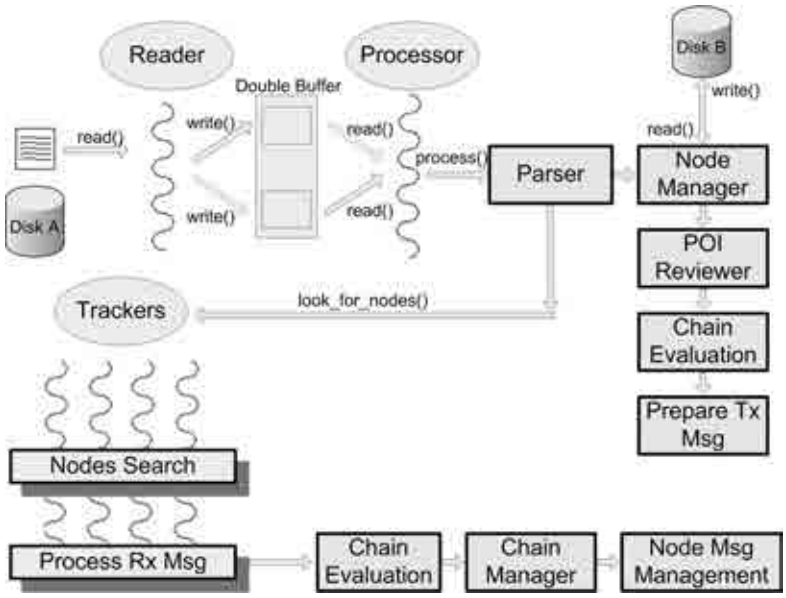


Figure 4.2: System processes map

if the node should review a POI or prepare a message to be transmitted. If the node needs to send a message a group of threads is notified to look for nodes in range and process the received message, if any are found. All of these processes will be extensively detailed in the following sections. However, before any further explanations, it should be remarked that even though memory management was our first priority, we were also able to take full advantage of the multi core CPU at our disposal, by dividing tasks into independent sub-tasks and implementing them in multiple threads so that they could be parallelized.

4.3 Memory Snapshot

The mobility trace being used largely determined the memory structures depicted in Fig. 4.3, and that is the reason why its understanding was so important. That trace simulates the traffic patterns of 259.978 vehicles over 24 hours. In that period of time trips began and were ended, hence not all vehicles were traveling at the same time. Several tests were performed to study the trace and concluded that 55.197 is the maximum number of vehicles traveling at the same time. As a result, the simulator is designed to store active vehicle's information in memory while the rest is kept on disk. It should be noted, that the exact number of nodes allocated

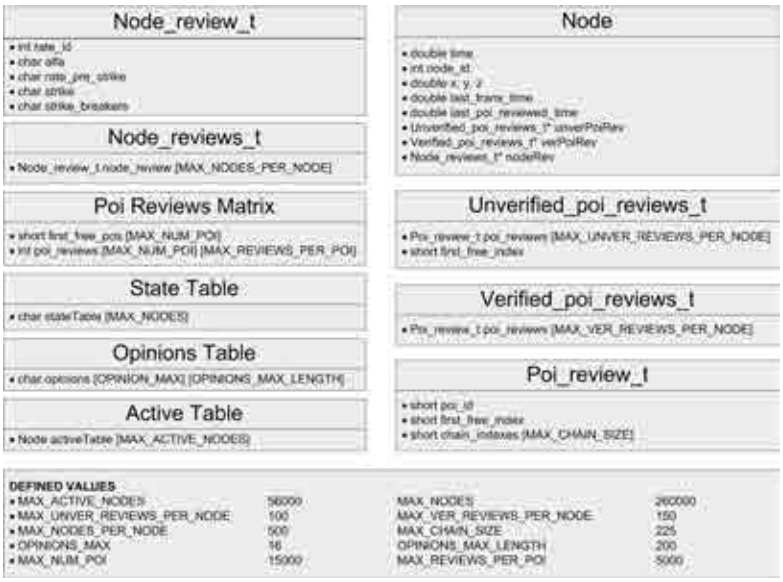


Figure 4.3: Memory map

in memory is slightly larger (56.000) to account as well for the simulator's internal operations so that vehicles can be moved in and out of memory as required by the trace without dragging down the performance.

The second improvement is derived from carefully examining the goal of the application and it affects the way POIs are stored in chains. The main objective is the dissemination of POIs information over the network, and that information translates into POI chains which in their turn are an aggregate of POI reviews. In other words, many nodes will have common parts of POI chains, i.e., repeated POI reviews since what it is trying to accomplish leads to the repetition of information. Therefore, a matrix is designed to hold every review ever created in the system (*Poi Reviews Matrix* in Fig. 4.3) and instead of storing the reviews in the nodes, they only store the indexes to the matrix. That allows the system to save half as many bytes for every repeated review.

In addition, the system is designed to avoid the extra bytes lost to memory alignment, when possible, by grouping pieces of information together. This technique was used in the *Poi Reviews Matrix* to store a user identifier and the rate he assigned to a POI and in the *Node_reviews_t* table of every node to store levels of trust and user identifiers. We would like to remark that the identifiers and the rates (or levels of trust) are fields that had they not been grouped together, they would have been accessed sequentially. Therefore, the extra selection operations

Memory Analysis	
Structure	Size
Node	80 bytes
Unverified_poi_reviews_t	45402 bytes
Verified_poi_reviews_t	68100 bytes
Active Table (56000 nodes)	6.58 GB
Poi Reviews Matrix	300 MB
Opinions Table	3200 bytes
State Table	56000 bytes
Simulation data produced in each simulation (260000 nodes)	31 GB

Table 4.1: Size of the memory structures used by *poiSim*

are compensated by one less access to memory. It should also be noted that the *Node_reviews_t* table always has to be ordered by the level of trust (so that the most trusted nodes can be easily found and sent in recognition messages), hence the importance of allocating the rate in the first byte and the identifier in the lower three. Given two values a and b, $a > b$ if and only if $a.highest\_byte > b.highest\_byte$. Hence, the ordering operation can be performed disregarding the fact that those bytes contain different bits of information.

The result of those optimizations is displayed in Table 6.4. It should be noticed that the simulation uses over 6.9 GB of main memory (between *Active Table* and *Poi Reviews Matrix*) and produces a volume of 31 GB of data in disk at the end of the simulation, which contains the state of each individual vehicle when it is not traveling.

As far as memory initialization is concerned, it is performed at the beginning of the simulation, even before it starts to process the mobility trace. Most of the memory is allocated dynamically (unverified, verified POI tables and nodes tables) while the rest of the system is stored in static memory. However, nodes are not allocated and freed every time they are created and destroyed. The *Active Table* allocates dynamic regions when it is created and until the simulation finishes it does not free them, mainly to avoid memory fragmentation.

4.4 Processing the MMTS Trace

Once the memory has been allocated the simulator can begin reading the trace. The file is read in blocks of 8192 bytes by a thread that copies them into a double buffer. On the other side of the buffer another thread feeds on those blocks and processes them. The idea is to minimize the wait of the *Processor* thread on retrieving the data from disk by having another thread perform the task, while at the same time keeping them both synchronized so that every block is processed. To that end the double buffer is protected with what in pthreads notation are called *Condition Variables* which is a combination of signals and mutexes: before writing or reading a block from or of the memory structure each thread tries to acquire a lock, if unsuccessful it blocks until the current lock owner sends him a signal to indicate that the lock has been released.

```
/* Memory definitions */
#define DISK_BLOCK_SIZE 8192

struct disk_double_buffer{
char bufferA[DISK_BLOCK_SIZE];
char bufferB[DISK_BLOCK_SIZE];
short dataReadyA;
short dataReadyB;
};

struct disk_double_buffer exchange_buffer;
pthread_mutex_t bufferA_mutex, bufferB_mutex;
pthread_cond_t bufferA_cond, bufferB_cond;

/* readerThread.c - fills the buffer with blocks */
res = fread(block,1,DISK_BLOCK_SIZE,fd);

if(res > 0)
{
  pthread_mutex_lock(&bufferA_mutex);
  if(exchange_buffer.dataReadyA != 0)
  {
    pthread_cond_wait(&bufferA_cond,&bufferA_mutex);
  }
  memcpy(exchange_buffer.bufferA,block,res);
  exchange_buffer.dataReadyA = res;
```

```
  pthread_cond_signal(&bufferA_cond);
  pthread_mutex_unlock(&bufferA_mutex);
}

/* managerThread.c - processes the buffer */

pthread_mutex_lock(&bufferA_mutex);
if(exchange_buffer.dataReadyA == 0)
{
  pthread_cond_wait(&bufferA_cond,&bufferA_mutex);
}
      ...

processBlock(exchange_buffer.bufferA,exchange_buffer.dataReadyA
   );
exchange_buffer.dataReadyA = 0;
pthread_cond_signal(&bufferA_cond);
pthread_mutex_unlock(&bufferA_mutex);
```

Listing 4.1: Code excerpt to illustrate how the double buffer works

The *Processor* thread reads the block line by line, translating each and every line into a simulated step. Each of those steps indicate the simulator that one of the following events has occurred: a trip has began, a vehicle's position has changed or a trip has come to an end. In the *Node Management* phase the *Processor* becomes responsible for the interpretation of those instructions, i.e., it has to create and destroy nodes as the trace dictates, bringing them from memory to the *Active Nodes* table and back to memory once the trip finishes, besides updating their position when needed. To speed up the process of looking for nodes in the table a Dictionary was implemented using the $node_{ID}$ mod MAX_ACTIVE_NODES as key. As in any other dictionary, the idea is to check if the *key* position is empty; otherwise move forward to the next one and retry. Notice that, as depicted in Fig. 4.2, the trace and the nodes are stored in separate disks in order to minimize the access latency.

While updating the position and the time of the vehicle the simulator checks if the user has to review one of his POIs, and if so the thread enters the *POI Reviewer* phase. In this step of the simulation it has to select a POI_{ID} to be reviewed, which can either be accomplished by randomly choosing one of the unverified chains stored in the node or by randomly generating an identifier if no chains are available. Should that last option be the case things simplify considerably, as

detailed below.

- Random POI: it needs to create a review in the *Poi Reviews Matrix* and a new verified chain in which to store that review.

- Unverified POI chain: it needs to create a review in the *Poi Reviews Matrix* too, although in this case this is just the beginning of the process. In the *Chain Evaluation* phase it compares that review with the other reviews in the chain and increase or decrease the node's level of trust on the reviewers based on how much their opinions or rates differ. This rewards and penalties policy follows the process previously described in section 3.6. Notice that whenever the nodes level of trust is modified the *Node_reviews_t* table needs to be reordered, which as described in section 4.3 can be done disregarding the fact that two pieces of information are stored in that region of memory.

Finally, the *Processor* thread verifies if it is time for the user to transmit information to the network. If so, it prepares the messages, otherwise the processing of that parsed line finishes here. *poiSim* simulates two kinds of messages:

1. Periodic messages are made of 50 POI chains, the highest rated among the verified POI chains the node stores. Should there not be enough, unverified chains will be selected.

2. Recognition messages are made of the highest rated 25 POI chains and 25 Node reviews. Like in periodic messages, verified POI chains can be complemented with unverified chains.

Both messages will be prepared and depending on the situation the receiving node will select one or the other.

Once finished with the preparations, the *Processor* thread signals the *Trackers* threads to wake up. The *Active Table*, where all the active nodes are stored, is partitioned into 4 equal portions (one for each thread) and processed by the *Trackers*, which search for nodes in range. When a node is found the thread processes one message or the other depending on if the receiver previously knew the sender. This is why it was of paramount importance that everything was prepared beforehand, had it not been done that way each time a vehicle in range was found its thread would have had to look for the information instead of processing it directly from the message.

Since the messages contain different kinds of information, different paths will be followed when processing them.

- POI chains: when a POI chain is received it is marked as unverified and the thread looks for its POI_{ID} into the node's tables. If it is not found then the received chain is stored in the unverified table. If it is found and the POI chain has not yet been verified, then both chains are merged. Otherwise, if it receives a chain for a POI that it has already reviewed then it reviews the received chain assigning rewards and penalties to the reviewers, just as it was done in the *POI Reviewer* phase, and merge the chains storing them in the verified table (*Chain Management*).

- Node reviews: a node review is a $node_{ID}$ and a level of trust assigned to that node by the sender. The receiver of the message treats those reviews as if it they were his own with two conditions:

  1. the recommended level of trust for a certain node can never be greater than the level of trust the receiver has on the sender.
  2. the recommended level of trust is always decreased by 1, to signify one link in the chain of trust.

Finally, when the *Trackers* have finished processing all active nodes they signal back the *Processor* and the cycle can begin again.

4.5 Hardware Requirements

poiSim was executed in a PC running 64 bits Linux Fedora 12, with the following hardware specifications:

- Quad Core CPU Q6600 at 2.40 GHz, with 128 KB of L1 cache and 8 MB of L2 cache.

- 8 GB of DDR2 ram memory at 887 MHz with latencies 5-5-5-15 (tCL-tRCD-tRP-tRAS)[1]

[1]

- tCL: column address strobe (CAS) latency; the number of clock cycles required to access a specific column of data. (The initial t refers to time.)
- tRCD: row address strobe (RAS)-to-CAS delay; the number of clock cycles needed between a row address strobe and a column address strobe.
- tRP: RAS pre-charge; the number of clock cycles needed to close one row of memory and open another.

- 32 GB SSD disk to store the OS and the mobility trace.
  - 64 MB onboard cache.
  - Read maximum performance: up to 210 MB/s.
  - Write maximum performance: up to 75 MB/s.
- 96 GB SSD disk to store the simulation data.
  - Read maximum performance: up to 285 MB/s.
  - Write maximum performance: up to 275 MB/s.
  - Sustained write performance: up to 250 MB/s.

With this hardware, a simulation of the 24 hours vehicular trace lasts approximately 120 minutes.

4.6 Message Formats

In this section, the message formats and sizes for the simulation will be defined according to the scalability results presented in section 3.8. Considering the following format for a periodic message M as defined in section 3.2:

$$R = \{ \underbrace{Id}_{88 \text{ bytes}} || \underbrace{Rate}_{1 \text{ byte}} || \underbrace{Timestamp}_{8 \text{ bytes}} \} \tag{4.1}$$

$$R' = \{ \underbrace{H(Id)}_{8 \text{ bytes}} || \underbrace{Rate}_{1 \text{ byte}} || \underbrace{Timestamp}_{8 \text{ bytes}} \} \tag{4.2}$$

$$M = \{ \underbrace{R_1}_{97 \text{ bytes}} || \underbrace{\{H(R_1)\}_{K_{priv_{Node\ 1}}}}_{17 \text{ bytes}} || \underbrace{\{R'_2\}_{K_{priv_{Node\ 2}}}}_{17 \text{ bytes}} ||...|| \underbrace{\{R'_n\}_{K_{priv_{Node\ n}}}}_{17 \text{ bytes}} || \underbrace{K_{pub_{Node\ 1}}\}}_{128 \text{ bytes}} ||...|| \underbrace{K_{pub_{Node\ n}}\}}_{128 \text{ bytes}} \tag{4.3}$$

Taking into account that the total amount of information has to be approximately 400.000 bytes, information about 25 POIs will be sent, each containing

– tRAS: the number of clock cycles needed to access a specific row of data in RAM.

107 user's reviews adding up to a total of 390.300 bytes. It should be noted that periodic messages are fragmented in a 1000 bytes packets including certain redundancy, so that if a packet is lost the rest of the message can still be read.

Recognition messages will also contain an user reviews message M':

$$R_{Node\ i} = \{\underbrace{K_{pub_{Node\ i}}}_{128\ \text{bytes}} || \underbrace{Level\_of\_Trust_{Node\ i}(as\_POI\_reviewer)}_{1\ \text{byte}}\} \tag{4.4}$$

$$M' = \{\underbrace{R_{Node\ 1}}_{129\ \text{bytes}} || ... || \underbrace{R_{Node\ n}}_{129\ \text{bytes}} || \underbrace{Timestamp}_{8\ \text{bytes}} || \underbrace{\{H(R_{Node\ 1}||...||R_{Node\ n}||Timestamp)\}_{K_{priv_{Sender}}}}_{10\ \text{bytes}} || \underbrace{K_{pub_{Sender}}}_{128\ \text{bytes}}\} \tag{4.5}$$

Considering that M' contains information about 25 users the message size amounts to 3.371 bytes.

4.7 Conclusions

In this section we have discussed the design and implementation of the application simulation tool *poiSim*. Unlike state of the art network simulators like OPNET, QualNet, OMNet++, ns-2, ns-3 or GloMoSim, *poiSim* is capable of simulating a 24 hours trace containing almost 260.000 vehicles in approximately 120 minutes. We have shown that by separating the communications from the application layer it is possible to build an application simulator which can execute simulations in the order of hundreds of thousands of nodes. This approach will yield more realistic results than using network simulators to simulate both the communication and application layers, as done in the vast majority of research articles [30, 31, 32, 33].

Chapter 5

Anonymous Chains of Trust

Chains of Trust, as explained in section 3, may allow an attacker to profile users and even link their public key to their real identity. If an attacker positioned himself at a frequented crossroad, given enough time he would be able to gather a large number of Point of Interest (POI) chains which he could analyze to know in what area every user lives, what habits they have or even know who they are. This is not a problem exclusive of *Chains of Trust*, this is a problem inherent to reputation systems, where anyone can gather all the recommendations or reviews made by a user. In *Anonymous Chains of Trust* we propose a solution to address this vulnerability.

5.1 General Overview

In this section we propose a new mechanism to preserve users privacy based on identity borrowing. In a reputation system, if two nodes trust each other it is because they both have had similar views or opinions on the information they have shared in the past. Particularly in *Chains of Trust*, if two users trust each other it is because they have similarly rated POIs in a given category and therefore have similar tastes. Since their rates for a certain POI category are similar, one user A can ask another B, who he trusts, to issue a review message with a certain rate for a certain POI with B's own identity, much like if he had reviewed the POI himself. For all intents and purposes, A will be borrowing B's identity for that single review.

Fig. 5.1 depicts in detail how the system works. Steps 1 and 2 are the first part of the protocol where user A requests user B to issue a POI review on his behalf.

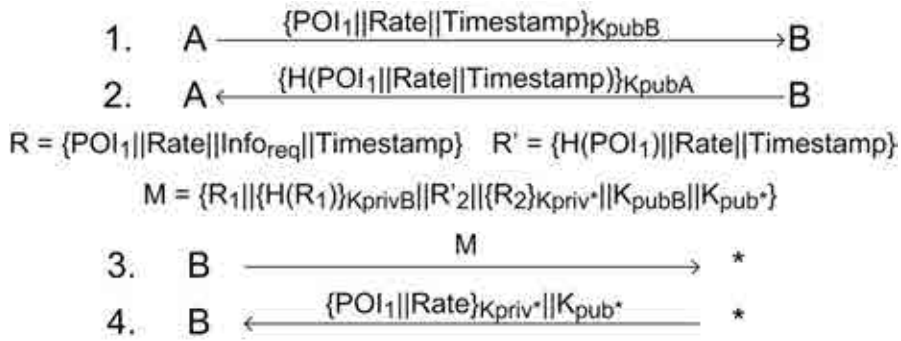


Figure 5.1: *Anonymous Chains of Trust.*

The second part, steps 3 and 4, allows B to determine how reviewing that POI on behalf of A affects his reputation.

User A reviews POI_1 and the system decides to request another node to issue the public review on his behalf. How often the system asks a user to review a POI on behalf of another is based on the system parameter α. Whenever A meets one of his trusted nodes (B) he sends a review request containing the POI identifier (POI_1), the rate he assigns to that POI and a timestamp, everything encrypted with B's public key (as we can see in step 1). In addition, he signs B's K_{pub} with his own K_{priv} so that B can verify A's identity. In step 2, if B recognizes A as one of his trusted users he acknowledges the reception of the message by sending the hash H of the received message encrypted with A's public key K_{pub_A}. Should that acknowledgement not reach A the system on A's vehicle will request the review to another trusted user. Once B has accepted to review POI_1 on A's behalf, he will include this review in the list of messages he transmits periodically.

In step 3, B prepares a periodic message M containing a chain of reviews of length 2 for POI_1, which includes the review R_1 he is issuing on behalf of A and another review R_2 he has received for that same POI from another user. For the sake of clarity, in this example M contains information about just one POI, i.e., one POI chain, although in reality periodic messages may include several concatenated chains for different POIs. Once that information is compiled, B's vehicle broadcasts it to the network. It should be noted that R, which is the first element of a chain of POI recommendations, contains a field named $Info_{req}$. This bit-field will be set to let the message receivers know that B would like to receive their reviews of that POI. In step 4, the receivers of M reply with their own rate for the requested POI. B will store this information, and once he has gathered enough data he will evaluate the review A sent to him and adjust his level of trust

in A accordingly (as explained in section 5.2).

It should be noted that nodes that receive the information request will reply with their own reviews, with reviews from trusted nodes or with reviews they have issued on behalf of other nodes. If they were only allowed to reply with their own reviews, an attacker would only need to broadcast a POI request for multiple POIs and gather all the information to profile the users.

Incidentally, in order to minimize the repetition of information in M, the POI identifier is only used in the first review of a chain of recommendations (R), while the rest use instead the hash H of that identifier (R').

The idea behind this scheme is that if enough users request their trusted fellows to review POIs on their behalf, then a user's individual identity is hidden by the identities of all the users he trusts. As a result, even an all-knowing attacker will not be able to profile individual users because he will have no way of knowing the identity of the real POI reviewers. This concept of privacy is somewhat similar to what group signatures provide [8, 39, 50], although without the overhead of specifically creating and managing a group.

Generally speaking, in a group signature scheme every user is part of a group, either preset or dynamically created, and every group has a group manager in charge of making public the information gathered by group members. In addition, the group manager needs to monitor the group members for misbehavior and evict them from the group if they misbehave.

5.2 Evaluation of Identity Borrowing

As seen in section 5.1, user B needs a mechanism to determine the impact that reviewing a POI on behalf of A has on his reputation. Whenever a user reviews a POI on behalf of somebody else he sets the $Info_{req}$ bit in the chain of reviews for that POI in the periodic message. After having gathered n reviews from other users (or if the time passed since he issued the review reaches a certain value $T_{evaluation}$) B evaluates A's review.

Let us define n as the number of reviews sent by different users regarding a certain POI POI_1, $U_1, ..., U_n$ as the users who sent their POI review and $\hat{U}_1, ..., \hat{U}_n$ as the subset of those nodes known by the user B, χ_{POI_1,U_1} as the rate that U_1 gave to POI_1 and $\lambda_{\hat{U}_i}$ as the level of trust that B has on U_i as a POI reviewer.

Then the POI consensual grade G is defined by:

$$G = \sum_{i=1}^{n} \left(\chi_{POI_1,\hat{U}_i} \cdot \frac{\lambda_{\hat{U}_i}}{\sum_{j=1}^{n} \lambda_{\hat{U}_j}} \right) \tag{5.1}$$

It should be noted that the rates assigned by unknown nodes are ignored as long as there is a known reviewer in the chain. Otherwise, the chain's rate is the arithmetic mean of the POI rates assigned by the unknown reviewers. Similarly, the reviews of the less trusted known nodes are ignored when there is a known node that belongs to the group of B's most trusted nodes (MTG).

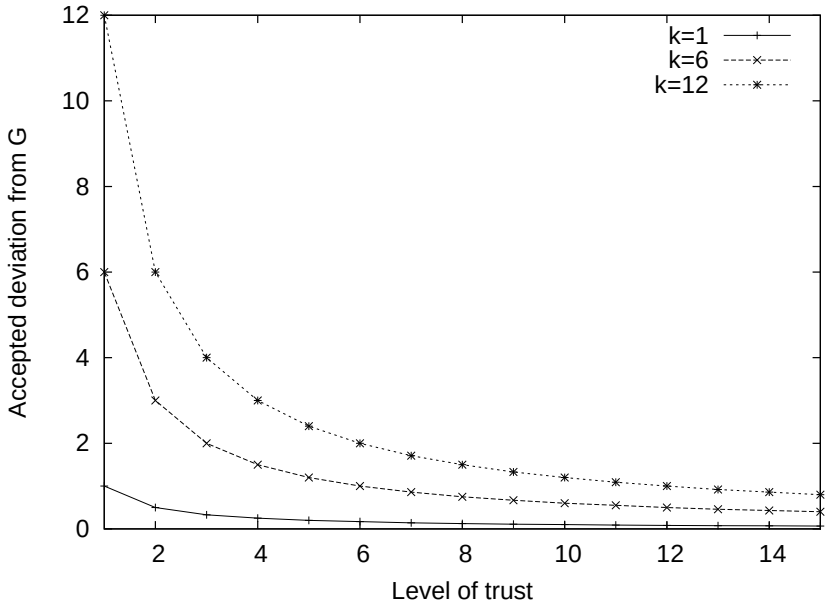


Figure 5.2: Progression of k/λ_A for different values of k.

Once B knows the value of G, he expects the rate A sent in his review of POI_1 to be:

$$G - k/\lambda_A \leq \chi_{POI_1,A} \leq G + k/\lambda_A \tag{5.2}$$

where k is a parameter defined by each user depending on how strict he wants to be when lending his reputation. k can take any value considering that $G +$

$k/\lambda_A \leq 15$, 15 being the maximum value for a node's reputation in the system, and $G - k/\lambda_A \geq 0$, 0 being the minimum. If $\chi_{POI_1,A}$ falls outside the limits defined by (5.2) then B will stop transmitting A's review and the level of trust B has on A, i.e. λ_A, will decrease by half its value.

It should be noted that too high values of k will allow misbehaving users to take advantage of the system and ruin the reviewer's reputation in the network. On the other hand, too low values will in all likelihood unfairly decrease the level of trust B has in A. Regardless of the value assigned to k, in Fig. 5.2 we can see that the allowed deviation from G decreases for high levels of trust between users. This responds to the fact that users with high levels of trust assign the most similar rates to the same POIs, and that should still be true when a user is lending his identity.

In the same way that B needs to make sure that A is not lying to him, A needs to know if B is really transmitting a review on his behalf. To that end A examines the periodic messages he receives looking for a chain of recommendations for the requested POI POI_1. If he does not find it after a certain time $T_{request}$, A will request the review of POI_1 to another of his trusted nodes. The level of trust that A has in B does not need to be decreased because A's reputation in the network was not damaged by B's inaction.

5.3 Scalability Analysis

In section 3.8 we determined with a ns-3 [98] simulation that in a 400 vehicles scenario such as the one depicted in Fig. 3.6, every user can broadcast 400 packets of a 1,000 bytes every 120 seconds yielding a 91.5% rate of successfully received packets. It should be noted that in our system every node broadcasts periodic messages to be received by all nodes within 1 hop distance.

The periodic message used in *Chains of Trust* has been modified to include the changes described in section 5.1 with the goal of achieving a reception rate still over 90%. Considering the following format for a periodic message M as defined in Fig. 5.1:

$$R = \{\underbrace{POI_{Id}}_{88\text{ bytes}} || \underbrace{Rate}_{1\text{ byte}} || \underbrace{Info_{req}}_{1\text{ bit}} || \underbrace{Timestamp}_{8\text{ bytes}}\} \tag{5.3}$$

$$R' = \{\underbrace{H(POI_{Id})}_{8\text{ bytes}} || \underbrace{Rate}_{1\text{ byte}} || \underbrace{Timestamp}_{8\text{ bytes}}\} \tag{5.4}$$

$$M = \{ \underbrace{R_1}_{97 \text{ bytes}} || \underbrace{\{H(R_1)\}_{K_{priv_{Node\ 1}}}}_{17 \text{ bytes}} || \underbrace{\{R'_2\}_{K_{priv_{Node\ 2}}}}_{17 \text{ bytes}} \\ ||...|| \underbrace{\{R'_n\}_{K_{priv_{Node\ n}}}}_{17 \text{ bytes}} || \underbrace{K_{pub_{Node\ 1}}}_{128 \text{ bytes}} ||...|||| \underbrace{K_{pub_{Node\ n}}}_{128 \text{ bytes}} \} \quad (5.5)$$

Taking into account that the total amount of information has to be approximately 400.000 bytes, information about 25 POIs will be sent, each containing 107 user's reviews adding up to a total of 390.303,125 bytes. It should be noted that periodic messages are fragmented in a 1000 bytes packets including certain redundancy, so that if a packet is lost the rest of the message can still be read.

In addition, in order to avoid flooding the network when users reply to a POI information request, it will only be allowed to set the $Info_{req}$ bit for a maximum of 5 POIs in a message M.

$$POI_{resp} = \{\underbrace{\{POI_{Id}||Rate||Timestamp\}_{K_{priv_S}}}_{97 \text{ bytes}} || \underbrace{K_{pub_S}}_{128 \text{ bytes}} \} \quad (5.6)$$

In the best case scenario every user will have information of all 5 POIs and reply with POI_{reqp}, a 1125 bytes message.

5.4 Experiments

Once the system has been defined we need to determine how it will perform in a realistic scenario. To that end, we have modified the simulation tool *poiSim* to simulate *Anonymous Chains of Trust*.

In *Anonymous Chains of Trust* whenever a user reviews a POI the system needs to choose between: (i) broadcasting that review,i.e., making it public, and (ii) waiting until the user's vehicle recognizes a trusted node and asking him to review that POI on his behalf. As explained in section 5.1, this decision depends on the system parameter α. In the early stages of the application deployment, that delay can hamper the development of the *Web of Trust* between users. Determining the degree to which the system deployment is affected is our main goal.

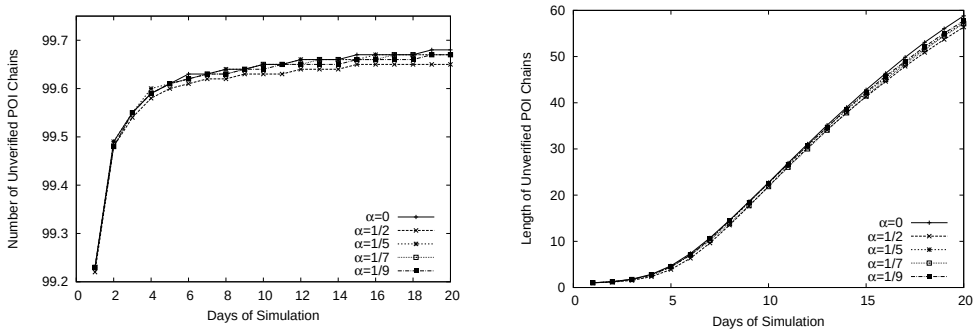
In this experiment, every user inputs a new review into the system every 5 days and we study different values for α: a user requests another user to review a POI on his behalf once every 2 reviews ($\alpha = 1/2$), 1 review of every 5 ($\alpha = 1/5$), 1 review

of every 7 ($\alpha = 1/7$), 1 review of every 9 ($\alpha = 1/9$) and a control sample where users do not review POIs on behalf of other users ($\alpha = 0$). The reviews or rates users assign to POIs are between 0 and 15 and follow a normal distribution with mean 7 and $\sigma = 2$. The evaluation of user misbehavior is outside the scope of this simulation. It should be noted that the real measure of the system performance is given by how many users every user knows and how much he trusts them, because (i) the more users he knows the more information he has to choose a truthful recommendation from and (ii) the more users he knows the more users he can ask to review a POI on his behalf and make his identity harder to discover.

In Figs. 5.3a and 5.3b the evolution of the number and length of unverified POI chains can be seen. After the first 5 days of simulation the number of unverified chains and its length is very similar regardless of the reviewing rate. The fact that the average number of unverified chains is over 90 (the simulator can store up to 100) and its length is approximately 5 (considering any of the α's) means that there has been interaction between the users and some have already started to build a better reputation in the network. Moreover, considering the results after 20 days of simulation it can be seen that they do not differ significantly.

As far as verified chains are concerned, in Fig. 5.3c the direct relation between the reviewing rate and the number of verified chains the nodes store can be observed. After 20 days of simulation it can be observed that difference between a $\alpha = 1/5$ and the control group with $\alpha = 0$ is almost 1, increasing to almost 2.5 for $\alpha = 1/2$. Overall, the more often a user request another user to review a POI on his behalf the lower his number of verified chains will be, which is logical considering that highest request frequencies introduce a greater delay to information transmission. Fig. 5.3d shows the mean of the length of verified POI chains. It can be observed that is very similar to 5.3b, which is natural considering that every time a POI is reviewed its unverified chain moves on to the verified state. Regarding the rate assigned to the POIs in the verified chains, in Fig. 5.3e it can be observed that the rate of the reviewed POIs varies until it stabilizes around 7, which is expected since the randomly chosen rates are distributed around that value, as previously described in this section. The different simulated values for α determine how fast the POI rate converges to 7.

Figs 5.4a and 5.4b present the user reputation results. In 5.4a we can see that after 20 days of simulation, nodes in the control group ($\alpha = 0$) know on average 160 users, while nodes with $\alpha = 1/5$ know approximately 130 users and nodes with $\alpha = 1/2$ know slightly under 100. Regarding the level of trust in those users depicted in Fig. 5.4b, we can say that they are very similar regardless of the value of α, the maximum difference shown by $\alpha = 0$ and $\alpha = 1/2$.



(a) Number of unverified POI chains: for every node the number of unverified POI chains is computed, their mean is the depicted result.

(b) Length of unverified POI chains: for every node the mean of its unverified POI chains length is computed, the mean of those means is the depicted result.

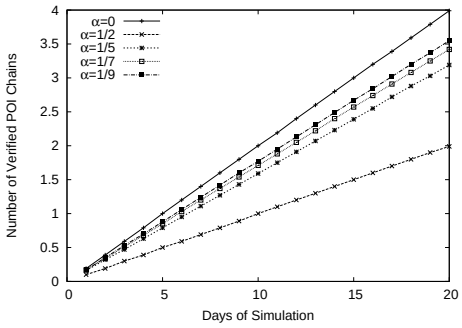
Figure 5.3: Evolution of the length and number of unverified and verified chains.

5.5 Conclusions

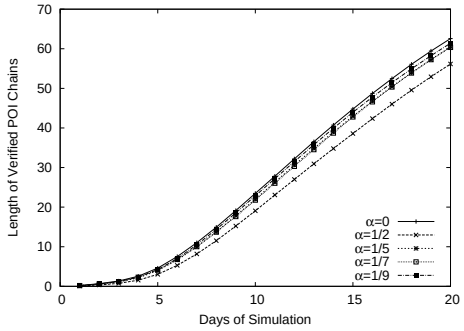
In this section we have presented a novel mechanism to preserve users privacy in a reputation system. By allowing users to borrow each other's identities an attacker can never be sure of who was the real reviewer behind a given POI recommendation. In other words, users that trust each other form a virtual group where any user can use anybody else's identity, thus hiding behind the group. Moreover, this technique should be transparent to the user reputation, since identity borrowing can only occur between users that trust each other, which by definition implies that their reviews for a given POI category are very similar and therefore interchangeable. To the best of our knowledge, this is the first time this technique has been applied to reputation systems.

The results of our simulation tell us that regardless of the value of α we have used (how often a user reviews POI on behalf of another) the length of unverified and verified chains and their rates remains very similar. Regarding the number of users known by every node and his level of trust in them we have shown that even if the known number of users is slightly lower for $\alpha = 1/5$ the difference when compared to the control group is not significant and does not constraint the development of the reputation system. When we compare the control group with $\alpha = 0$ we can start to see a decrease in the system performance (it has a fewer number of verified chains and knows less nodes).

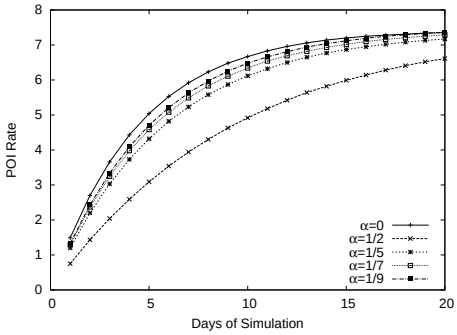
Privacy wise, the fact that after just 10 days of simulation every user knows



(c) Number of verified POI chains: for every node the number of verified POI chains is computed, their mean is the depicted result.

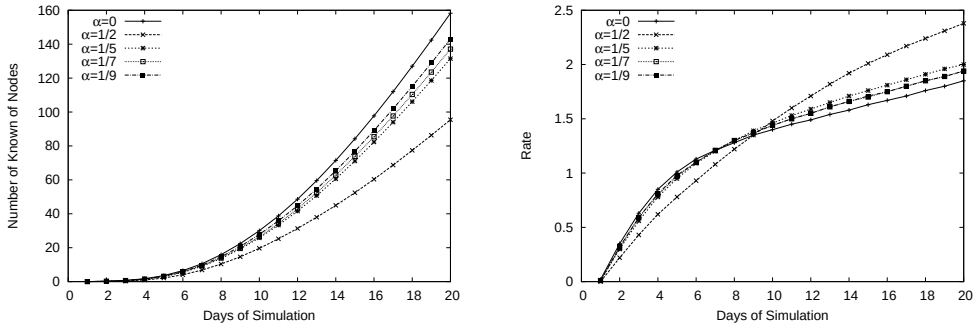


(d) Length of verified POI chains: for every node the mean of its verified POI chains length is computed, the mean of those means is the depicted result.



(e) Rate in the verified POI chains: the mean of the rates users assign to POIs.

Figure 5.3: Evolution of the length and number of unverified and verified chains (continued).



(a) Number of known nodes: mean of the number of known nodes by every node.

(b) Rate or level of trust of the known nodes: mean of the rates users assign to other users as POI reviewers.

Figure 5.4: Number of known nodes and their levels of trust progress.

about 20 other users which he trusts with a rate of approximately 1.5 tells us that an attacker trying to profile a user will have to guess which of the 20 trusted nodes he relies on really issued the review. This problem becomes increasingly harder as the days go by. For instance, after 20 days of simulation an attacker would have to find the real reviewer from a group of approximately 120 users.

All in all, the results show that reviewing POIs on behalf of other users with a moderate frequency has hardly an impact on the system performance while their privacy is protected. However, in a scenario where users review as many POIs on behalf of others as they do for themselves the results point to the fact that borrowing identities to preserve user privacy poses a constraint on how fast the reputation system develops.

Chapter 6

Visual Light Communication in Vehicular Ad-hoc Networks (VANETs)

Chains of Trust and *Anonymous Chains of Trust* rely on radio communication (Wireless Access in Vehicular Environments (WAVE)-Direct Short Range Communication (DSRC)) to transmit information. Radio communication, however, is inherently vulnerable to natural interferences and intentional jamming. Furthermore, in areas with a high number of vehicles their radio devices compete for access to the transmission medium, which means that some users may be able to transmit while others may not. In this section we explore a different approach.

The last goal of this thesis is to determine whether Visual Light Communication (VLC) could be an effective way to transmit information in a Vehicular Ad-hoc Network (VANET) (either on its own or in collaboration with WAVE-DSRC). However, the fact that the technology is not yet fully developed has to be taken into consideration. In addition, current research is focused on indoor applications because of its lower complexity. As a result, our experiments will only focus on the transmission range and we will consider 5m to be the maximum VLC range, because beyond that distance the data rate decreases dramatically. Notice how 5m should be enough to allow a vehicle to at least communicate with its immediate neighbors.

In the simulated scenario every vehicle is equipped with a set of VLC emitters and receivers distributed as depicted in Fig. 6.1. Even though the emitter's transmission cone is yet to be defined by manufacturers, we do know that LEDs are relatively inexpensive, which allows us to install several emitter-receiver sets in ar-

ray to maximize the chance of a successful transmission regardless of the vehicles' position.

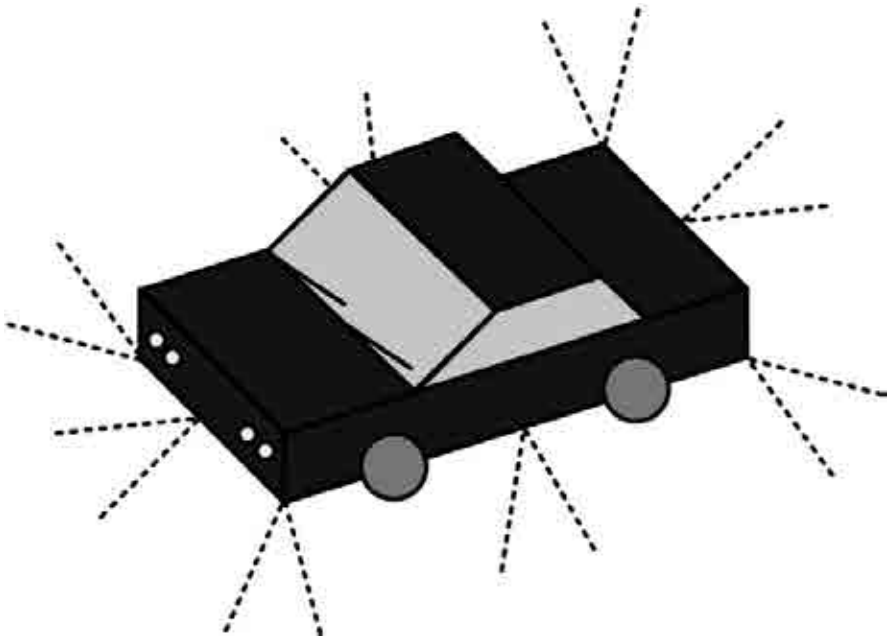


Figure 6.1: Emitter-receiver sets positioned in a vehicle and their transmission cone.

In order to determine how VLC would perform in a real VANET we need a realistic simulation tool. Simulation tools like Glomosim or ns-2 were discarded because in order to simulate hundreds of thousands of nodes they require a massive amount of memory. Thus, we were inclined to use a modified design of our own simulation tool [110, 111]. Like in [88], it was decided to analyze the realistic vehicular trace produced by the *Multi-Agent Traffic Simulator* (MMTS) developed by K.Nagel at ETH Zurich. The MMTS is capable of simulating public and private traffic over real regional road maps of Switzerland with a high level of realism. It models the behavior of people living in the area, reproducing their movement (using vehicles) within a period of 24 hours. The decision of each individual depends on the area it lives in. The individuals in the simulation are distributed over the cities and villages according to statistical data gathered by a census. Within the 24 hours of simulation, all individuals choose a time to travel and the mean of transportation according to their needs and environment, e.g., one individual might take a car and go to work in the early morning, another one wakes up later

and goes shopping using public transportation, etc. All in all, with over 260.000 simulated nodes or vehicles in an area of around 250 km x 260 km, this mobility trace suited our simulation needs.

The mobility trace roughly consists in a x, y, z position update for every node every t seconds (different periods t for every node). It has 3 different types of updates: node starts a trip, node updates its position and node finishes a trip. Every time the trace provides an update on a vehicle's position, the simulation tool computes a rectilinear trajectory between the previous x, y, z and the new x', y', z' coordinates for the updated node, as depicted in Fig. 6.2. Then, its trajectory is compared to the trajectory of every active node (every vehicle currently on the road) and it determines if their paths cross and should that be the case if the crossing point falls within the segment delimited by the x, y, z and x', y', z' coordinates. Finally, it also takes into consideration the speed of both vehicles and the transmission range of VLC to determine if the vehicles are in range of one another and if the transmission succeeded.

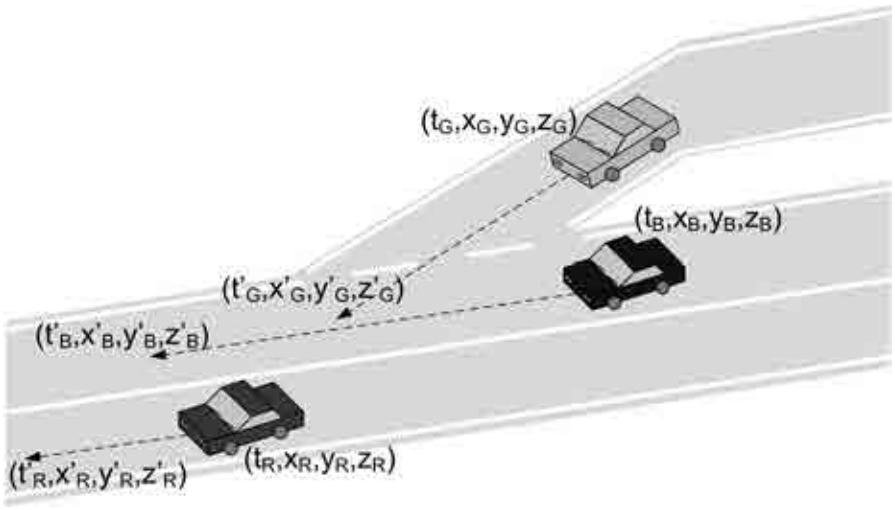


Figure 6.2: In range detection based on vehicles R, G, B trajectories.

In the next sections we present the results of our simulations. VLC can transmit at 115Kbps at approximately 5m [100], although in order to account for future improvements on the technology we will also consider ranges of 10m and 15m and compare those results to the results yielded by the range of WAVE-DSRC (120m). It should be noted that the vehicles or nodes being simulated spend an average of 3,134.17s on the road (slightly less than an hour) and make 1.99 trips. Our

simulations were designed with the following goals in mind:

- compute the mean of the number of packets received by each node and its distribution.
- study the transmission of information over an area with a gossip protocol.
- identify the limitations of WAVE-DSRC on the usage of the physical medium.

6.1 Average Number of Received Packets

As depicted in Fig.6.3, the average number of packets received by every node is computed. For ranges 5m, 10m and 15m it can be seen that a similar number of packets was received (443.38 packets, 458.55 packets, 473.88 packets). However, when compared with the 120m range of WAVE (1,491.60 packets) the difference in performance is quite evident. If we look at the distribution of the mean, it can be observed that the VLC ranges share similar results: over 150,000 nodes receive between 0 and 499 packets, while over 300 receive 2,500 or more. With a range of 120m, over 70,000 nodes in the WAVE VANET receive between 0 and 499 packets, while over 50,000 nodes receive 2,500 or more.

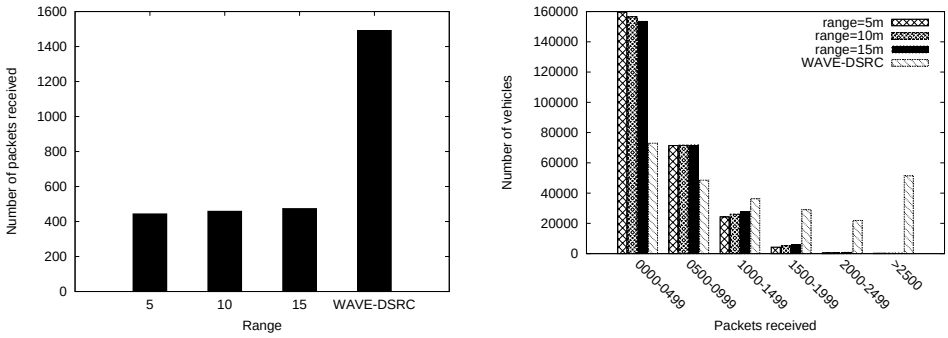


Figure 6.3: Mean and distribution of the number of packets received by each node.

Solely looking at these results it can be firmly stated that VLC cannot replace WAVE-DSRC without a decrease in the network's throughput. That being said, the results also show that even with a range of 5m 443 packets were received, which means that VLC may be able to work together with WAVE to protect VANETs from DoS attacks.

range: 5 m - packets transmitted: 25,507,586

0	23,660	11,644	818	6
0	3,362	171,904	520,186	219,578
64	3,934	142,216	16,823,206	3,580,711
20,877	87,488	118,624	2,502,195	1,245,696
0	2,968	27,065	1,384	0

range: 10 m - packets transmitted: 25,555,445

0	24,645	12,008	818	6
0	3,377	173,775	523,223	219,924
64	3,935	142,428	16,851,110	3,588,316
20,877	87,636	118,771	2,505,502	1,247,582
0	2,968	27,093	1,387	0

range: 15 m - packets transmitted: 25,581,042

0	24,646	12,013	818	6
0	3,377	174,011	523868	220,204
64	3,936	142,564	16,866,764	3,591,878
20,877	87,716	118,982	2,508,462	1,249,387
0	2,968	27,114	1,387	0

range: 120 m - packets transmitted: 25,859,059

0	25,724	12,435	818	6
0	3,411	177,226	534,066	224,307
69	4,055	146,004	17,017,671	3,638,438
22,139	93,307	121,331	2,534,698	1,271,323
0	2,968	27,627	1,436	0

Figure 6.4: Distribution of packets transmitted in the traveled area.

6.2 Received Packets over an Area

In order to find out how important the transmission range is to propagate a message over a certain area another experiment was designed. Considering the results from the previous section, a node which received an average number of packets was selected as a representative sample of the network population. In the new simulation, that node will broadcast a packet every time its position is updated, at the same time the rest of the network will remain silent until they receive that message. From that point onwards they too will broadcast the message to its neighbors and so on until the simulation finishes.

In Fig. 6.4 we can see the result of the described scenario in the number of packets that were transmitted. The three different ranges for VLC (5m, 10m and 15m) obtained very similar results both in number of packets and their distribution. As far as WAVE is concerned, even though it produced approximately 300,000 transmissions more than VLC it did so with a very similar distribution. These results show that shorter transmission ranges can be compensated by the use of gossip broadcast protocols.

6.3 Analysis of WAVE Scalability

In order to analyze the scalability of WAVE-DSRC a simulation in ns-3 [98] was implemented defining a vehicular scenario with 400 nodes arranged in 4 lanes as depicted in Fig.3.6, connected through a WAVE-DSRC 27Mbps link with a

Percentage of received broadcasts						
Number of packets / Period (s)	10	20	30	40	50	60
100	71.82	87.08	91.48	93.66	95.03	95.93
200	36.23	71.79	82.08	87.04	89.71	91.46
300	15.77	54.50	71.75	79.71	84.06	87.05
400	9.45	36.71	60.52	71.88	78.14	82.21
500	6.83	23.43	48.62	63.59	72.13	77.26
600	5.28	15.89	36.77	54.55	65.24	71.99
700	4.28	11.88	27.18	45.53	58.29	66.44
800	3.64	9.51	20.41	36.85	51.05	60.65

Table 6.1: Percentage of received broadcasts for every simulated scenario.

120 meters range (like it was done in section 3.8). This scenario represents a traffic jam, which is the worse possible situation for radio communication due to the high density of vehicles. It should be noted that our simulation uses ns-3 *YansWifiPhyHelper* and *YansWifiChannelHelper* classes, as defined in [107].

In a nutshell, the simulation schedules the broadcast of $numPackets$ 1000 bytes packets at a randomly chosen time between the start of the simulation and its ending point, defined as *period*. For every scenario ($numPackets/period$ combination) the number of broadcasts received by each of the 400 simulated nodes is computed ($results_{numPackets,period}$) and compared with how many broadcasts each of those nodes would have received without packets loss ($reference_{numPackets}$), considering the mean as the scenario's result:

$$Received\ broadcasts\ \% = \sum_{node=1}^{400} \left(\frac{results^{node}_{numPackets,period}}{reference^{node}_{numPackets}} \right) \tag{6.1}$$

Looking at the results in table 6.1 it can be seen that for 400 packets every 30s the percentage of received broadcasts drops to 60.52%; the general tendency is that for a high number of packets transmitted over short periods the network throughput decreases. It should be noted that in this simulation we considered a scenario where every node broadcasts a message and there are no acknowledgements or retries. Had we considered bidirectional communication between vehicles and a road side unit the network throughput would have been even lower due to the number of retries. We strongly believe that VLC could help improve the delivery rate because in VLC users do not have to compete for the physical medium.

6.4 Conclusions

In this section we have explored the future possibilities of VLC replacing or complementing the current standard for communication in VANETs (WAVE-DSRC). Several experiments were prepared, each with a different objective in mind: (i) determine how many packets are received by each node (on average), (ii) how the transmitted information is distributed when VLC and WAVE are compared and (iii) analyze the success data rate of a worst case scenario (traffic jam) with WAVE. To the best of our knowledge, we have been the first to realistically consider the use of VLC in the VANET environment and provide realistic results that back our theory.

The first experiment shows that every node receives at least three times as many packets with WAVE as they receive with VLC in any of its different transmission ranges. For the second simulation we choose a node which receives an average number of packets and make him transmit in an epidemic way (at the beginning of the simulation he is the only one transmitting, but once a node receives that packet he starts transmitting as well). The results show that even though WAVE-DSRC obtained a higher number of transmitted packets, i.e., infected more nodes, the distribution in the x, y, z space was very similar. Which leads us to the conclusion that the short range of VLC can be made up for with the use of epidemic or gossip protocols. Finally, the third simulation shows at which point WAVE-DSRC stops getting information through due to the high competition for the medium and the resulting packet collisions. At that point, the network throughput could be improved by using VLC to transmit as well, since in VLC nodes do not need to compete for the physical medium due to the nature of light communication.

In addition, we also need to consider the fact that while WAVE, like all radio communication, is subject to jamming VLC is not. With WAVE, an attacker with a powerful enough radio device could easily cause a blind spot in the network (which would lead to a DoS) with dimensions depending on how good is his equipment. However, in order to jam the transmission of information in VLC the attacker would have to physically block the beam of light from the emitter to the receiver.

All in all, we believe that once VLC is ready to be deployed in the open air it will be an important addition to VANET communication. Working together with WAVE-DSRC, it will provide an extra link which can be used by public safety applications and whenever the WAVE-DSRC performance is below a certain threshold either due to the medium congestion or to an attack.

Chapter 7

Final Conclusions

In this thesis we have presented the current state of the art in security and reputation systems for Vehicular Ad-hoc Networks (VANETs) while examining each proposal and discussing its benefits and drawbacks.

Following the required background, we have introduced *Chains of Trust*, a new Point of Interest (POI) information dissemination scheme that builds a reputation system, which unlike most current solutions is solely based on the vehicular ad-hoc network, i.e., it requires no roadside infrastructure. Users manage their own identities and the information they input into the system is kept distributed among the vehicles in the network, i.e., there is no central entity where all the information is stored, thus protecting user privacy. In addition, it uses information aggregation techniques to accumulate POI reviews and increasing the speed at which the reputation system is built.

In order to determine how *Chains of Trust* would behave in a realistic scenario, we designed *poiSim*, our own simulation tool capable of handling a scenario with over 260,000 vehicles. Our objective was to show that by separating the communications from the application layer it is possible to build an application simulator which can execute simulations in the order of hundreds of thousands of nodes. This approach will produce more realistic results than using network simulators to simulate both the communication and application layers, as done in the vast majority of research articles.

We have also presented a novel mechanism to preserve users privacy in a reputation system. By allowing users to borrow each other's identities an attacker can never be sure of who was the real reviewer behind a given POI recommendation. In other words, users that trust each other form a virtual group where any user can use anybody else's identity, thus hiding behind the group. Moreover, this

technique should be transparent to the user reputation, since identity borrowing can only occur between users that trust each other, which by definition implies that their reviews for a given POI category are very similar and therefore interchangeable. To the best of our knowledge, this is the first time this technique has been applied to reputation systems.

Finally, we looked into the future of vehicular communication. Visual Light Communication (VLC) solves one of the biggest problems of radio communication by providing a secure communication channel resilient against jamming. VLC would provide a one-hop transmission system which could be specially helpful in case of an emergency, e.g., car accident, or whenever the radio channel was too populated and transmission became virtually impossible.

Chapter 8

Future Work

This section outlines several methods to expand this thesis. In the future, misbehavior simulation should be added to the simulation tool *poiSim*, so that the rewards and penalties system can effectively be evaluated. In addition, we would like to develop a modified version of *Anonymous Chains of Trust* where users have a direct link to an Internet Service Provider through cellular technologies, thus incorporating a Certification Authority (CA) in our scheme and compare how it performs compared to the ad-hoc version. Moreover, we would also like to explore the possibility of adapting our application to pedestrian networks.

Chapter 9

Acknowledgements

This work was partially supported by the EuroNF NoE and by Spanish grants TIN2010-21378-C02-01 and 2009-SGR-1167.

Bibliography

[1] D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, W. Schulz, Cartalk 2000: safe and comfortable driving based upon inter-vehicle-communication, in: Intelligent Vehicle Symposium, 2002. IEEE, Vol. 2, 2002, pp. 545–550 vol.2.

[2] M. Raya, J.-P. Hubaux, The security of vehicular ad hoc networks, in: SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks, ACM, New York, NY, USA, 2005, pp. 11–21. doi:http://doi.acm.org/10.1145/1102219.1102223.

[3] C.-H. Yeh, Y.-M. Huang, T.-I. Wang, H.-H. Chen, Descv–a secure wireless communication scheme for vehicle ad hoc networking, Mob. Netw. Appl. 14 (5) (2009) 611–624.

[4] P. D. Dawoud, D. S. Dawoud, R. Peplow, A proposal for secure vehicular communications, in: ICIS '09: Proceedings of the 2nd International Conference on Interaction Sciences, ACM, New York, NY, USA, 2009, pp. 1026–1032.

[5] K. P. Laberteaux, J. J. Haas, Y.-C. Hu, Security certificate revocation list distribution for vanet, in: VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking, ACM, New York, NY, USA, 2008, pp. 88–89.

[6] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, A. Lioy, Efficient and robust pseudonymous authentication in vanet, in: VANET '07: Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks, ACM, New York, NY, USA, 2007, pp. 19–28.

[7] F. Picconi, N. Ravi, M. Gruteser, L. Iftode, Probabilistic validation of aggregated data in vehicular ad-hoc networks, in: VANET '06: Proceedings of

the 3rd international workshop on Vehicular ad hoc networks, ACM, New York, NY, USA, 2006, pp. 76–85.

[8] M. Raya, A. Aziz, J.-P. Hubaux, Efficient secure aggregation in vanets, in: VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks, ACM, New York, NY, USA, 2006, pp. 67–75.

[9] T. Moore, M. Raya, J. Clulow, P. Papadimitratos, R. Anderson, J.-P. Hubaux, Fast exclusion of errant devices from vehicular networks, in: Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on, 2008, pp. 135–143.

[10] P. Kamat, A. Baliga, W. Trappe, An identity-based security framework for vanets, in: VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks, ACM, New York, NY, USA, 2006, pp. 94–95.

[11] A. Wasef, X. S. Shen, Rep: Location privacy for vanets using random encryption periods, Mob. Netw. Appl. 15 (1) (2010) 172–185.

[12] A. Jøsang, J. Golbeck, Challenges for robust trust and reputation systems., in: Proceedings of the 5th International Workshop on Security and Trust Management (STM 2009), 2009.

[13] K. Hoffman, D. Zage, C. Nita-Rotaru, A survey of attack and defense techniques for reputation systems, ACM Comput. Surv. 42 (2009) 1:1–1:31. doi:http://doi.acm.org/10.1145/1592451.1592452.
URL http://doi.acm.org/10.1145/1592451.1592452

[14] R. A. Uzcategui, G. Acosta-Marum, Wave: a tutorial, Communications Magazine, IEEE 47 (5) (2009) 126–133.

[15] H. L. Minh, D. O'Brien, G. Faulkner, L. Zeng, K. Lee, D. Jung, Y. Oh, E. T. Won, 100-mb/s nrz visible light communications using a postequalized white led, Photonics Technology Letters, IEEE 21 (15) (2009) 1063 –1065. doi:10.1109/LPT.2009.2022413.

[16] H. L. Minh, D. O'Brien, G. Faulkner, L. Zeng, K. Lee, D. Jung, Y. Oh, 80 mbit/s visible light communications using pre-equalized white led, in: Optical Communication, 2008. ECOC 2008. 34th European Conference on, 2008, pp. 1 –2. doi:10.1109/ECOC.2008.4729532.

[17] IEEE 802.15 Task Group 7 (TG7) Visible Light Communication (Nov. 2012).
URL `http://www.ieee802.org/15/pub/TG7.html`

[18] Visible Light Communications Consortium (Nov. 2012).
URL `http://www.vlcc.net`

[19] Realistic mobility vehicular trace by K. Nagel at ETH Zurich (Nov. 2012).
URL `http://www.lst.inf.ethz.ch/research/ad-hoc/car-traces/`

[20] Ns-2.
URL `http://isi.edu/nsnam/ns/`

[21] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, H. Hartenstein, Overhaul of ieee 802.11 modeling and simulation in ns-2, in: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, MSWiM '07, ACM, New York, NY, USA, 2007, pp. 159–168. doi:http://doi.acm.org/10.1145/1298126.1298155.

[22] J. Chen, C.-C. Wang, F. C.-D. Tsai, C.-W. Chang, S.-S. Liu, J. Guo, W.-J. Lien, J.-H. Sum, C.-H. Hung, The design and implementation of wimax module for ns-2 simulator, in: Proceeding from the 2006 workshop on ns-2: the IP network simulator, WNS2 '06, ACM, New York, NY, USA, 2006. doi:http://doi.acm.org/10.1145/1190455.1190458.

[23] D. Mahrenholz, S. Ivanov, Real-time network emulation with ns-2, in: Distributed Simulation and Real-Time Applications, 2004. DS-RT 2004. Eighth IEEE International Symposium on, 2004, pp. 29–36. doi:10.1109/DS-RT.2004.34.

[24] GloMoSim.
URL `http://pcl.cs.ucla.edu/projects/glomosim/`

[25] X. Zeng, R. Bagrodia, M. Gerla, Glomosim: a library for parallel simulation of large-scale wireless networks, SIGSIM Simul. Dig. 28 (1998) 154–161. doi:http://doi.acm.org/10.1145/278009.278027.

[26] E. Royer, P. Melliar-Smith, L. Moser, An analysis of the optimum node density for ad hoc mobile networks, in: Communications, 2001. ICC 2001. IEEE International Conference on, Vol. 3, 2001, pp. 857–861 vol.3. doi:10.1109/ICC.2001.937360.

[27] K. Xu, X. Hong, M. Gerla, An ad hoc network with mobile backbones, in: Communications, 2002. ICC 2002. IEEE International Conference on, Vol. 5, 2002, pp. 3138–3143 vol.5. doi:10.1109/ICC.2002.997415.

[28] The OPNET modeler.
URL `http://www.opnet.com`

[29] X. Chang, Network simulations with opnet, in: Simulation Conference Proceedings, 1999 Winter, Vol. 1, 1999, pp. 307–314 vol.1. doi:10.1109/WSC.1999.823089.

[30] Q. Ding, X. Li, M. Jiang, X. Zhou, Reputation management in vehicular ad hoc networks, in: Multimedia Technology (ICMT), 2010 International Conference on, 2010, pp. 1–5. doi:10.1109/ICMULT.2010.5632149.

[31] A. Patwardhan, A. Joshi, T. Finin, Y. Yesha, A data intensive reputation management scheme for vehicular ad hoc networks, in: Mobile and Ubiquitous Systems: Networking Services, 2006 Third Annual International Conference on, 2006, pp. 1–8. doi:10.1109/MOBIQ.2006.340422.

[32] S. Dhurandher, M. Obaidat, A. Jaiswal, A. Tiwari, A. Tyagi, Securing vehicular networks: A reputation and plausibility checks-based approach, in: GLOBECOM Workshops (GC Wkshps), 2010 IEEE, 2010, pp. 1550–1554. doi:10.1109/GLOCOMW.2010.5700199.

[33] N.-W. Lo, H.-C. Tsai, A reputation system for traffic safety event on vehicular ad hoc networks, EURASIP J. Wirel. Commun. Netw. 2009 (2009) 9:1–9:2. doi:http://dx.doi.org/10.1155/2009/125348.

[34] IEEE, Ieee trial-use standard for wireless access in vehicular environments (wave)–networking services, IEEE Std 1609.3-2007 (2007) c1–87.

[35] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, J. pierre Hubaux, Certificate revocation in vehicular networks.

[36] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, K. Sezaki, Caravan: Providing location privacy for vanet, in: in Embedded Security in Cars (ESCAR, 2005.

[37] L. Huang, K. Matsuura, H. Yamane, K. Sezaki, Enhancing wireless location privacy using silent period, in: Wireless Communications and Networking Conference, 2005 IEEE, Vol. 2, 2005, pp. 1187–1192 Vol. 2.

[38] U. S. D. of Transportation National highway traffic safety administration, Vehicle safety communications project - final rep., Tech. rep. (April 2006). URL `http://www-nrd.nhtsa.dot.gov/`

[39] X. Lin, X. Sun, P.-H. Ho, X. Shen, Gsis: A secure and privacy-preserving protocol for vehicular communications, Vehicular Technology, IEEE Transactions on 56 (6) (2007) 3442–3456.

[40] M. Gerlach, A. Festag, T. Leinmüller, G. Goldacker, C. Harsch, Security architecture for vehicular communication, in: Fourth International Workshop on Intelligent Transportation (WIT 2007), 2007.

[41] P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, M. Raya, Architecture for secure and private vehicular communications, in: Telecommunications, 2007. ITST '07. 7th International Conference on ITS, 2007, pp. 1–6.

[42] C. Laurendeau, M. Barbeau, Secure anonymous broadcasting in vehicular networks, in: Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on, 2007, pp. 661–668.

[43] S. Rass, S. Fuchs, M. Schaffer, K. Kyamakya, How to protect privacy in floating car data systems, in: VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking, ACM, New York, NY, USA, 2008, pp. 17–22.

[44] B. K. Chaurasia, S. Verma, Maximizing anonymity of a vehicle through pseudonym updation, in: WICON '08: Proceedings of the 4th Annual International Conference on Wireless Internet, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2008, pp. 1–6.

[45] P. P. Papadimitratos, G. Mezzour, J.-P. Hubaux, Certificate revocation list distribution in vehicular communication systems, in: VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking, ACM, New York, NY, USA, 2008, pp. 86–87.

[46] A. Studer, E. Shi, F. Bai, A. Perrig, Tacking together efficient authentication, revocation, and privacy in vanets, in: Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on, 2009, pp. 1–9. doi:10.1109/SAHCN.2009.5168976.

[47] Y. Sun, R. Lu, X. Lin, X. Shen, J. Su, A secure and efficient revocation scheme for anonymous vehicular communications, in: Communications (ICC), 2010 IEEE International Conference on, 2010, pp. 1–6. doi:10.1109/ICC.2010.5502130.

[48] M. Nowatkowski, H. Owen, Certificate revocation list distribution in vanets using most pieces broadcast, in: IEEE SoutheastCon 2010 (SoutheastCon), Proceedings of the, 2010, pp. 238–241. doi:10.1109/SECON.2010.5453881.

[49] B. Wiedersheim, Z. Ma, F. Kargl, P. Papadimitratos, Privacy in inter-vehicular networks: Why simple pseudonym change is not enough, in: Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on, 2010, pp. 176–183. doi:10.1109/WONS.2010.5437115.

[50] E. van Heyst D. Chaum, Group signatures, Advances in Cryptology 1981 - 1997 - EUROCRYPT '91, Springer-Heidelberg 1440 (1999) 127–133.

[51] A. Shamir, Identity-based cryptosystems and signature schemes, in: Proceedings of CRYPTO 84 on Advances in cryptology, Springer-Verlag New York, Inc., New York, NY, USA, 1985, pp. 47–53.

[52] D. Boneh, M. K. Franklin, Identity-based encryption from the weil pairing, in: CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, Springer-Verlag, London, UK, 2001, pp. 213–229.

[53] B. L. N. M. M. Barreto, J. Quisquater, Efficient and provably-secure identity-based signatures and signcryption from bilinear maps, in: Proc. Advances in Cryptology - Asiacrypt'05, Vol. 3788, Springer-Verlag, 2005, pp. 515–532.

[54] D. Boneh, X. Boyen, H. Shacham, Short group signatures, in: M. Franklin (Ed.), Proceedings of Crypto 2004, Vol. 3152 of LNCS, Springer-Verlag, 2004, pp. 41–55.

[55] D. S. G. Atenies, G. Tsudik, Quasi-efficient revocation of group signatures, in: Proc. Financ. Cryptogr., 2002, pp. 183–197.

[56] D. Boneh, H. Shacham, Group signatures with verifier-local revocation, in: Proc. ACM CCS, ACM Press, 2004, pp. 166–177.

[57] Y. T. A. Kiayias, M. Yung, Traceable signature, in: Proc. Adv. Cryptology - Eurocrypt, Vol. 3027 of LNCS, Springer-Verlag, 2004, pp. 571–589.

[58] M. Raya, P. Papadimitratos, J.-P. Hubaux, Securing vehicular communications, Wireless Communications, IEEE 13 (5) (2006) 8–15.

[59] M. Bellare, D. Micciancio, B. Warinschi, Foundations of group signatures: formal definition, simplified requirements and a construction based on trapdoor permutations, in: E. Biham (Ed.), Advances in cryptology - EUROCRYPT 2003, proceedings of the internarional conference on the theory and application of cryptographic techniques, Vol. 2656 of Lecture Notes in Computer Science, Springer-Verlag, Warsaw, Poland, 2003, pp. 614–629.

[60] M. Bellare, H. Shi, C. Zhang, Foundations of group signatures: The case of dynamic groups, in: CT-RSA'05, Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 136–153.

[61] F. Dotzer, Privacy issues in vehicular ad hoc networks, in: Privacy Enhancing Technologies, Vol. 3856 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006, pp. 197–209.

[62] D. Reid, An algorithm for tracking multiple targets, Automatic Control, IEEE Transactions on 24 (6) (1979) 843–854. doi:10.1109/TAC.1979.1102177.

[63] G. A.-M. Roberto A. Uzcátegui, Wave: A tutorial, IEEE Communications Magazine.

[64] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, J.-P. Hubaux, Eviction of misbehaving and faulty nodes in vehicular networks, Selected Areas in Communications, IEEE Journal on 25 (8) (2007) 1557–1568.

[65] B. H. Bloom, Space/time trade-offs in hash coding with allowable errors, Commun. ACM 13 (7) (1970) 422–426.

[66] P. Golle, D. Greene, J. Staddon, Detecting and correcting malicious data in vanets, in: VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, ACM, New York, NY, USA, 2004, pp. 29–37.

[67] B. Xiao, B. Yu, C. Gao, Detection and localization of sybil nodes in vanets, in: DIWANS '06: Proceedings of the 2006 workshop on Dependability issues

in wireless ad hoc networks and sensor networks, ACM, New York, NY, USA, 2006, pp. 1–8.

[68] M. Ghosh, A. Varghese, A. Gupta, A. A. Kherani, S. N. Muthaiah, Detecting misbehaviors in vanet with integrated root-cause analysis, Ad Hoc Networks 8 (7) (2010) 778–790. doi:DOI: 10.1016/j.adhoc.2010.02.008.

[69] R. Anderson, M. Kuhn, Tamper resistance - a cautionary note, Proceedings of the Second Usenix Workshop on Electronic Commerce (November 1996).

[70] R. Anderson, M. Kuhn, Low cost attacks on tamper resistant devices, in: IWSP: International Workshop on Security Protocols, LNCS, 1997.

[71] E. Biham, A. Shamir, Differential fault analysis of secret key cryptosystems, in: CRYPTO, 1997, pp. 513–525.

[72] J. Douceur, J. S. Donath, The sybil attack, 2002, pp. 251–260.

[73] T. Moore, J. Clulow, S. Nagaraja, R. Anderson, New strategies for revocation in ad-hoc networks.

[74] T. Nadeem, S. Dashtinezhad, C. Liao, L. Iftode, Trafficview: Traffic data dissemination using car-to-car communication, ACM SIGMOBILE Mobile Computing and Communications Review 8 (2004) 2004.

[75] B. Scheuermann, C. Lochert, J. Rybicki, M. Mauve, A fundamental scalability criterion for data aggregation in vanets, in: MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking, ACM, New York, NY, USA, 2009, pp. 285–296.

[76] S. Dietzel, E. Schoch, B. Konigs, M. Weber, F. Karl, Resilient secure aggregation for vehicular networks, IEEE Network 24 (1) (2010) 26–31.

[77] B. Yu, J. Gong, C.-Z. Xu, Catch-up: a data aggregation scheme for vanets, in: VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking, ACM, New York, NY, USA, 2008, pp. 49–57.

[78] K. Ibrahim, M. C. Weigle, Cascade: Cluster-based accurate syntactic compression of aggregated data in vanets, in: In Proceedings of the IEEE International Workshop on Automotive Networking and Applications (AutoNet), Dec., 2008.

[79] S. Dietzel, B. Bako, E. Schoch, F. Kargl, A fuzzy logic based approach for structure-free aggregation in vehicular ad-hoc networks, in: Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking, VANET '09, ACM, New York, NY, USA, 2009, pp. 79–88. doi:http://doi.acm.org/10.1145/1614269.1614283.

[80] C. Lochert, B. Scheuermann, M. Mauve, Probabilistic aggregation for data dissemination in vanets, in: VANET '07: Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks, ACM, New York, NY, USA, 2007, pp. 1–8.

[81] P. Flajolet, G. N. Martin, Probabilistic counting algorithms for data base applications, Journal of Computer and System Sciences 31 (2) (1985) 182–209. doi:DOI: 10.1016/0022-0000(85)90041-8.

[82] S.-B. Lee, G. Pan, J.-S. Park, M. Gerla, S. Lu, Secure incentives for commercial ad dissemination in vehicular networks, in: MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing, ACM, New York, NY, USA, 2007, pp. 150–159.

[83] Y. Zhang, J. Zhao, G. Cao, Roadcast: a popularity aware content sharing scheme in vanets, SIGMOBILE Mob. Comput. Commun. Rev. 13 (4) (2009) 1–14.

[84] A. Tajeddine, A. Kayssi, A. Chehab, A privacy-preserving trust model for vanets, in: Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, 2010, pp. 832–837. doi:10.1109/CIT.2010.157.

[85] G. Montenegro, C. Castelluccia, Statistically unique and cryptographically verifiable (sucv) identifiers and addresses, in: In Proceedings of the 9th Annual Network and Distributed System Security Symposium (NDSS, 2002.

[86] S. Park, B. Aslam, C. Zou, Long-term reputation system for vehicular networking based on vehicle's daily commute routine, in: Consumer Communications and Networking Conference (CCNC), 2011 IEEE, 2011, pp. 436–441. doi:10.1109/CCNC.2011.5766507.

[87] K. Fall, A delay-tolerant network architecture for challenged internets, in: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '03, ACM, New York, NY, USA, 2003, pp. 27–34. doi:10.1145/863955.863960.

[88] V. Naumov, R. Baumann, T. Gross, An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces, in: MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '06, ACM, New York, NY, USA, 2006, pp. 108–119. doi:http://doi.acm.org/10.1145/1132905.1132918.

[89] Simulation of urban mobility.
URL http://sumo.sourceforge.net

[90] M. Piórkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, J.-P. Hubaux, Trans: realistic joint traffic and network simulator for vanets, SIGMOBILE Mob. Comput. Commun. Rev. 12 (1) (2008) 31–33. doi:http://doi.acm.org/10.1145/1374512.1374522.

[91] U. S. Census Bureau, Topologically integrated geographic encoding and referencing (TIGER) system.
URL http://www.census.gov/geo/www/tiger

[92] J. Härri, M. Fiore, F. Filali, C. Bonnet, Vehicular mobility simulation with vanetmobisim, SIMULATION 87 (4) (2011) 275–300. doi:10.1177/0037549709345997.

[93] J. Harri, F. Filali, C. Bonnet, Mobility models for vehicular ad hoc networks: a survey and taxonomy, Communications Surveys Tutorials, IEEE 11 (4) (2009) 19–41. doi:10.1109/SURV.2009.090403.

[94] QualNet developer.
URL http://www.scalable-networks.com

[95] S. Roy, D. Saha, S. Bandyopadhyay, T. Ueda, S. Tanaka, A network-aware mac and routing protocol for effective load balancing in ad hoc wireless networks with directional antenna, in: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '03, ACM, New York, NY, USA, 2003, pp. 88–97. doi:http://doi.acm.org/10.1145/778415.778427.

[96] OmNet.
URL http://www.omnetpp.org/

[97] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, S. Valentin, Simulating wireless and mobile networks in omnet++ the mixim vision, in: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, Simutools '08, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2008, pp. 71:1–71:8.

[98] T. R. Henderson, S. Roy, S. Floyd, G. F. Riley, ns-3 project goals, in: Proceeding from the 2006 workshop on ns-2: the IP network simulator, WNS2 '06, ACM, New York, NY, USA, 2006. doi:http://doi.acm.org/10.1145/1190455.1190468.

[99] A. G. Bell, On the production and reproduction of sound by light, American Journal of Science, Third Series XX (11) (1880) 305–324.

[100] IEEE 802.15 Visible Light Communication Overview (Nov. 2012). URL `http://ieee802.org/802_tutorials/2008-03/15-08-0114-02-0000-VLC_Tutorial_MCO_Samsung-VLCC-Oxford_2008-03-17.pdf`

[101] C. Davis, I. Smolyaninov, S. Milner, Flexible optical wireless links and networks, Communications Magazine, IEEE 41 (3) (2003) 51 – 57. doi:10.1109/MCOM.2003.1186545.

[102] P. Chowdhury, M. Tornatore, S. Sarkar, B. Mukherjee, Building a green wireless-optical broadband access network (woban), Lightwave Technology, Journal of 28 (16) (2010) 2219 –2229. doi:10.1109/JLT.2010.2044369.

[103] S. Sarkar, S. Dixit, B. Mukherjee, Hybrid wireless-optical broadband-access network (woban): A review of relevant challenges, Lightwave Technology, Journal of 25 (11) (2007) 3329 –3340. doi:10.1109/JLT.2007.906804.

[104] Q. Liu, C. Qiao, G. Mitchell, S. Stanton, Optical wireless communication networks for first- and last-mile broadband access], J. Opt. Netw. 4 (12) (2005) 807–828. doi:10.1364/JON.4.000807.

[105] D. O'Brien, L. Zeng, H. Le-Minh, G. Faulkner, J. Walewski, S. Randel, Visible light communications: Challenges and possibilities, in: Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on, 2008, pp. 1 –5. doi:10.1109/PIMRC.2008.4699964.

[106] S. Iwasaki, M. Wada, T. Endo, T. Fujii, M. Tanimoto, Basic experiments on paralle wireless optical communication for its, in: Intelligent Vehicles Symposium, 2007 IEEE, 2007, pp. 321 –326. doi:10.1109/IVS.2007.4290134.

[107] M. Lacage, T. R. Henderson, Yet another network simulator, in: Proceeding from the 2006 workshop on ns-2: the IP network simulator, WNS2 '06, ACM, New York, NY, USA, 2006. doi:http://doi.acm.org/10.1145/1190455.1190467.

[108] Energy, transport and environment indicators - eurostat pocketbooks. eurostat 2010.
URL http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-DK-10-001/EN/KS-DK-10-001-EN.PDF

[109] Swiss confederation - population size and population composition.
URL http://www.bfs.admin.ch/bfs/portal/fr/index/themen/01/02/blank/key/bevoelkerungsstand.html

[110] D. A. Rivas, M. Guerrero-Zapata, Chains of trust in vehicular networks: A secure points of interest dissemination strategy, Ad Hoc Networks 10 (6) (2012) 1115 – 1133. doi:10.1016/j.adhoc.2012.02.011.

[111] D. A. Rivas, M. Guerrero-Zapata, Simulation of points of interest distribution in vehicular networks, SIMULATION 88 (11) (2012) 1390–1404. doi:10.1177/0037549712456440.

Appendix A

Publications

The following is a list of current and future publications, the journal to where they were submitted to and the date of submission/publication.

- David Antolino Rivas, Manel Guerrero Zapata, José M. Barceló Ordinas, Julian Morillo-Pozo: "Security on Vehicular Ad-hoc Networks (VANETs): Privacy, Misbehaving Nodes, False Information and Secure Data Aggregation". **Published** on **Journal of Network and Computer Applications**. JCR Impact Factor 1.065 [COMPUTER SCIENCE, HARDWARE & ARCHITECTURE 50/21 Q2]. July 2011.

- David Antolino Rivas, Manel Guerrero Zapata: "Chains of Trust: a Secure Points of Interest Dissemination Strategy". **Published** on **Ad Hoc Networks**. JCR Impact Factor 2.110 [TELECOMMUNICATIONS 79/13 Q1]. August 2012.

- David Antolino Rivas, Manel Guerrero Zapata: "Simulation of Points of Interest Distribution in Vehicular Networks". **Published** on **Simulation: Transactions of the Society for Modeling and Simulation International**. JCR Impact Factor 0.793 [SOFTWARE ENGINEERING 103/59 Q3]. October 2012.

- David Antolino Rivas, Manel Guerrero Zapata: "Anonymous Chains of Trust in Vehicular Networks: Preserving Users Privacy in a Reputation System". **Submitted** to **Ad Hoc Networks**. JCR Impact Factor 2.110 [TELECOMMUNICATIONS 79/13 Q1]. January 2013.

- David Antolino Rivas, Manel Guerrero Zapata: "Visual Light Communication in VANETs". **Submitted** to **Ad Hoc Networks**. JCR Impact Factor 2.110 [TELECOMMUNICATIONS 79/13 Q1]. November 2012.

Journal of Network and Computer Applications 34 (2011) 1942–1955

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Review

Security on VANETs: Privacy, misbehaving nodes, false information and secure data aggregation

David Antolino Rivas *, José M. Barceló-Ordinas, Manel Guerrero Zapata, Julián D. Morillo-Pozo

Department of Computer Architecture, Polytechnic University of Catalonia, C. Jordi Girona 1-3, Barcelona 08034, Spain

ARTICLE INFO

Article history:
Received 27 March 2011
Received in revised form
12 June 2011
Accepted 11 July 2011
Available online 20 July 2011

Keywords:
Security
Vehicular Ad hoc Networks
VANETs
Privacy
Certificates
Pseudonyms
Anonymity
Data aggregation

ABSTRACT

This article is a position paper on the current security issues in *Vehicular Ad hoc Networks* (VANETs). VANETs face many interesting research challenges in multiple areas, from privacy and anonymity to the detection and eviction of misbehaving nodes and many others in between. Multiple solutions have been proposed to address those issues. This paper surveys the most relevant while discussing its benefits and drawbacks. The paper explores the newest trends in privacy, anonymity, misbehaving nodes, the dissemination of false information and secure data aggregation, giving a perspective on how we foresee the future of this research area.

First, the paper discusses the use of *Public Key Infrastructure* (PKI) (and certificates revocation), location privacy, anonymity and group signatures for VANETs. Then, it compares several proposals to identify and evict misbehaving and faulty nodes. Finally, the paper explores the differences between syntactic and semantic aggregation techniques, cluster and non-cluster based with fixed and dynamic based areas, while presenting secure as well as probabilistic aggregation schemes.

Contents

1. Introduction

With the massive deployment of wireless technologies on motorized vehicles, automotive industries have opened a wide variety of possibilities for drivers and their passengers. Theoretically, anything from finding out the road conditions ahead to watching a movie through streaming is possible. Different kinds of applications will need different requirements. As mentioned by Reichardt et al. (2002) and Raya and Hubaux (2005a) applications can be categorized as follows:

1. *Safety related*:
   (a) *Traffic information messages*: used to disseminate traffic conditions in a region and thus affect public safety only indirectly—they are not time-critical.

* Corresponding author.
E-mail address: antolino@ac.upc.edu (D. Antolino Rivas).

1084-8045/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2011.07.006

(b) *General safety-related messages*: used by public safety applications such as cooperative driving and collision avoidance—they should satisfy an upper bound delay.
(c) *Liability-related messages*: they are only exchanged in liability-related situations such as accidents—time is not an issue, but the messages should be able to reveal the senders' ID to the law authorities.

2. *Others*:
(a) *Toll applications*: electronic toll collection systems like *AutoPASS* in Norway allow drivers to continue driving without having to stop at tolls.
(b) *TV and other multimedia content*: used to provide users with entertainment and information (movies, newspapers, etc.).
(c) *Advertisements*: businesses along the road (such as gas-stations and restaurants) could advertise themselves to drivers before they reached the business location, giving them enough time to compare different offers.

As far as safety applications' requirements are concerned, the integrity and the non-repudiation of the messages have to be ensured, albeit maintaining at the same time the user's privacy, as will be discussed in Section 3. Other applications, e.g., multimedia content distribution, may also need to encrypt their traffic to avoid eavesdropping from non-registered users. The use of *Certification Authorities* (CAs) and public key cryptography to protect *Vehicle to Vehicle* (V2V) and *Vehicle to Infrastructure* (V2I) communication fulfills most security requirements.

Vehicles have to be equipped with *On Board Units* (OBUs) to be able to communicate among them and with *Road Side Units* (RSUs). RSUs compose the roadside infrastructure which connects the vehicular network to a central system (e.g., a CA) or to the Internet.

There are a few published papers that survey the area of security in vehicular networks (Parno and Perrig, 2005; Raya and Hubaux, 2005b; Plobl et al., 2006). Nevertheless, they are quite outdated since their most recent cited papers are from the year 2005 while most of this article's references are from 2006 onward. In addition, they do not analyze more recent trends like the use of group signatures and specific aggregation techniques.

The remainder of this paper is organized as follows. In Section 2 the communications architecture used in VANETs is introduced. Section 3 explains how certificates and *Certificate Revocation Lists* (CRLs) are used and what are its main advantages and drawbacks and how the use of pseudonyms, group signatures and anonymous certificates can improve privacy and anonymity. Following, Section 4 introduces how to identify and exclude misbehaving and faulty nodes. In Section 5 several schemes for secure data aggregation are presented. Finally, in Section 6 we present our conclusions.

2. Vehicular communications and architecture

Vehicles will be equipped with a set of processors and sensors (Papadimitratos et al., 2009) dedicated to collect and analyze data related to (i) mechanical and electronic components of the vehicle (e.g., battery charge, brakes, fuel) and (ii) vehicle traveling related information (e.g., GPS data, vehicle speed and direction, radar data). Furthermore, vehicles will obtain data from other vehicles in their neighborhood and from RSUs.

Vehicular communication technologies comprise cellular (GPRS/UMTS), *Dedicated Short Range Communications* (DSRC) and the IEEE 802.11 technology family. Cellular communications can be used as a basis for long-range communications at low data rates (i.e., less than 2 Mb/s), mainly for V2I communication. Alternatively, WIFI IEEE 802.11a,b,g may provide short-range access (i.e., less than 100 m) to RSUs at medium–high data rates (i.e., between 1 and 54 Mb/s). Finally, *Wireless Access in Vehicular Environments* (WAVE) standards allow short-range communications (i.e., less than 1000 m) at data rates between 3 and 27 Mb/s. IEEE 802.11p WAVE (Uzcategui and Acosta-Marum, 2009) is defined to allow both V2V and V2I communications. WAVE comprises IEEE 802.11p and IEEE 1609.x standards. WAVE units support multichannel operation: primary management frames and *Wave Short Messages* (WSM) use a fixed *Control Channel* (CCH) while other management frames and data frames (e.g., IP datagrams) use a *Service Channel* (SCH). SCH exchanges require the devices to be members of the *WAVE Basic Services* (WBS) that act as the corresponding service sets in IEEE 802.11. At higher layers, the WAVE stack allows the transport of TCP/UDP using IPv6 datagrams. In this way, legacy of TCP/IP connectivity is ensured. Besides, WAVE also defines a *WAVE Short Message Protocol* (WSMP) to accommodate high-priority, time-sensitive traffic. It should also be considered that the WAVE 1609.2 standard defines security services for the WAVE stack, which include confidentiality, authenticity, integrity and anonymity services.

3. Techniques to achieve privacy

In the near future, VANETs are going to change the way people drive and it will solely depend on the security measures that are implemented if they do it for the better or for the worse. The creation of VANETs can help improve traffic management and roadside safety. Unfortunately, a VANET also comes with its own set of challenges, particularly in security and privacy. As a special implementation of mobile ad hoc networks, a VANET is subject to many security threats, which can lead to attacks and service abuses. For instance, an attacker could tamper with traffic applications and make its users believe that there is a traffic jam in a particular road making them to take an alternative way, thus freeing the original road for the attacker's benefit. A more dangerous example would be for an attacker to sign liability messages with a fake identity so that he could not be linked to a car accident scene. Furthermore, network applications could also be used for more subtle and equally illegal objectives such as tracking people on their vehicles. Therefore, there is a real demand for security mechanisms, especially for those that protect the user's privacy.

The security architecture developed by the *Vehicle Safety Communications Consortium (VSCC)* and subsequently submitted to *IEEE P1609.2* (IEEE, 2007) defines a PKI-based approach for securing messages sent in V2V and V2I communication. The standard, however, does not address privacy issues. Raya et al. (2006b) propose different mechanisms for certificate revocation and discuss privacy issues in vehicular networks. Conditional privacy preservation must be achieved in the sense that user-related private information, e.g., driver's name, license plate, position has to be protected, while at the same time authorities have to be able to reveal the identity of message senders in case of a traffic event dispute, such as a car accident. Therefore, it is critical to develop a suite of elaborate and carefully designed security mechanisms to achieve security and conditional privacy preservation in VANETs before they can be deployed.

Among the proposals to achieve privacy, different techniques can be identified:

- anonymous certificates,
- group signatures,
- pseudonyms and pseudonyms certificates.

Table 1 summarizes the privacy schemes and classifies them according to whether a scheme uses (i) anonymous certificates, (ii) group signatures or (iii) pseudonyms to achieve privacy. Table 2 indicates if a work considers problems as group formation, traceability, revocation or message linkability. The *dynamic*

Table 1
Taxonomy of privacy and certificate revocation schemes.

	Anonymous certificates	Group signatures	Pseudonyms	Group formation
Raya and Hubaux (2005a)	X		X	
Sampigethaya et al. (2005)			X	
Huang et al. (2005)			X	
National Highway Traffic Safety Administration (2006)	X			
Raya et al. (2006a)		X	X	X
Lin et al. (2007)		X		
Gerlach et al. (2007)			X	
IEEE (2007)			X	
Papadimitratos et al. (2007)			X	
Calandriello et al. (2007)		X	X	
Laurendeau and Barbeau (2007)	X			
Rass et al. (2008)			X	
Chaurasia and Verma (2008)			X	
Papadimitratos et al. (2008)				
Studer et al. (2009)	X			
Sun et al. (2010)			X	
Nowatkowski and Owen (2010)				
Wiedersheim et al. (2010)	X		X	

Table 2
Taxonomy of privacy and certificate revocation schemes (continued).

	Revocation	Traceability	Dynamic	Linkability
Raya and Hubaux (2005a)	X	X	X	
Sampigethaya et al. (2005)			X	
Huang et al. (2005)			X	
National Highway Traffic Safety Administration (2006)	X	X	X	
Raya et al. (2006a)			X	
Lin et al. (2007)	X	X		
Gerlach et al. (2007)	X		X	X
IEEE (2007)				
Papadimitratos et al. (2007)				
Calandriello et al. (2007)	X	X	X	
Laurendeau and Barbeau (2007)	X			
Rass et al. (2008)			X	
Chaurasia and Verma (2008)			X	
Papadimitratos et al. (2008)	X			
Studer et al. (2009)	X			
Sun et al. (2010)	X			
Nowatkowski and Owen (2010)	X			
Wiedersheim et al. (2010)	X			X

column shows if the scheme dynamically changes the message signature keys.

Although the problem of certificate revocation is commented when needed throughout the whole section, we add at the end a specific subsection to point out other references in that field and discuss the most relevant mechanisms to reduce the size of CRLs.

3.1. Achieving privacy through anonymous certificates

One solution to the privacy problem is to use a list of anonymous certificates for message authentication, where the relationship of the list of anonymous certificates with a vehicle's driver is stored in a *Transportation Regulation Center* (TRC). For instance, Raya and Hubaux (2005a) introduce a security protocol based on anonymous certificates. With a pool of approximately 43 800 certificates, every time a vehicle wants to communicate with the network it randomly chooses one of the available certificates to sign a particular message and then discards it. In this way, the driver's privacy is guaranteed, since there is no way for an attacker to tell if two messages were sent by the same user.

To achieve conditional traceability, a unique electronic ID is assigned to each vehicle by which the police and authorities can verify the identity of the owner in case of any dispute. Although this scheme can effectively meet the conditional privacy requirement, it is far from efficient and can hardly become a scalable and reliable approach. Since the ID management authority stores all the anonymous certificates for each vehicle in its administrative region (province or country), once a malicious node is detected, the authority has to exhaustively search in a large database (probably 43 800 certificates $\times$ millions of cars) to find the ID related to the misbehaving anonymous public key. Besides, if a node needs to be revoked all its anonymous certificates have to be included in the CRL, which will then grow very fast.

In National Highway Traffic Safety Administration (2006) a similar solution is proposed. They also use short-lived certificates, although they are blindly signed by the CA. The *Escrow Authority* (EA) is responsible for maintaining the link between the anonymous certificates and the vehicle's real identity using a linkage marker, in order to deal with the "insider" attack. Still, they suffer from the same problems, because in order to revoke a vehicle all of its non-expired anonymous certificates have to be included in the CRL.

Laurendeau and Barbeau (2007) devise a scheme following a very different approach from the ones described above. In a nutshell, all the nodes share a *Network Authorization Key* (AK), which grants the privilege of broadcasting messages in the VANET. In addition, every vehicle has a secret key (SK) only known by the CA and itself. Whenever a node wants to broadcast a message it needs to ask the CA for the AK, which as we will see below needs to be a short-lived key. In order to enable the revocation of rogue vehicles their identifier is included in the message, although for privacy concerns it is encrypted with the CAs public key. Let us define the OBU_{id} of an anonymous node A as

$$\{Id_A, H_{SK_A}(Id_A \mid H_{AK}(M))\}_{CA} \quad (1)$$

The OBU_{id} is added to any message A wants to broadcast to prove its authorization to transmit a broadcast message M by hashing it with the network authorization key to produce a message digest $H_{AK}(M)$.

$$\{M, H_{AK}(M), CA, OBU_{id}\} \quad (2)$$

It should also be noted that the scheme relies on CRLs to revoke nodes from the network and the CA is the only one qualified to include them in the list. However, the AK is not updated until it has expired. Hence the need for a short-lived AK, since nothing keeps the rogue node from broadcasting bogus messages until the AK expires (vulnerability window). On the other hand, if we consider a scheme where information messages are transmitted from OBUs to RSUs, validated at the CA and then issued back from the RSUs as trusted messages to the vehicles (to which they would respond diminishing speed or stopping) the vulnerability window disappears, because the CA has permanent access to the CRL and can discard any message coming from a revoked node. However, safety message applications would suffer a great delay in comparison to schemes where the information is actually collected and delivered directly by the vehicle's neighbors. Therefore, this solution is not the best suited for this kind of applications.

3.2. Achieving privacy through group signatures

The main feature of the group signature scheme is that it provides anonymity to the group members, because any node inside the group can verify if a certain message was sent by a group member without knowing the sender's real identity inside the group.

Lin et al. (2007) integrate the techniques of Group Signature (Chaum and Van Heyst, 1999) and Identity-based Signature (Shamir, 1985) to solve the issues on security and conditional privacy preservation. They divide that problem into two parts: communication coming from an OBU and communication coming from an RSU. The main idea is to use group signatures to address the first part of the problem, so that messages are anonymously signed, while the identities of the senders can still be recovered by the authorities. In order to address the second part of the privacy problem they introduce a signature scheme that uses *Identity-based Cryptography* (IBC) (Boneh and Franklin, 2001) to digitally sign each message sent by an RSU to ensure its authenticity.

1. *Communication from an OBU*: The main issue is how to solve the contradiction between making the messages anonymous and at the same time traceable by the authorities. A secure group signature must be correct (honestly generated signatures can be verified), anonymous and unlinkable to the original identity although traceable under some circumstances (Lin et al., 2007). By using a group signature scheme such as the one described by Chaum and Van Heyst (1999) a verifier can judge whether the signer belongs to a group without actually knowing the signer's real identity in the group. Besides, if the situation ever requires it, the CA, which serves as a group manager, can reveal the signer's true identity. Lin et al. (2007) propose a role separation between the authority that provides the keys for the group and the law authorities that may need to trace a group member's real identity. Therefore, the role of the group manager is divided into a *Membership Manager* (MM), whose task is to assign private and group public keys to the vehicles, and a *Tracing Manager* (TM), i.e., the law authorities.
2. *Communication from an RSU*: Messages sent from RSUs do not need to remain anonymous. Therefore, the identifier string of each RSU can be used as the public key to sign its messages. The probably secure identity-based signature scheme described by Barreto et al. (2005) is the one chosen by Lin et al. (2007), since the length of the signature is greatly reduced thanks to the use of bilinear pairing.

Figure 1 depicts how the system works. Three types of network entities are identified: the TM, the MM and the mobile OBUs. The main idea is that all vehicles need to be registered with the MM and pre-loaded with the group public key and their own private key before they can join the network. When the vehicles are on the road, they regularly broadcast routine traffic related messages (position, speed, etc.). Should an accident occur (or any other kind of event that required the vehicles' real identities to be revealed) police officers would submit the messages collected at the time of the accident to the TM, who is responsible for the authorization of revealing the real identities of the wanted vehicles. The TM would then forward recovered clues and evidences to the MM which would search the real identity in its membership database.

In the article, the authors emphasize the need for a system that has the ability to selectively revoke the group membership of a compromised vehicle either by updating the group keys or by releasing a customized version of the *Revocation Lists* (RLs). If the group keys are updated, the private keys of the revoked vehicles are distributed in an RL so that unrevoked vehicles can locally update their private and group public keys, whereas the revoked vehicles

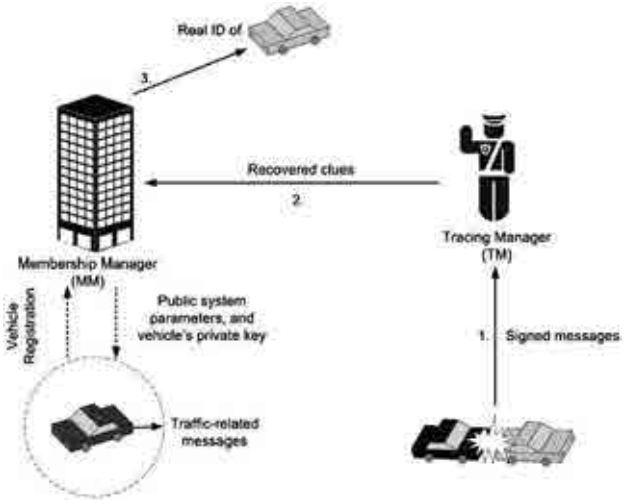


Fig. 1. Secure communication system.

cannot due to the signature scheme being used (Strong Diffie Hellman in groups with a bilinear map) (Boneh et al., 2004). However, this option introduces significant overhead due to the periodic changes of keys. Alternatively, a *Verifier-Local Revocation* (VLR) scheme (Atenies et al., 2002; Boneh and Shacham, 2004; Kiayias et al., 2004), similar to the traditional CRL, is very efficient (as long as the number of compromised vehicles is low) since only message verifiers are involved in the revocation check-up operation. In Lin et al. (2007), a hybrid scheme is proposed, which in general terms consists in using VLR until the number of revoked vehicles reaches a certain threshold T and then switching to key updating.

Some aspects remain unclear in Lin et al. (2007). For instance, the authors do not cover how the groups are formed, or if there is communication among them, so that if a node is revoked from a group it is revoked from all groups. Besides, if VRL relies on the fact that only the verifiers deal with revoked nodes that means that most of the group nodes are just dummy nodes (they do not interpret the message information) or even all if the verifier is the MM, which makes the whole scheme unsuitable for safety information applications. In our view, the authors should specify what VANETs applications can take advantage of their scheme.

Raya et al. (2006a) present a technique for secure group formation. Although the paper is centered on secure data aggregation it provides some insights in group formation techniques that could be used to increase privacy.

3.2.1. Achieving privacy through group signatures: how groups are formed

There are many ways to form groups in VANET applications. For example, all public transport buses can be members of a *preset group*. This is the easiest and most efficient way of group formation, but it requires prior knowledge of all group members, as well as a common authority over them. This is not the case when individual drivers on a highway decide to join a platoon in order to improve their driving experience. This requires *on-the-fly group* formation where a group leader is elected and group membership is managed dynamically. This latter category of groups is the most useful due to its flexibility, but it is also the most difficult to implement due to several issues, such as group leader election, group overlap, and the related security hurdles.

Raya et al. (2006a) introduce the concept of location-based groups, where the roads are divided into small area cells that define the groups. In this fashion, a vehicle will automatically

know to which group it belongs, the group leader will be by definition the closest vehicle to the center of the cell and naturally, it will be elected dynamically. It should be noted that, in the leader election process, vehicles do not broadcast their real identities but rather pseudonyms for privacy purposes, so the authors combine the use of groups with the use of pseudonyms for intra-cluster privacy.

On the plus side of this proposal, the group formation process is simplified and when using geographic routing determining which groups should relay messages is straightforward. However, for an attacker to always be elected group leader will suffice to place himself in the center of the cell permanently.

Vehicles periodically broadcast their public keys, so upon the formation of the group or whenever a new vehicle A joins the group, the leader L broadcasts the group key encrypted with the node's public key followed by its signature:

$$L \rightarrow A : \{K\}_{PuK_A} Sig_{PrK_L}[\{K\}_{PuK_A}] \quad (3)$$

This technique leaves room for improvement if the vehicles travel together in platoon formation, since the platoon may span over more than one cell.

Also Raya et al. (2006a) propose another solution named *Dynamic Group Key Creation*. The key idea is that once the leader and members of the group are identified, the leader creates a key request message that transmits to the CA. The CA will use that information to generate an asymmetric group key pair and broadcast it to all the group members. The key pair will be encrypted with the symmetric group key included in the key request message. In addition, the CA assigns to each group member a unique ID for non-repudiation purposes. Finally, once the asymmetric group key is established, any group member can send a message signed on behalf of the group (although accompanied by its certificate issued by the CA to allow the receivers to verify the signature). The message also includes the unique ID assigned by the CA to the group member that sent the message, which implies that the privacy of the individual vehicle is broken. Note, however, that the objective of the work reported in Raya et al. (2006a) is to reduce the overhead with *data aggregation* and does not explicitly address the problem of privacy.

3.3. Achieving privacy through pseudonyms

Pseudonymous authentication is widely accepted in the VANET community (Gerlach et al., 2007; IEEE, 2007; Calandriello et al., 2007; Papadimitratos et al., 2007), especially as an alternative to anonymous authentication, which can incur in additional overhead (Raya and Hubaux, 2005a; National Highway Traffic Safety Administration, 2006).

The work reported in Gerlach et al. (2007) presents a security architecture organized in layers. While the lowest layer is concerned with vehicle application registration and identification, higher layers are concerned with proper system operation, appropriate security measures and user privacy protection. In this group of higher layers we can find the *pseudonym* and the *revocation layer*.

The *pseudonym* layer provides a basic level of anonymity by introducing the possibility to use changing pseudonyms that cannot be linked by unauthorized parties. As pointed out by the authors, pseudonyms shall perform the same roles as the certificate issued for the node. This scheme uses dynamic pseudonyms to provide privacy, while at the same time an *Escrow Authority* (EA) is responsible for revoking and uncovering the user's real identity, if required.

The *revocation* layer is responsible for excluding nodes from the system. It contains a database of revoked pseudonyms and distributes this data to all nodes in the system if necessary, depending on the scale of the revocation decision, which can range from only node-local to system-wide.

We should note that when a node is revoked, all its pseudonyms are included in the revocation data. The authors do not specify how frequently pseudonyms should be changed or how large the pool of pseudonyms should be, however it is clear that there is a scalability problem.

From the system architecture perspective, the following entities are required:

- the vehicle manufacturer and the registration authority for the registration of nodes,
- the inspection site for test and certification of nodes,
- the "Escrow Authorities", entities with the authoritative power (e.g., police and courts) to identify and revoke nodes,
- the communication security infrastructure, which includes the communication systems, processing and databases necessary to carry out online testing, pseudonym provision for nodes, revocation of nodes and infrastructure based data assessment and intrusion handling.

As far as operation is concerned, vehicles use the certificate issued at the inspection site to request pseudonyms, which will be used to sign application messages. It is important to note that the scheme assumes sporadic access to the infrastructure. Some modules, such as the pseudonym provider may need reliable and on-demand connectivity, which could be provided by cellular technologies. As discussed in Raya et al. (2006b), distributing revocation information can also be achieved by simple terrestrial broadcast.

Calandriello et al. (2007) go a step further and combine the use of *pseudonyms* and *group signatures*. They describe a scheme which relies on the concept of pseudonymous authentication, which they name *Baseline Pseudonyms* (BP). The novelty with respect to previous works presented in this section is that it allows on-the-fly generation of the nodes own pseudonyms using *Group Signatures*, which in combination with the BP approach they term *Hybrid Scheme*.

By BP we understand a system where each node (vehicle) V is equipped with a set of pseudonyms, that is, public keys certified by the CA without any information identifying V, where each pseudonym is used at most for a period τ and then discarded. For the i-th pseudonym K_v^i for node V, the CA provides a certificate $Cert_{CA}(K_v^i)$, which is simply a CA signature on the public key K_v^i. The private key k_v^i is used by the node to digitally sign messages. To enable message validation, the pseudonym and certificate of the signer are attached in each message. With $\sigma_{k_v^i}()$ denoting V's signature under its i-th pseudonym and m the signed message payload, the message format is

$$m,\ \sigma_{k_v^i}(m),\ K_v^i,\ Cert_{CA}(K_v^i) \quad (4)$$

The CA maintains a map of the long-term identity of V to the K_v^i set of pseudonyms provided to a node. When required, the CA can extract the signer's identity from a message.

Assuming the general availability of the public key of the CA, upon the reception of Msg. (4) a node validates $Cert_{CA}(K_v^i)$. It makes use of a CRL, assumed to be distributed to vehicles via the infrastructure, as described in Raya et al. (2006c). If K_v^i is not included in the CRL and the CA signature on K_v^i is valid the node validates $\sigma_{k_v^i}(m)$.

The main idea behind the *Hybrid Scheme* mentioned above is that each node V is equipped with a group signing key gsk_v and a group public key gpk_{CA}. Instead of protecting messages with the group signature, a node generates its own set of pseudonyms K_v^i (and corresponding private keys k_v^i), and uses gsk_v to generate a group signature $\Sigma_{CA,V}()$ on each pseudonym K_v^i.

Basically, the nodes generate and "self-certify" K_v^i using $\Sigma_{CA,V}()$, hence producing $Cert_{CA}^H(K_v^i)$. The H denotes the Hybrid scheme differentiating it from the BP certificate and the subscript CA

confirms that the certificate was generated by a legitimate node registered with the *CA*. Similar to Msg. (4) we have

$$m,\ \sigma_{k_v^i}(m),\ K_v^i,\ Cert_{CA}^H(K_v^i) \tag{5}$$

Upon the reception of Msg. (5) the group signature is validated using the gpk_{CA} and the CRL. In this case, in order to disclose the identity of a message sender an *open* operation on the $Cert_{CA}^H(K_v^i)$ group signature is necessary (Bellare et al., 2003, 2004).

In the article, the pseudonym lifetime τ is also considered. On the one hand, it makes the vehicles less traceable as it decreases. On the other hand, it negatively impacts on the size of *Revocation Lists* (RLs) and the revocation process performance. Varying τ from 60 down to 3 s the signing and verification costs are $4.6e-3$ and $2.3e-3$ s/msg respectively. Even though those timings may seem low at first glance, in a densely populated area with over 100 nodes within the range it may be a problem for a safety messaging application, as they themselves remark.

Raya and Hubaux (2005a) present an intuitive method to compute how often should an anonymous key or pseudonym be changed, adapting to the vehicle speed. Considering a tracking scenario where an attacker controls stationary base stations separated by a distance d_{att} and captures all the received safety messages. Assuming that the attacker can correlate two keys if the sender moves at a constant speed in the same direction on the same lane between two observation points.

Assuming the speed of the target V is v_t, its transmission range d_r and d_v is the distance over which a vehicle does not change its speed and lane (hence, the vulnerability window). As illustrated in Fig. 2, the vehicle's anonymity is vulnerable over a distance equal to d_v+2d_r. Which means that it is not worth changing the key over smaller distances since an observer can correlate keys with high probability. This defines the lower bound on the key changing interval T_{key} when $d_{att} \leq d_v+2d_r$:

$$min(T_{key}) = \frac{d_v+2d_r}{v_t}\ (\text{s}) \tag{6}$$

However, if $d_{att} > d_v+2d_r$, V can avoid being tracked by changing its key as long as it does not use the same key for a distance equal or longer than d_{att}. This in its turn defines the upper bound on the key changing interval:

$$max(T_{key}) = \frac{d_{att}}{v_t}\ (\text{s}) \tag{7}$$

Since V does not know d_{att}, but knows d_r and d_v, it can choose a value of T_{key} that is slightly larger than $min(T_{key})$. If we denote by r_m the message rate, one key should be used for at most:

$$N_{msg} = \lceil r_m \times T_{key} \rceil (\text{messages}) \tag{8}$$

For instance, assume $d_{att}=2$ km, $r_m=3.33$ msg/s (1 message every 300 ms), $d_v=30\text{ s} \times v_t$ (i.e. V does not change its lane and speed over 30 s), $d_r=10\text{ s} \times v_t$ (according to DSRC, the transmission range is equal to the distance traveled in 10 s at the current speed), and $v_t=100$ km/h. Then $\min(T_{key})=50$ s and $\max(T_{key})=72$ s. V can choose T_{key} to be 55 s; as a result, $N_{msg}=184$ messages.

Rass et al. (2008) elaborate on the idea of using a pseudonym for a trip and then deriving several pseudonyms from it to use in the messages (sample identifier). They explicitly want the sample identifiers to be relatable to the trip identifiers, and at the same time different trip identifiers should also be relatable among themselves if a trip becomes interrupted by events like pauses or leaving and entering the highways with rural roads in between.

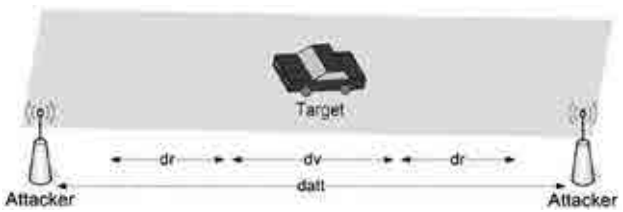


Fig. 2. Attack scenario.

Sampigethaya et al. (2005), Huang et al. (2005) and Chaurasia and Verma (2008) introduce the idea of a silent period between key changes, although each one with their own particular approach.

For instance, Chaurasia and Verma (2008) claim that in order to maximize anonymity, a moving vehicle V needs to continually observe the number of neighbors that are communicating in its vicinity. Then, after a pseudonym update a vehicle does not actually change its pseudonym and start sending messages with it for a short fixed period of time. After that period V observes the number k of communicating neighbors and only if k is greater than a predefined threshold τV transmits with the updated pseudonym. Otherwise, it remains silent.

The approach above is not suited for safety message applications. If the vehicles in the VANET need to periodically broadcast safety messages for cooperative navigation, then the period between those broadcasts will be the maximum time a vehicle can remain silent, which needs to be quite small (order of hundred milliseconds, Sampigethaya et al., 2005) regardless of the number of neighbors. Sampigethaya et al. (2005) introduce the use of a random silent period between the update of pseudonyms. They propose that vehicles form groups and that a group leader is elected. That group leader acts as a proxy for the rest of vehicles in the group for V2I communications, so that the rest of nodes in the group can remain silent for a longer period of time. Nevertheless, they direct this scheme to *Location Based Services* (LBS)[1] and not to safety message applications.

Opposed to the use of silent periods between pseudonyms update are the *Mix-Zones* (MZs) described in Dtzer (2006). Basically, in an MZ all the vehicles in a certain zone agree to change their pseudonyms at the same time, which according to the author makes any attempt to trace a certain vehicle V nearly impossible (provided that enough nodes are in that particular zone). However, this technique is also faulted for safety message applications for the very same reasons described for the previous technique.

Similarly, Gerlach et al. (2007) introduce *Context Mixes*, where vehicles only change their pseudonym if they consider it is safe, i.e., they have enough neighbors.

Contrary to the widespread belief that changing pseudonyms protects vehicles privacy, Wiedersheim et al. (2010) conclude that use of multiple pseudonyms may not be enough. Using *Multiple Hypothesis Tracking* (MHT) (Reid, 1979) and considering an attacker model where the attacker has the capability to capture all beacons sent to the network, they conclude that in a scenario with vehicles sending beacon messages at 1 Hz, changing their pseudonyms every 10 s and considering an equipment rate of 20% (rate of vehicles equipped with OBUs) an attacker can effectively track vehicles with an accuracy of almost 100%.

3.4. Achieving privacy through PKI: managing certificate revocation

PKI is a widely accepted solution (Raya et al., 2006b; Lin et al., 2007; Calandriello et al., 2007; Papadimitratos et al., 2008) as stated by the IEEE 1609 family of standards for *Wireless Access in Vehicular Environments* (WAVE) (IEEE, 2006). Vehicles in the network need the appropriate certificates in order to participate

[1] LBS make use of the vehicle position to provide a service, for instance finding the nearest hospital.

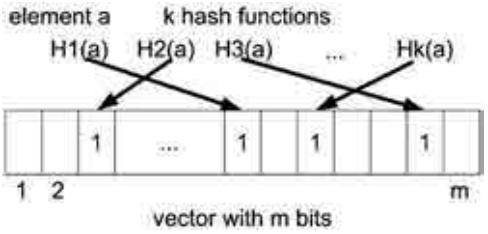


Fig. 3. Bloom filter.

in the system operation. Nevertheless, the certificates should only be valid for limited periods of time after their generation and the CA should reserve the right to revoke any nodes' certificates, essentially evicting them from the network. In several articles (Raya and Hubaux, 2005a; Raya et al., 2006b, 2007), it is accepted that vehicles will carry a trusted component or *Tamper Proof Device* (TPD) where the keys and certificates for network operation are stored and protected.

One of the main concerns of using PKI systems is managing the CRLs, with millions of users in the system, the potential size of the CRL is huge. Raya et al. (2006b, 2007) present a way to compress CRLs using Bloom filters (Bloom, 1970). The main characteristic of Bloom filters is that they return a configurable rate of false positives, but there are no false negatives (if the Bloom filter claims that an element is not in the set, we can be sure it is not). A Bloom filter (Fig. 3) consists of a sequence of m bits, initially all set to zero. A key or element can be included in the filter by hashing it with a specific number k of independent hash-functions (each ranging from 1 to m) and by setting to 1 the vector bits that are set to 1 in the result. After having added several keys to the filter, it is certainly possible that one bit is set to 1 multiple times. To check if an element is contained in the filter, the element is hashed and the status of the corresponding bits is checked. If at least one bit that should be one is not, one can surely affirm that the element is not contained in the filter. On the other hand, if all necessary bits equal 1, with high probability the element is included. However, it may also be possible that the bits were set to 1 by a combination of several other keys, as explained before. Therefore, the more elements added to the set, the larger the probability of false positives. Alternatively, Papadimitratos et al. (2008) take advantage of the multi-tier (regional) CAs setup to decrease the size of the CRLs. Regional CAs will only manage the certificates of vehicles in their region.

Studer et al. (2009) propose a scheme based on *Temporary Anonymous Certified Keys* (TACK), used to authenticate messages sent by the vehicles, whose CRL size is linear in terms of the number of revoked vehicles and unrelated to the size of the vehicle anonymous certificate set. There are three main entities:

- M: managing authority that acts as the root of trust.
- R: set of valid *Regional Authorities* (RAs). RAs act as intermediary authorities and can grant vehicles temporary region-specific certificates. M issues certificates to RAs and certifies them to be valid intermediary authorities.
- V: set of valid vehicles or *On Board Units* (OBUs). Any vehicle with a valid certificate from M or a region-specific short-lived certificate from R (while in the proper region) is considered part of V.
- $\neg V$: set of expired or revoked vehicles.

The main idea is to apply group signatures considering a group which comprises all of the above described entities. M is defined as the group manager. It initializes the group signature scheme to generate a group public key *gpk* and a group master key *gmk*. It publishes *gpk* and retains *gmk* for itself. Each valid OBU has a group user key guk_i, issued by M, which is installed during annual vehicle inspections. It should be noted that M maintains a history of all key/OBU pairs it has issued, so that it can later trace misbehaving vehicles. When a vehicle enters a new region it needs to update its TACK following these steps:

1. Randomly select new short-lived public and private keys from the key space (K_S^+, K_S^-).
2. Use the group user key guk_i to sign K_S^+ and send it to the RA.
3. RA verifies that the user is not in the RL. If it is not, the RA signs a certificate for the OBU's TACK public key K_S^+ using the RA's secret signing key K_{RA}^{-1}.
4. RA waits for δ seconds to queue up all certificate requests for that region and broadcasts the certificates.

Whenever a user wants to send a message it signs it with its TACK private key K_S^{-1} and periodically broadcasts the RA signed certificate of its TACK public key K_S^{-1}. Whenever a user misbehaves, to determine which OBU generated a signature ψ the group manager tests ψ against the group user keys of OBUs in V. Once M identifies V_i it is added to the RL and distributed to the RAs.

Similarly, Sun et al. (2010) try to achieve the same small CRL size with a pseudonymous authentication scheme. The network architecture is composed by a *Trusted Authority* (TA), RSUs and vehicles or OBUs. The TA issues a certificate $Cert_{TA,R_x}$ for a certain RSU R_x, and a series of pseudonymous certificates for a vehicle V_i to be installed during periodic vehicle inspections. It should be noted that the identities in the pseudonyms certificates are derived from two random seeds using a one-way hash function. The TA divides the maximum time between vehicle inspections into time windows. For every window, the TA chooses a random secret key to sign the vehicle's pseudonymous certificates, so that in every window the vehicle has to request R_x to re-sign the pseudonymous certificate for that window. In this scenario, an RSU can be revoked by including its only certificate in a CRL. To revoke a vehicle it would suffice for the TA to release the random seeds from which V_i's pseudonymous identities are computed, so that the RSUs do not issue the re-signature key to V_i in following windows. At the same time the valid pseudonymous certificate of V_i should be revoked.

Nowatkowski and Owen (2010) define *Most Pieces Broadcast* (MPB) technique to distribute CRLs. The first step is to break the large CRL file down into small pieces, taking into consideration the coding rate (rate of pieces generated from a file) and the code overhead (number of pieces needed to recover the original file). MPB ensures that only the node with the largest number of pieces broadcasts in a certain area to maximize the use of the wireless channel. It should be noted that RSUs will always be selected as the node with most pieces. The authors show that MPB is more effective than letting all OBUs broadcast their CRL pieces without control, which results in a broadcast storm of unneeded CRL pieces that slows down the CRL distribution.

3.5. Position

Three main techniques for achieving privacy have been discussed in this section: *anonymous certificates*, *group signatures* and *pseudonyms/pseudonymous certificates*. All these techniques have been widely studied throughout the literature and from our point of view are mature enough. The use of these techniques (or a combination of them, as we have seen) in VANETs is generally justified by the fact that they contribute to the users' privacy. However, by taking a closer look at the methods described in this section we realize that in order to keep the users' identity traceable under some circumstances those methods need PKI. Therefore, the need for revoking certificates and managing large CRLs. It has been shown that applications may face that particular

problem in different ways. Some may appoint certain nodes as message verifiers and they will be the only ones working with CRLs. In the global picture, that could give the impression of efficiency (since the amount of nodes repeating work decreases) although that is certainly not a good idea for safety message applications because most of the network works blindfolded. On the other hand, some other schemes may apply techniques to compress the CRLs like Bloom filters or to directly reduce the amount of certificates that need to be revoked, and thus included in the CRL. We believe extensive effort will be dedicated to reduce the CRL size as done in Sun et al. (2010) and Studer et al. (2009) and to study the most efficient ways to distribute it (Nowatkowski and Owen, 2010).

Special mention deserves the work of Wiedersheim et al. (2010) for considering the effectiveness of pseudonyms change. Privacy is a major concern in VANETs security, and so far the use of pseudonyms seemed to be a perfect solution for the traceability problem. We believe that extensive research should be performed to verify if the authors' claim of complete traceability holds for equipment rates higher than 20%.

4. Detection and eviction of misbehaving and faulty nodes

In the previous section we have focused on schemes that provide a secure and reliable network and try to keep attackers from disrupting its normal operation. However, due to the attackers ability or just to the devices aging process at some point in time there will be misbehaving or faulty nodes in the VANET. That is why in this section we outline several techniques to detect and evict them from the network (Table 3).

Golle et al. (2004) develop a heuristic called *adversarial parsimony*, which informally means finding the best explanation for corrupted data. The first step is to enhance the vehicles sensing capabilities giving them physical means to distinguish its neighbors, for instance with cameras or exchanging information in the infrared light spectrum to verify that a vehicle is where it claims to be, thus preventing sybil attacks. That information needs to be exchanged between vehicles, and once enough evidence has been collected the heuristic will find inconsistencies, if any. For instance, if there is a group of nodes that are linked to the rest of the network by only one node then that link node is probably impersonating all the others.

Xiao et al. (2006) present a solution to reliably detect sybil attacks based on radio signal strength analysis and on the fact that a vehicle cannot be on different places at the same time. For clarity of description, they define three categories or roles:

1. *Claimer*: Each node periodically broadcasts a beacon message at beacon intervals t_b for the purpose of neighbor discovery. In the beacon message, it claims its identity and position. The goal of the scheme is to verify its claimed position.
2. *Witness*: All neighboring nodes, within the signal range of the claimer, would receive the previous beacon message. They measure the signal strength and save the corresponding neighbor information in their memory. Next time they broadcast a beacon message, they will attach their neighbor list including the signal strength measurements.
3. *Verifier*: After receiving a beacon message, a node waits for a verifying interval t_v during which it collects enough signal strength measurements concerning the previous beacon message from neighboring witnesses. With the collected measurements, the node can locally compute an estimated position for the claimer, for instance, by performing *Minimum Mean-Square Error* (MMSE). However, to be as accurate as possible, before actually making the computations to locate the sender of a message the node needs to discard all the signal strength information that comes from sybil nodes.

In order to discard sybil nodes information they rely on two principles or rules.

1. *Rule* 1: An RSU or *Base Station* (BS) issues a position certification for each vehicle passing by. The position certification contains a time stamp and a location information of the BS and therefore can prove the presence of the vehicle near the base station at a certain time.
2. *Rule* 2: All witnesses for a claimer should consist of vehicles in the opposite traffic flow.

With Rule 1, we can ensure where a certain vehicle comes from. Looking at Fig. 4 node *a* can get a position certification from *BS2*, when passing by *BS2*, and node *b* can also get one from *BS1*. When *a* and *b* meet each other, it is easy for them to prove that they come from opposite directions by exchanging certificates. With Rule 2, we can ensure that each witness in the opposite traffic flow is a physical vehicle instead of a sybil one. For instance, suppose that a malicious node *m* fabricates seven sybil nodes, in which s_7 is traveling in the opposite direction and the rest in the same. When trying to verify the positions of $s_1, \ldots, s_6$, node s_7 would be ignored because it cannot prove that it comes from the upstream of the road.

On the whole, with the help of roadside infrastructure, dishonest sybil nodes can be detected through position verification.

Raya et al. (2007) rely on the vehicle's TPD to execute their protocol and even revoke itself if it detects it has been tampered with. They also assume the existence of a honest majority in the attacker's neighborhood. Unfortunately, TPDs usually end up becoming just Tampered Devices, as shown in Anderson and Kuhn (1996, 1997) and Biham and Shamir (1997). Therefore, an attacker could just modify their programming to impersonate several vehicles (Sybil attack) (Douceur and Donath, 2002), rendering the honest majority hypothesis invalid. And even if the TPD remained tamper-proof nothing can stop an attacker from actually stealing the physical device from another car and once again mount a Sybil attack. Nevertheless, the authors devise

Table 3
Taxonomy of misbehavior protection schemes.

	Tamper proof device	Requires certification authority	Honest majority	Sybil attack protection
Golle et al. (2004)			X	X
Xiao et al. (2006)		X	X	X
Raya et al. (2007)	X	X	X	
Moore et al. (2008b)		X	X	
Ghosh et al. (2010)				X

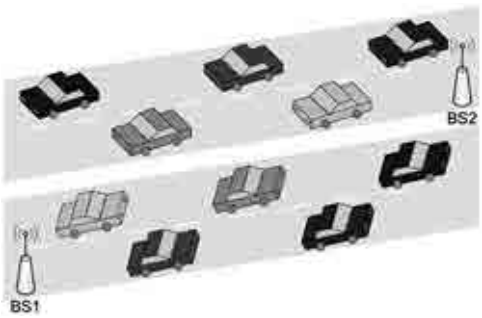


Fig. 4. A scenario with roadside infrastructures.

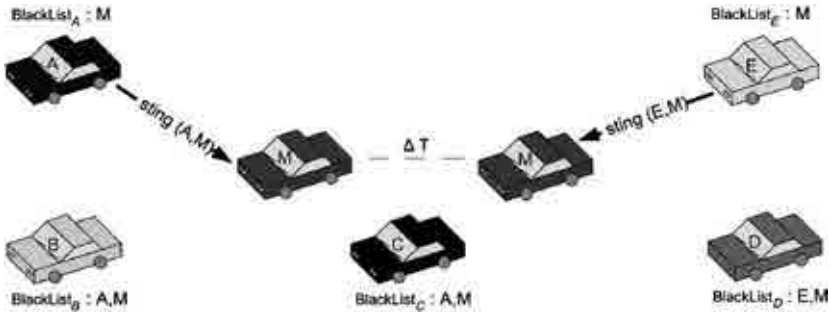


Fig. 5. Multiple stings for misbehaving node *M* as it moves over time.

a *Misbehavior Detection System* (MDS) as well as a *Local Eviction of Attackers by Voting Evaluators* (LEAVE) protocol to detect and exclude misbehaving nodes.

MDS basically consists in each node using its own sensory inputs, messages received from its neighbors and a set of evaluation rules to classify the received safety messages from a given node as faulty or correct. Messages that are outdated, received beyond their theoretical area of propagation or contradictory to the node's own state are considered false. Their senders, as long as they are neighbors of the node running MDS are tagged as misbehaving and their identity is passed on to LEAVE.

The main principle of LEAVE is simple: the neighbors of a misbehaving vehicle should temporarily evict it. It should be noted that the system does not require a permanent connection to the CA to work, as we will see below. It is not a revocation protocol, but rather a collective warning system against misbehaving nodes. Upon detecting an attacker, vehicles broadcast warning messages to all vehicles in range, so that the sharing of information improves the effectiveness of the stand-alone detection system. Besides, those warnings can be very valuable when vehicles receive them even before being able to observe the misbehaving node themselves. Any vehicle receiving a warning message adds the warned device to an accusation list, and once enough warning votes against a node are collected, its identifier is added to a local blacklist. After entering the blacklist, *disregard* messages are repeatedly broadcasted to the local neighborhood to ignore the attacker's messages. The eviction is temporarily limited to the duration of the contact between the attacker and its neighbors running LEAVE. However, once the connection to the CA is re-established a global-scale revocation protocol can be initiated.

Moore et al. (2008b) devise another scheme based on *suicide attacks* (Moore et al., 2008a) called Stinger, which also relies on a honest majority. In a nutshell, should a node believe another one has misbehaved it will send a message that will evict them both from the network. The idea is to make the sacrifice of future participation so costly that discourages false accusations. Stinger deviates from a suicide attack in the following aspects:

1. *Stinger temporarily prohibits devices from transmitting messages, but allows them to continue to receive and forward messages.* Temporary removal could be used to rapidly ignore an errant transmitter. The authors assume that most interactions are short-lived and therefore temporary removal is as effective as permanent removal in tackling misbehavior. While the sting instruction prevents the bad and the good device from sending out additional warnings, both will still receive safety instructions from other cars. The authors claim that this solution minimizes the noticeable impact on the sacrificing vehicle while still penalizing a malicious device. However, in our view, when considering safety message applications the noticeable impact is indeed noticeable since the accusing nodes will not be able to send the information collected by their own sensors.
2. *Stinger does not allow more than one node to sacrifice itself for a misbehaving one (in a local context).* Figure 5 illustrates how the protocol works as the cars move. Misbehaving node *M* is detected by *A*, which broadcasts $sting_{A,M}$ to indicate vehicles near *A* to ignore *M*. Hence, nodes *B* and *C* add both *A* and *M* to their local blacklists, while *D* and *E* do not because they did not receive the sting message. As *M* moves into range of *D* and *E*, *E* issues a new removal for *M*, $sting_{E,M}$. *D* adds *E* and *M* to its local blacklist, but *C* does not because it has already ignored *M* from *A*'s sting.
3. *Stinger permits good devices to continue to accuse bad ones even after having issued one sting.* The authors claim this last condition to be necessary to prevent the so-called *motorway attacker* who widely broadcasts misbehavior and moves around quickly to attract many stings and prevent good nodes from excluding subsequent attackers. However, we think the *motorway attack* is still possible just by doing the exact opposite. Since good nodes are always allowed to accuse misbehaving ones, it would suffice for the attacker to move around accusing good nodes instead of misbehaving himself. By using scheme like MDS (described at the beginning of the section) which had only local visibility (a node only gathers information about its neighbors), there would be no possible way for a group of nodes that encountered the attacker for the first time to identify him as such because of something he had done in the previous group. This could be solved by a misbehavior detection system which had a global vision on all the groups in the network.

Ghosh et al. (2010) present a system, which just like MDS, uses its sensory input to detect misbehavior. After receiving an alert message, a vehicle *V* compares the sensed behavior of the surrounding cars with a model of expected behavior under that kind of alert and analyzes how it deviates. For example, if the vehicles ahead of *V* start slowing down after he has received an alert message claiming there has been accident that will match the expected behavior. This kind of techniques could help an OBU to determine whether alert messages are true or not, but they require fine tuned models of expected behaviors for each of the possible alerts. Something we believe to be unfeasible given the large number of possible alert situations.

4.1. Position

Several of the articles covered in this section introduce schemes designed to evict nodes from a VANET while there is no direct connection to the CA, problem that could be easily solved using cellular technologies to establish that link (as described in Section 2). As seen in Xiao et al. (2006), Raya et al. (2007) and Ghosh et al. (2010), roadside infrastructure and enhancing the vehicles sensing capabilities are valuable assets to verify other vehicle's messages and prevent Sybil attacks. In our view, preventing multiple identity attacks is of paramount

importance to protect the honest majority hypothesis on which so many protocols rely. However, we foresee that approaches following Ghosh et al. (2010) will be very seldom used since the generation of models of expected sensed behaviors for each of the possible alerts with a reasonably low rate of false positives seems to be a daunting task if feasible at all.

5. Techniques for secure data aggregation

One way to use available bandwidth more efficiently is to aggregate the information of several vehicles into a single message or record, as done in the V2V traffic information system described in Nadeem et al. (2004), where vehicles share information about each other. Data aggregation shall be able to aggregate events according to temporal and spatial dimensions. Moreover, filtering old reports is an essential part of any aggregation scheme. Thus, any aggregated record has to include an expiration time after which the information is no longer valid. More difficult is the definition of spatiality. In terms of aggregation, the key question is how far a primary record (i.e., an original record) can participate in an aggregation process.

Scheuermann et al. (2009b) prove that any successful aggregation scheme must reduce the bandwidth at which information about an area at distance d is provided to the cars asymptotically faster than d^2. In their scheme, data aggregation is originated at *measurement points* (Scheuermann et al., 2009b) and goes to *destinations* (i.e., set of vehicles that are interested in information from *a measurement point*). Many data aggregation schemes consider *measurement points* as specific areas that can be fixed (e.g., a road segment) or dynamic (e.g., based on the location of a set of vehicles). Other schemes consider groups of vehicles called *clusters* with a specific vehicle, *the cluster-head*, in charge of aggregating the primary reports. Clusters can be organized based on their fixed geographical area or can be dynamically formed by mobile vehicles. Furthermore, according to Picconi et al. (2006) data aggregation in VANETs can be classified as syntactic and semantic. *Syntactic aggregation* compresses data from multiple vehicles in order to fit the data in a unique record or frame. For example, an application that extracts a subset of each individual record and adds it to a single record is reducing the original information. *Semantic aggregation* means that the data from individual vehicles is summarized. For example, an application that instead of sending the location of each vehicle, only reports the number of vehicles in a given area.

Aggregation, however, aggravates the security problem. A malicious aggregator may send aggregated records that do not correspond to real data. For instance, it may falsely report a congested road by pretending to have aggregated more records than it has actually received from cars ahead of it. *Secure Data Aggregation (SDA)* aims to ensure the integrity of the data aggregation mechanisms in the presence of malicious nodes that can alter the result of the aggregation. Forging or suppressing a single record can have low impact in both syntactic and semantic aggregation. Thus, the main threat (Raya et al., 2006a), in SDA is the generation of false aggregation information. Secure data aggregation is a topic well studied in sensor networks. However, due to the mobility nature of vehicular ad hoc networks and the fact that nodes move following specific paths, the re-use of wireless sensor network SDA mechanisms is not possible in VANETs.

Dietzel et al. (2010) propose the following generic aggregate structure for SDA schemes:

$$A = \left[\underbrace{(a_1,b_1),\ \ldots,(a_n,b_n)}_{\textit{index-dimensions}} \middle| \underbrace{(v_1,\ \ldots,\ v_p)}_{\textit{values}} \middle| \underbrace{(m_1,\ \ldots,\ m_p)}_{\textit{meta-inform.}} \right] \quad (9)$$

Table 4
Taxonomy of secure data aggregation (SDA) schemes.

	Syntactic	Semantic	Cluster based	Fixed/dynamic areas
Picconi et al. (2006)	X	X		D
Raya et al. (2006a)		X	X	F
Yu et al. (2008)		X		F
Ibrahim and Weigle (2008)	X		X	F
Dietzel et al. (2009)	X			F
Dietzel et al. (2010)	X			D
Lochert et al. (2010)		X		D

where the index dimensions indicate the area and time about which an aggregate contains information. The values are the information and the meta-information contains that additional information used in security mechanisms. In general, most of the SDA proposals found in the literature follow similar structures, although there is not a consensus in a well defined aggregated structure.

Table 4 summarizes the SDA schemes covered in this section and classifies them according to whether a scheme (i) performs syntactic or semantic aggregation, (ii) is cluster-based (cluster-head responsible for aggregating reports) and (iii) is defined for fixed or dynamic geographical areas.

Picconi et al. (2006) propose a technique to probabilistically detect malicious vehicles that generate false aggregated information. In particular, they focus on validating speed and location information using syntactic aggregation although their solution is also applicable to certain cases of semantic aggregation. The proposal targets aggregated information from a measurement point to a destination, without the need of creating groups or clusters of vehicles. The main idea behind this scheme is to challenge the aggregator to provide a proof that can be used to probabilistically validate the aggregated record. An aggregated record is created by combining and compressing information contained inside several individual records. To validate the aggregated record the aggregator is asked to provide a randomly chosen original signed record (whose information was included in the aggregated record) after the aggregated record has been sent. If the corresponding record was made up it will not be possible for the aggregator to produce the original signed record, and he will be caught. It should be noted that the probability of a misbehaving node being caught is directly proportional to the amount of bogus information it includes in the aggregated record and that for the system to work the penalty needs to be severe enough to discourage misbehavior (e.g. permanent eviction from the network).

In order to avoid a two-phase protocol, vehicles are equipped with a *Tamper Proof Device* (TPD) which acts as a proxy for the receiver. As a proxy, it first provides a transmit buffer (data placed on this buffer cannot be tampered with and will be transmitted) and second it challenges the application (aggregator) to provide a randomly chosen original signed record to be sent with the aggregated data. The whole process can be observed in Fig. 6. The application extracts the data and car ID from each regular record (a) and places it in the transmit buffer where the TPD appends a secure time stamp and the randomly generated number 83 (b). The application takes the regular record corresponding to entry $i=(83 \bmod 3)=2$ (i.e. the third entry) (c) and appends it to the transmit buffer. Finally, the TPD signs R1+R2 (d) and broadcasts the contents of the transmit buffer (e).

Even though this method may indeed prove itself to be effective against malicious aggregators who try to insert false information in

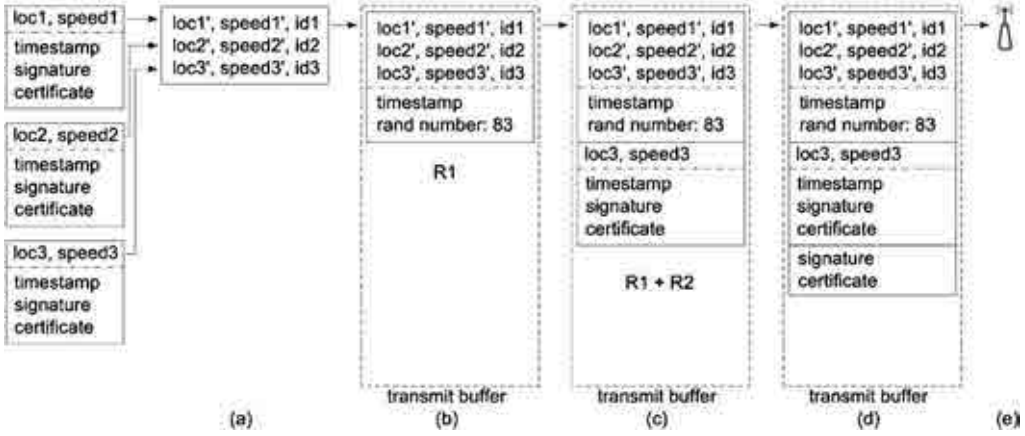


Fig. 6. Secure aggregation using the TPD as a proxy for the receiver.

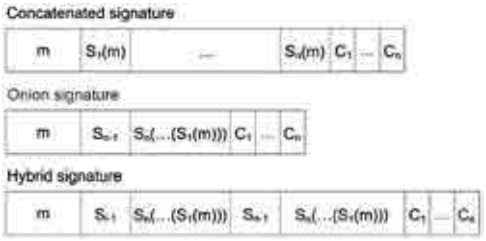


Fig. 7. Three different types of combined signatures. n is the total number of signers. C_i is the certificate of i-th user.

the network, it leaves the vehicles unprotected from malicious aggregators that leave out information from the aggregated records. In our view, the TPD could also serve as the entry point for received records and it should keep track that the vehicle identities in the received messages at some point before an upper bound τ are included in an aggregated message to be broadcasted.

Raya et al. (2006a) claim that bandwidth efficiency can be achieved using combined signature techniques. The authors address secure group formation, where each group is composed by those vehicles in a specified geographical area or *cell*. The group leader is chosen as that one closest to the center of the cell. Thus, the group leader is in charge of aggregating and disseminating data. Group leaders receive signed reports from vehicles creating a new message with a combined signature. Therefore, combined signatures is a semantic SDA mechanism since there is only one message m signed by the combination of all vehicles that participate in the event detection (Fig. 7). The following combined signatures are proposed:

1. *Concatenated Signatures*: The idea behind this scheme is that whenever a vehicle receives a message if it agrees with the message information (based on its own sensors input) it appends its signature. This form of source aggregation results in a smaller data verification delay than destination aggregations where the receiver collects messages from different sources and then crosschecks them. Another advantage is that an invalid signature does not affect the whole message, in contrast to the next scheme.
2. *Onion Signatures*: The signature sizes are constant, since each message is hashed before being signed. Instead of simply appending a new signature, a vehicle signs the signature of the previous transmitter, although before retransmitting the new message, it should also include the last signature, i.e., the one it received, so that the vehicle at the next hope can verify the previous signature. The improvement in signature size comes at a cost. In this case, a single invalid signature will affect the whole message and the message needs to be verified at each hop, increasing the overall verification time. In our view, this last feature if correctly exploited could lead to a denial of service attack.
3. *Hybrid Signatures*: Consists of several concatenated onion signatures, each of a given depth. The signature depth representing the number of layers it includes. This solution looks for a compromise between the previous two, both on their advantages and their drawbacks.

We find that *Hybrid Signatures* are a very interesting possibility to explore when considering safety message applications. We propose that the different kinds of safety messages of the application be assigned a degree of time criticality and needed trust and depending on those values the appropriate depth of the *Hybrid Signature* be chosen. For example, if a vehicle is on a crossing with no visibility on the right side of the road we can safely assume its driver will not mind waiting a few seconds before it can safely traverse. Therefore, in that case the better suited solution would probably be a *Hybrid Signature* with depth 0.

In Catch-up (Yu et al., 2008), the authors propose an aggregation scheme for applications where a delay of tens of seconds is acceptable, not suited for safety messaging applications but perfectly valid for general traffic information. Aggregation is performed in road sections for the same frame interval of time. Their objective is to perform semantic aggregation by generating a single secure report with aggregation functions such as MAX, MIN, AVG. Any vehicle can aggregate the data and thus there is not any cluster structure created. The basic idea in this scheme is to insert a delay before forwarding a report to the next hop. However, their scheme makes this delay more controllable in order to increase the probability that a report can be merged with reports ahead or reports behind. Intelligent delay-control policies are made based on local observations of individual vehicles. They also design a future reward model to define the benefits of different delay-control policies, and then establish a decision tree to help a vehicle choose an optimal policy from the perspective of long-term rewards.

CASCADE (Ibrahim and Weigle, 2008) is a cluster-based syntactic SDA scheme. Each vehicle presents location information based on its difference from the location of the cluster's center and its speed based on its difference with the median speed of those vehicles in the cluster. The primary record is signed by the vehicle using ECDSA and includes a timestamp to prevent replay attacks and the vehicle's public key. Each vehicle, then builds its own local view from primary records. Records are grouped based on their distance from the receiving vehicle. First each data record

is compressed using differential encoding. Second, an aggregated cluster record is built which is the concatenation of compact data records (syntactic aggregation). The signature is calculated by the aggregating vehicle over all fields of the aggregated frame except the certificate which is signed by the CA and the sender's location that represents the last location of the last vehicle that broadcasted the record.

Dietzel et al. (2009) argue against fixed segmentation of roads because it contradicts the real situation. They propose a completely structure-free aggregation mechanism, which enables to aggregate data purely based on their correlation. On a conceptual level, all aggregation systems have the following basic components:

- *Decision criteria*: Decide if two pieces of information are similar enough to be aggregated.
- *Information fusion*: Once the decision to aggregate two data items has been reached, a defined method is require to combine them.
- *Dissemination mechanism*: Having aggregated to data items, the new information is only available to the aggregator. Thus, the node needs to disseminate the new data into the network.

The authors propose a fuzzy logic scheme to be used for the decision criteria, which allows a dynamic fragmentation of the road. First, all influences on the aggregation decision, i.e. location difference of two aggregates or a maximum standard tolerable deviation of the average speed values, are fuzzified by applying fuzzy set theory. Then, they use fuzzy logic operations to reason about the influences and reach a decision.

Dietzel et al. (2010) present a syntactic SDA scheme. The mechanism chooses a subset of all atomic primary reports to generate an aggregate report. The authors employ a list of criteria to selectively choose which primary reports contribute to the aggregate report. The criteria include the identification of those primary reports that led to an aggregate current maximum and minimum in time and space, defining a specific location area. The scheme is a cluster-based mechanism where the cluster borders are defined by the location of a subset of primary reports and those reports corresponding to vehicles inside the borders of the area will be selected to produce an evenly distribution that represents the whole area.

Lochert et al. (2010) introduce the concept of soft-state sketches for probabilistic hierarchical data aggregation, which derive from Flajolet–Martin sketches (FM sketches) defined in Flajolet and Martin (1985). An FM sketch is a data structure for probabilistic counting of distinct elements. It represents an approximation of a positive integer by a bit field $S = s_1, \ldots, s_w$ of length $w \geq 1$. The bit field is initialized to zero at all positions. To add an element x to the sketch (an observation), it is hashed by a hash function h with geometrically distributed positive integer output, where $P(h(x) = i) = 2^{-i}$. The entry $s_{h(x)}$ is then set to one. In soft-state sketches, the authors use small counters of n bits instead of single bits at each index position. These counters represent a time to live (TTL) for a certain bit. Therefore, the operation of setting a bit to one after an observation is replaced by setting the corresponding counter to the maximum TTL. An approximation $C(S)$ of the number of distinct elements added to the sketch can be obtained from the length of the initial, uninterrupted sequence of ones, given by

$$Z(S) \coloneqq min(i \in \mathbb{N}_0 \,|\, i < w \wedge s_{i+1} = 0 \cup \{w\}) \tag{10}$$

by calculating

$$C(S_1, \ldots, S_m) \coloneqq m \cdot \frac{2^{\sum_{i=1}^{m} Z(S_i)/m} - 2^{-K \cdot \sum_{i=1}^{m} Z(S_i)/m}}{\varphi} \tag{11}$$

with $\varphi \approx 0.77351$, $K \approx 1{,}75$ and using m sketches.

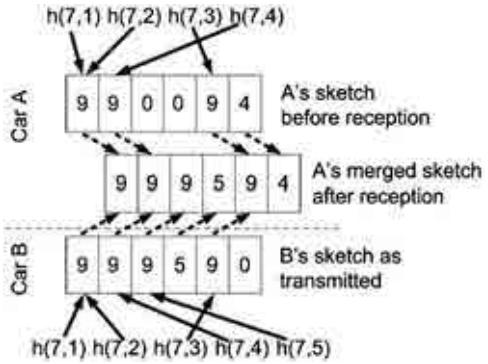


Fig. 8. Aggregation of soft-state sketches.

Locally stored sketches are periodically broadcasted to the vehicle's one-hop neighbors, which upon reception merges them with its own. For example, consider an application where the number of free parking spots on a road segment is disseminated in the network. Two cars, *A* and *B*, make independent observations on the same road segment (with ID 7). *A* observes four free parking places and thus hashes the tuples (7,1), …, (7,4) into its sketch for road 7. *B* observes five free parking places, and consequently adds (7,1), …, (7,5). If *A* and *B* meet they will exchange sketches, as depicted in Fig. 8 and perform a position-wise maximum operation. Previously inserted elements die out after their TTL has expired, unless they are refreshed by a newer observation.

5.1. Position

Information aggregation is a process of paramount importance in VANETs. Hundreds of vehicles transmitting information and relaying that very same information to other vehicles next to them in a multi-hop network. Besides considering the number of samples a vehicle takes every minute is enough to make us realize of the large traffic load involved in VANET applications (particularly in safety messaging). Therefore, if there is any way to decrease the network traffic load it should be exploited.

We find particularly relevant the contribution of Scheuermann et al. (2009a) where the authors give an analytical measure of scalable data aggregation schemes. We also consider *intelligent delay control* policies a field were extensive research needs to be performed, since they can help optimize the use of the wireless medium.

SDA schemes are defined according to whether the aggregation is syntactic or semantic and thus the proposed schemes are bounded on what kind of aggregation is performed. Furthermore, most of the schemes are bounded on whether the aggregation is performed in fixed or dynamic areas and who is the node that aggregates the information. A general framework for both semantic and syntactic aggregation would facilitate the definition of SDA for any kind of application and network topology. In this direction, Picconi et al. (2006) and Dietzel et al. (2010) are the ones that contribute to more general specifications.

Some SDAs make use of TPDs, such as Picconi et al. (2006), and the whole aggregation process depends on their correct behavior. As already discussed in this article, *Tamper Proof Devices* are not always as tamper-proof as they should be. Therefore, we consider protocols that place their security on TPDs to be inherently flawed.

6. Conclusions

In this paper we have surveyed the newest trends in the research area of VANET security. The proposals that we have

analyzed deal with trade-offs between complexity, response time, privacy and non-repudiation. In this section we conclude with a summary of which trends and approaches we foresee as the winners of this research area and why.

1. *Security*: Even though the use of PKI is widespread and generally recognized as a valid solution it still has several issues to address, like nodes revocation (and the CRL size) and privacy. Some of the pseudonymous authentication schemes presented here address both issues (Sun et al., 2010; Studer et al., 2009): the first by maintaining the CRL growth linear while the second is solved by the use of pseudonyms. Nonetheless, with millions of users in a VANET the CRL size is still considerably large. Therefore, we believe that the use of Bloom filters (Raya et al., 2006b, 2007) together with the pseudonyms approach could really make an impact in the field. However, it should also be noted that the use of pseudonyms may not be as secure as one may assume for low equipment rates. We feel there is still extensive research to be carried out in both directions: minimize the CRL size and find out if pseudonyms users are traceable for higher equipment rates.
2. *Misbehaving and faulty nodes*: Accepting that at some point the security mechanisms defending your network will be overcome is being realistic. That is why misbehavior detection and eviction protocols need to be part of every system. In our view, Sybil attacks together with Denial of Service are on the top of the most dangerous attacks a VANET can suffer. Being able to impersonate an unlimited number of vehicles increases the attacker's ability to disrupt network operation. That is the reason why we consider particularly important the solution proposed in Xiao et al. (2006) to detect multiple identities by using roadside infrastructure and enhancing the vehicles sensing capabilities.
3. *Data secure aggregation*: VANETs produce an enormous amount of data. In addition to every vehicle collecting its own information and broadcasting it, they also have to forward the others messages. Hence the need for aggregation. We consider particularly relevant the contribution of Scheuermann et al. (2009a) for providing an analytical method to study scalable aggregation schemes. However, as it has been said before, aggregation must be secure, since aggregating nodes can be tempted to include false messages or leave out valid ones. This is where the mechanisms described in this section need to be set into place. Moreover, in order to optimize the use of the wireless medium techniques like *intelligent delay control* policies, as described in Yu et al. (2008), can help improve the aggregation process.

Acknowledgments

This work was partially supported by the EuroNF NoE and by Spanish grant TIN2010-21378-C02-01.

References

Anderson R, Kuhn M. Tamper resistance—a cautionary note. In: Proceedings of the second Usenix workshop on electronic commerce; November 1996.

Anderson R, Kuhn M. Low cost attacks on tamper resistant devices. In: IWSP: International workshop on security protocols. Lecture notes in computer science; 1997.

Atenies G, Song D, Tsudik G. Quasi-efficient revocation of group signatures. In: Proceedings of the Finance cryptography; March 2002. p. 183–97.

Barreto M, Libert B, Quisquater J. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Proceedings of the advances in cryptology—Asiacrypt'05, vol. 3788, Springer-Verlag; 2005. p. 515–32.

Bellare M, Micciancio D, Warinschi B. Foundations of group signatures: formal definition, simplified requirements and a construction based on trapdoor permutations. In: Biham E, editor. Advances in cryptology—EUROCRYPT 2003. Proceedings of the international conference on the theory and application of cryptographic techniques. Lecture notes in computer science, vol. 2656. Warsaw, Poland: Springer-Verlag; May 2003. p. 614–29.

Bellare M, Shi H, Zhang C. Foundations of group signatures: the case of dynamic groups. In: CT-RSA'05. Lecture notes in computer science. Springer-Verlag; 2004. p. 136–53.

Biham E, Shamir A. Differential fault analysis of secret key cryptosystems. In: CRYPTO; 1997. p. 513–25.

Bloom BH. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM 1970;13(7):422–6.

Boneh D, Boyen X, Shacham H. Short group signatures. In: Proceedings of the Crypto 2004. Lecture notes in computer science, vol. 3152. Springer-Verlag; August 2004. p. 41–55.

Boneh D, Franklin MK. Identity-based encryption from the Weil pairing. In: CRYPTO '01: Proceedings of the 21st annual international cryptology conference on advances in cryptology. London, UK: Springer-Verlag; 2001. p. 213–29.

Boneh D, Shacham H. Group signatures with verifier-local revocation. In: Proceedings of the ACM CCS; October 2004. p. 166–77.

Calandriello G, Papadimitratos P, Hubaux J-P, Lioy A. Efficient and robust pseudonymous authentication in VANET. In: Proceedings of the ACM international workshop on vehicular ad hoc networks (VANET); September 2007. p. 19–28.

Chaurasia BK, Verma S. Maximizing anonymity of a vehicle through pseudonym updation. In: WICON '08: Proceedings of the 4th annual international conference on wireless internet. Brussels, Belgium; ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering); 2008. p. 1–6.

Chaum D, Van Heyst E. Group signatures. In: Advances in cryptology 1981–1997: EUROCRYPT '91, vol. 1440. Heidelberg: Springer; 1999. p. 127–33.

Dietzel S, Bako B, Schoch E, Kargl F. A fuzzy logic based approach for structure-free aggregation in vehicular ad-hoc networks. In: Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking. VANET '09. New York, NY, USA: ACM; 2009. p. 79–88. 〈http://doi.acm.org/10.1145/1614269.1614283〉.

Dietzel S, Schoch E, Konigs B, Weber M, Karl F. Resilient secure aggregation for vehicular networks. IEEE Network 2010;24(1):26–31.

Douceur J, Donath JS. The sybil attack; 2002. p. 251–60.

Dtzer F. Privacy issues in vehicular ad hoc networks. In: Privacy enhancing technologies. Lecture notes in computer science, vol. 3856. Berlin, Heidelberg: Springer; 2006. p. 197–209. 〈http://www.springerlink.com/content/r4v42078437x2720/〉.

Flajolet P, Martin GN. Probabilistic counting algorithms for data base applications. Journal of Computer and System Sciences 1985;31(2):182–209. 〈http://www.sciencedirect.com/science/article/B6WJ0-4B4RWGM-1S/2/dd76c13a6efb88d5c9d8d4384b348738〉.

Gerlach M, Festag A, Leinmüller T, Goldacker G, Harsch C. Security architecture for vehicular communication. In: Fourth international workshop on intelligent transportation (WIT 2007); 2007.

Ghosh M, Varghese A, Gupta A, Kherani AA, Muthaiah SN. Detecting misbehaviors in VANET with integrated root-cause analysis. Ad Hoc Networks 2010;8(7):778–90. 〈http://www.sciencedirect.com/science/article/B7576-4YK7J4W-1/2/9da4b4bbdbf4bb799ee35ff75e23c317〉.

Golle P, Greene D, Staddon J. Detecting and correcting malicious data in vanets. In: VANET '04: Proceedings of the 1st ACM international workshop on vehicular ad hoc networks. New York, NY, USA: ACM; 2004. p. 29–37.

Huang L, Matsuura K, Yamane H, Sezaki K. Enhancing wireless location privacy using silent period. In: Wireless communications and networking conference, vol. 2. IEEE; March 2005. p. 1187–92.

Ibrahim K, Weigle MC. Cascade: cluster-based accurate syntactic compression of aggregated data in vanets. In: Proceedings of the IEEE international workshop on automotive networking and applications (AutoNet); December 2008.

IEEE. Ieee p1609.2 version 1—standard for wireless access in vehicular environments: security services for applications and management messages; 2006.

IEEE. Ieee trial-use standard for wireless access in vehicular environments (wave)—networking services. IEEE Std 1609.3-2007, c1–87; 2007.

Kiayias A, Tsiounis Y, Yung M. Traceable signature. In: Proceedings of the advances in cryptology—Eurocrypt. Lecture notes in computer science, vol. 3027. Springer-Verlag; 2004. p. 571–89.

Laurendeau C, Barbeau M. Secure anonymous broadcasting in vehicular networks. In: 32nd IEEE conference on local computer networks, (LCN 2007); October 2007. p. 661–8.

Lin X, Sun X, Ho P-H, Shen X. Gsis: a secure and privacy-preserving protocol for vehicular communications. IEEE Transactions on Vehicular Technology 2007;56(6):3442–56.

Lochert C, Scheuermann B, Mauve M. A probabilistic method for cooperative hierarchical aggregation of data in vanets. Ad Hoc Networks 2010;8(5):518–30. Vehicular Networks. 〈http://www.sciencedirect.com/science/article/B7576-4Y52R2F-2/2/b6957762453f8723a9658de431b975eb〉.

Moore T, Clulow J, Nagaraja S, Anderson R. New strategies for revocation in ad-hoc networks; 2008a. 〈http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.65.6831〉.

Moore T, Raya M, Clulow J, Papadimitratos P, Anderson R, Hubaux J-P. Fast exclusion of errant devices from vehicular networks. In: 5th Annual IEEE

communications society conference on sensor, mesh and ad hoc communications and networks, 2008, SECON '08; June 2008b. p. 135–43.
Nadeem T, Dashtinezhad S, Liao C, Iftode L. Trafficview: traffic data dissemination using car-to-car communication. ACM SIGMOBILE Mobile Computing and Communications Review 2004;8.
National Highway Traffic Safety Administration, U.S.D.o.T. Vehicle safety communications project—final reports. Technical report; April 2006 〈http://www-nrd.nhtsa.dot.gov/〉.
Nowatkowski M, Owen H. Certificate revocation list distribution in VANETs using most pieces broadcast. In: Proceedings of the IEEE southeastCon 2010 (SoutheastCon); 2010. p. 238–41.
Papadimitratos P, Buttyan L, Hubaux J-P, Kargl F, Kung A, Raya M. Architecture for secure and private vehicular communications. In: 7th International conference on ITS telecommunications, 2007. (ITST '07); June 2007. p. 1–6.
Papadimitratos P, Fortelle AL, Evenssen K, Brignolo R, Cosenza S. Vehicular communication systems: enabling technologies, applications, and future outlook on intelligent transportation. IEEE Communications Magazine 2009; 47(11):84–95.
Papadimitratos PP, Mezzour G, Hubaux J-P. Certificate revocation list distribution in vehicular communication systems. In: VANET '08: Proceedings of the 5th ACM international workshop on VehiculAr Inter-NETworking. New York, NY, USA: ACM; 2008. p. 86–7.
Picconi F, Ravi N, Gruteser M, Iftode L. Probabilistic validation of aggregated data in vehicular ad-hoc networks. In: VANET '06: Proceedings of the 3rd international workshop on vehicular ad hoc networks. New York, NY, USA: ACM; 2006. 76–85.
Parno B, Perrig A. Challenges in securing vehicular networks. In: Proceedings of the workshop on hot topics in networks (HotNets-IV); November 2005.
Plobl K, Nowey T, Mletzko C. Towards a security architecture for vehicular ad hoc networks. In: International conference on availability, reliability and security; 2006. p. 374–81.
Rass S, Fuchs S, Schaffer M, Kyamakya K. How to protect privacy in floating car data systems. In: VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking. New York, NY, USA: ACM; 2008. p. 17–22.
Raya M, Aziz A, Hubaux J-P. Efficient secure aggregation in vanets. In: VANET '06: Proceedings of the 3rd international workshop on vehicular ad hoc networks. New York, NY, USA: ACM; 2006a. p. 67–75.
Raya M, Hubaux J-P. The security of vehicular ad hoc networks. In: SASN '05: Proceedings of the 3rd ACM workshop on security of ad hoc and sensor networks. New York, NY, USA: ACM; 2005a. p. 11–21.
Raya M, Hubaux J-P. The security of vehicular ad hoc networks. In: Proceedings of the 3rd ACM workshop on security of ad hoc and sensor networks, SASN '05. New York, NY, USA: ACM; 2005b. p. 11–21. 〈http://doi.acm.org/10.1145/1102219.1102223〉.
Raya M, Jungels D, Papadimitratos P, Aad I, Pierre Hubaux J. Certificate revocation in vehicular networks; 2006b. 〈http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.92.2291〉.
Raya M, Papadimitratos P, Aad I, Jungels D, Hubaux J-P. Eviction of misbehaving and faulty nodes in vehicular networks. IEEE Journal on Selected Areas in Communications 2007;25(8):1557–68.
Raya M, Papadimitratos P, Hubaux J-P. Securing vehicular communications. IEEE Wireless Communications 2006c;13(5):8–15.
Reichardt D, Miglietta M, Moretti L, Morsink P, Schulz W. Cartalk 2000: safe and comfortable driving based upon inter-vehicle-communication. In: Intelligent vehicle symposium, vol. 2. IEEE; 2002. p. 545–50.
Reid D. An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control 1979;24(6):843–54.
Sampigethaya K, Huang L, Li M, Poovendran R, Matsuura K, Sezaki K. Caravan: providing location privacy for vanet. In: Embedded security in cars (ESCAR); 2005.
Scheuermann B, Lochert C, Rybicki J, Mauve M. A fundamental scalability criterion for data aggregation in vanets. In: Proceedings of the 15th annual international conference on mobile computing and networking. MobiCom '09. New York, NY, USA: ACM; 2009a. 285–96. 〈http://doi.acm.org/10.1145/1614320.1614352〉.
Scheuermann B, Lochert C, Rybicki J, Mauve M. A fundamental scalability criterion for data aggregation in vanets. In: ACM MobiCom; September 20–25, 2009b.
Shamir A. Identity-based cryptosystems and signature schemes. In: Proceedings of the CRYPTO 84 on advances in cryptology. New York, NY, USA: Springer-Verlag; 1985. p. 47–53.
Studer A, Shi E, Bai F, Perrig A. Tacking together efficient authentication, revocation, and privacy in vanets. In: Sixth annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks, 2009. SECON '09; 2009. p. 1–9.
Sun Y, Lu R, Lin X, Shen X, Su J. A secure and efficient revocation scheme for anonymous vehicular communications. In: IEEE international conference on communications (ICC), 2010; May 2010. p. 1–6.
Uzcategui RA, Acosta-Marum G. Wave: a tutorial. IEEE Communications Magazine 2009;47(5):126–33.
Wiedersheim B, Ma Z, Kargl F, Papadimitratos P. Privacy in inter-vehicular networks: why simple pseudonym change is not enough. In: Seventh international conference on wireless on-demand network systems and services (WONS); 2010. p. 176–83.
Xiao B, Yu B, Gao C. Detection and localization of sybil nodes in vanets. In: DIWANS '06: Proceedings of the 2006 workshop on dependability issues in wireless ad hoc networks and sensor networks. New York, NY, USA: ACM; 2006. p. 1–8.
Yu B, Gong J, Xu C-Z. Catch-up: a data aggregation scheme for vanets. In: VANET '08: Proceedings of the 5th ACM international workshop on VehiculAr Inter-NETworking. New York, NY, USA: ACM; 2008. p. 49–57.

Ad Hoc Networks 10 (2012) 1115–1133

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

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

Chains of Trust in vehicular networks: A secure Points of Interest dissemination strategy

David Antolino Rivas *, Manel Guerrero-Zapata

Department of Computer Architecture, Polytechnic University of Catalonia, Barcelona 08034, Spain

ARTICLE INFO

Article history:
Received 24 May 2011
Received in revised form 12 January 2012
Accepted 17 February 2012
Available online 3 March 2012

Keywords:
Security
VANETs
Chains
Trust
POI
Reputation

ABSTRACT

This article describes a scheme which to the best of our knowledge is the first one to use user signatures to share information about *Points of Interest* in *Vehicular Ad hoc Networks*. In this scheme, users rate restaurants, hotels, etc. and sign those rates with their private key. Then, they broadcast that information and other vehicles store it for future use. When another user needs a *Point of Interest* recommendation he queries the system for the other users stored reviews and after he visits that *Points of Interest* for himself, he evaluates it and his level of trust in the reviewers with rates similar to his own increases. In the end, a user will be able to request to his vehicle information on a certain *Point of Interest* category and it will respond with the recommendations made by other users, prioritizing the ones in the user's *Web of Trust*. *poiSim* is the tool designed to simulate this scheme. It processes a 24 h mobility trace produced by a Multi-Agent Traffic Simulator, which realistically simulates public and private traffic over regional maps of Switzerland. The result is a *Chains of Trust* simulation with over 260,000 nodes, which shows that the proposed scheme performs satisfactorily in a realistic scenario.

1. Introduction

With the massive deployment of wireless technologies on motorized vehicles, the automotive industry has opened a wide range of possibilities for drivers and passengers alike: theoretically, anything from finding out the road conditions ahead to watching a movie through streaming is possible. So different requirements will lead to the deployment of different kinds of applications over the network. In [1,2] applications are classified based on the service they provide:

1. Safety related applications:
   (a) *Traffic information messages*: used to disseminate traffic conditions over an area; they affect public safety only indirectly (they are not time-critical).
   (b) *General safety-related messages*: used by public safety applications such as cooperative driving and collision avoidance (in order to prevent traffic accidents time is certainly an issue; at least they should satisfy an upper bound delay in delivering the information).
   (c) *Liability-related messages*: they are only exchanged in liability-related situations such as accidents. The senders' identities should be kept hidden from the other users in the network and only revealed to the law authorities (time is not an issue).
2. Other applications (some examples):
   (a) *Toll applications*: electronic toll collection systems like *AutoPASS* in Norway allow drivers to continue driving without having to stop at tolls.
   (b) *TV and other multimedia content*: used to provide users with entertainment and information (movies, newspapers, etc.).
   (c) *Advertisements*: businesses along the road (such as gas-stations and restaurants) could advertise themselves to drivers before they reached their location, giving them enough time to compare different offers.

* Corresponding author. Tel.: +34 93 405 40 44.
E-mail address: antolino@ac.upc.edu (D. Antolino Rivas).

1570-8705/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2012.02.011

Messages from safety applications should ensure their integrity and their non-repudiation albeit maintaining at the same time the user's privacy. Other applications may also need to encrypt their traffic to transmit sensitive information, whereas that may be unnecessary for applications in the first group.

Architecture wise, applications can also be divided in two groups. On one hand, there are *Zero-infrastructure applications* where the only hardware requirement is the installation of *On Board Units* (OBUs) in the vehicles. OBUs provide the vehicles with sensing, processing and wireless communication capabilities for *Vehicle to Vehicle* (V2V) communications, like in [3]. On the other hand, there are applications that also need *Road Side Units* (RSUs) to provide a *Vehicle to Infrastructure* (V2I) link, generally because they use *Public Key Infrastructure* (PKI) and they require access to a *Certification Authority* (CA) outside the network or to an Internet Service Provider [4–11]. However, with the recent development of cellular technologies like GPRS and UMTS the V2I link could by provided by the OBU itself, minimizing the dependency on road side infrastructure.

This article presents *Chains of Trust*, a secured *Zero-infrastructure* dissemination scheme based on a reputation system, focused on the distribution of *Points of Interest* (POIs) information.

Briefly summarized, every user or vehicle creates its own pair of public and private keys (of length L), and is responsible for its private key securing; the protocol does not require a CA. When users visit POIs they evaluate them and input their reviews into the system. The private key is used to sign those POI reviews, whereas the public key is attached to the transmitted information so that the rest of the network can verify the signatures.

The remainder of this work is organized as follows. In Section 2, several solutions to distribute information in VANETs are presented. Section 3 describes *Chains of Trust* in further detail, followed by a description of the simulation tool *poiSim* in Section 4 and the experimentation results in Section 5. Finally, the article closes with the conclusions drawn from those results.

2. Related work

This article introduces an information dissemination technique, which to the best of our knowledge is the first one to build a reputation scheme using user signatures to distribute *Points of Interest* (POIs) information in a *Vehicular Network* (VANET).

Nevertheless, there are other works that consider the distribution of content in VANETs. For instance, in [12] the authors describe a protocol for the distribution of advertisements. They propose a virtual cash scheme where the following actors are involved:

- *Certification Authority* (CA): every vehicle is loaded with a pair of keys (public and private) issued by a CA and with the CA's public key.
- *Vehicular Authority*: entity that approves every advertisement to be loaded in an *Ad Distribution Point*.
- *Ad Distribution Point*: broadcasts advertisements to the vehicles passing by.
- *Virtual Cashiers*: users are rewarded with virtual cash for forwarding advertisements. They sign each other receipts to prove the message forwarding. Later on, that cash can be exchanged for other services at the *Cashiers*.
- *Road Side Units* (RSU): provide a link to the CA for keys revocation purposes.

In [13] the authors present Roadcast, a popularity aware P2P content sharing scheme. Their technique relies on the idea that by ensuring that popular data is widely shared with other vehicles the overall query delay can be improved. If users request popular data, which is densely disseminated in the network, their queries can be answered in much shorter time than a request for rare data, because the chance of meeting another vehicle with that particular piece of information is much higher. In the opportunistic and unreliable VANET, the authors expect users to be more willing to receive data which approximately matches their request with a short delay than waiting for a longer time to receive exactly what they requested. Thus the need to forward the popular information with higher priority.

Data aggregation is another aspect of *Chains of Trust* that should be taken into consideration, since the number of POI and user reviews is so large. In [8], the authors detail several signature techniques to achieve data aggregation:

1. *Concatenated signatures*: each user's signature is appended (together with his certificate) to the original message. The greatest benefit, in contrast to other schemes, is that an invalid signature does not affect the whole message.
2. *Onion signatures*: every user signs the last user's signature and appends his certificate to the message. This technique is very good in terms of data aggregation, since not only the data, but also the signatures are aggregated. However, a single invalid signature could corrupt the whole message.
3. *Hybrid signatures*: several concatenated onion signatures, each of a given depth. This solution looks for a compromise between the previous two, both on their advantages and drawbacks.

Onion and hybrid signatures achieve better aggregation, which means that users can transmit more information in their messages. However, whenever the number of reviews in a message reaches its maximum size the chain of

Table 1
Percentage of received broadcasts for every simulated scenario.

Percentage of received broadcasts					
Number of packets/ period	60	120	180	240	300
100	95.97	97.99	98.62	98.96	99.15
200	91.57	95.91	97.27	97.93	98.38
300	86.86	93.72	95.82	96.87	97.54
400	82.16	91.50	94.37	95.78	96.64
500	77.23	89.28	93.01	94.80	95.85
600	71.93	87.00	91.57	93.73	94.98
700	66.30	84.58	90.03	92.57	94.06
800	60.54	82.19	88.51	91.48	93.22

signatures becomes stale, since there is no way to remove old signatures to leave room for the new ones (and its certificates). That is the reason why *Chains of Trust* uses concatenated signatures.

As important as content distribution and data aggregation are security and privacy. The following articles rely on the use of a *Certification Authority* (CA) and focus their efforts on minimizing the length and efficiently distributing the *Certificate Revocation Lists* (CRLs).

In [14] the authors introduce a pseudonymous authentication scheme whose CRL size is linear in terms of the number of revoked vehicles and unrelated to the size of the vehicle pseudonymous certificate set. The network architecture is composed by a *Trusted Authority* (TA), RSUs and vehicles or OBUs. The TA issues a certificate $Cert_{TA,\ R_x}$ for a certain RSU R_x, and a series of pseudonymous certificates for a vehicle V_i to be installed during periodic vehicle inspections. It should be noted that the identities in the pseudonyms certificates are derived from two random seeds using a one-way hash function. The TA divides the maximum time between vehicle inspections into time windows. For every window, the TA chooses a random secret key to sign the vehicle's pseudonymous certificates, so that in every window the vehicle has to request R_x to re-sign the pseudonymous certificate for that window. In this scenario, a RSU can be revoked by including its only certificate in a CRL. To revoke a vehicle it would suffice for the TA to release the random seeds from which V_i's pseudonymous identities are computed, so that the RSUs do not issue the re-signature key to V_i in following windows. At the same time the valid pseudonymous certificate of V_i should be revoked.

Similarly, authors in [15] try to achieve the same small CRL size with a different technique. They propose a scheme based on *Temporary Anonymous Certified Keys* (TACKs), used to authenticate messages sent by the vehicles. There are four main entities:

- M: managing authority that acts as the root of trust.
- R: set of valid *Regional Authorities* (RAs). RAs act as intermediary authorities and can grant vehicles temporary region-specific certificates. M issues certificates to RAs and certifies them to be valid intermediary authorities.
- V: set of valid vehicles or *On Board Units* (OBUs). Any vehicle with a valid certificate from M or a region-specific short-lived certificate from R (while in the proper region) is considered part of V.
- $\neq gV$: set of expired or revoked vehicles.

The main idea is to apply group signatures considering a group which comprises all of the above described entities. M is defined as the group manager. It initializes the group signature scheme to generate a group public key *gpk* and a group master key *gmk*. It publishes *gpk* and retains *gmk* for itself. Each valid OBU has a group user key guk_i, issued by M, which is installed during annual vehicle inspections. It should be noted that M maintains a history of all key/OBU pairs it has issued, so that it can later trace misbehaving vehicles. When a vehicle enters a new region it needs to update its TACK following these steps:

1. Randomly select new short-lived public and private keys from the key space (K_S^+, K_S^-).
2. Use the group user key guk_i to sign K_S^+ and send it to the RA.
3. RA verifies that the user is not in the RL. If it is not, the RA signs a certificate for the OBU's TACK public key K_S^+ using the RA's secret signing key K_{RA}^{-1}.
4. RA waits for δ seconds to queue up all certificate requests for that region and broadcasts the certificates.

Whenever a user wants to send a message it signs it with its TACK private key K_S^{-1} and periodically broadcasts the RA signed certificate of its TACK public key K_S^{-1}. Whenever a user misbehaves, to determine which OBU generated a signature ψ the group manager tests ψ against the group user keys of OBUs in V. Once M identifies V_i it is added to the RL and distributed to the RAs.

In [16], the authors define *Most Pieces Broadcast* (MPB) technique to distribute CRLs. The first step is to break the large CRL file down into small pieces, taking into consideration the coding rate (rate of pieces generated from a file) and the code overhead (number of pieces needed to recover the original file). MPB ensures that only the node with the largest number of pieces broadcasts in a certain area to maximize the use of the wireless channel. It should be noted that RSUs will always be selected as the node with most pieces. The authors show that MPB is more effective than letting all OBUs broadcast their CRL pieces without control, which results in a broadcast storm of unneeded CRL pieces that slows down the CRL distribution.

In [1,6,17–21] the authors deal with privacy issues and tracking vulnerabilities due to use of wireless communications. They discuss several techniques to improve security, such as silent periods, mix zones or the use of pseudonyms. In *Chains of Trust* that is unnecessary since there is no road side infrastructure collecting all the transmitted messages. If an attacker wishes to track a certain user, he needs to be constantly in range, thus physically following his victim or have enough resources to deploy his own road side infrastructure.

On the whole, *Chains of Trust* distinguishes itself from the other solutions presented in this section by using broadcast in a completely ad hoc network as a mean of information dissemination, by not using a CA and therefore not having to deal with the distribution of CRLs and by performing data aggregation through the use of concatenated signatures.

3. Chains of Trust: a Points of Interest dissemination strategy

3.1. Introduction

The vast majority of applications in VANETs use *Public Key Infrastructure* (PKI), because it provides confidentiality, integrity, authentication and non-repudiation and because it is a well known and reliable system. However, VANETs have their own peculiarities and if PKI does not adapt to them security issues arise. For instance, a vehicle continuously sending messages signed with the driver's private key becomes traceable, and thus the user's privacy is

violated. As seen in the previous section, another major issue comes from managing CRLs. CAs include revoked certificates in CRLs, which have to be distributed across the network. This poses a difficult challenge, particularly in the early stages of a VANET deployment, if the vehicles do not have permanent (or frequent enough) access to a CA. Furthermore, with millions of users in the system the potential size of the CRLs is huge.

This article presents a new technique for secure dissemination of *Points of Interest* (POIs) information over VANETs, which does not require a CA. Our scheme relies on a reputation system or *Web of Trust* based on human driving patterns where one users will trust another if they both give POIs similar reviews.

The main objective is to take advantage of those patterns and build a system, whose knowledge is distributed among the users' vehicles, which they can query for POIs information. Those POIs could be anything from road conditions to museums or restaurants.

Over the next sections the different elements of the system are detailed, some of which being: what can be considered a POI, what information should the vehicles transmit and store and how does the reputation scheme work. Once the gearing of the system has been precisely defined, the simulation tool used to prove the effectiveness of the technique (*poiSim*) and its simulation results will be presented.

3.2. Scheme overview

In the reputation system, every vehicle needs to store information about other vehicles and POIs (whether received from other users or reviewed by himself). Every node in the network shall store:

- POI chains: they are a series of reviews of the same POI from different users. As depicted in Fig. 1 POI chains can be divided in:
  - Unverified POI chains: they contain POI reviews that the user has received from other users but which he has not yet been able to verify (by visiting and rating the POI himself), e.g., a traffic jam alert or the review of a new restaurant. Every unverified chain is rated based on the level of trust the user has in the known reviewers in the chain. When a user queries his vehicle, the POIs information is displayed ordered by that rate as defined in Section 3.3.1.
  - Verified POI chains: once the user has a chance to check if there really is a traffic jam or how good that restaurant is, he evaluates the reviewers in the unverified chain and updates his level of trust in them depending on how truthful they were and marks the chain as verified. Verified chains are an essential part of the exchange of information between users, as will be explained in Section 3.5.
- Trust levels in other users (per category): every node needs to remember how much he trusts other users based on the verification of previous reviews. Besides, nodes not only share information about POIs, but also information about other nodes. For those recommended nodes several other properties will have to be stored, as will be detailed in Section 3.4.
- Information about the latest messages from every user, both about POIs and nodes, should be stored for misbehavior detection, e.g., if the user is in a misbehaving strike. Further details will be given in Section 3.7.

Since there is no CA, every user or vehicle will create its own pair of public and private keys (of length L) and will be responsible for its securing. Notice that K_{pub} is the user identifier, therefore L should be long enough to ensure

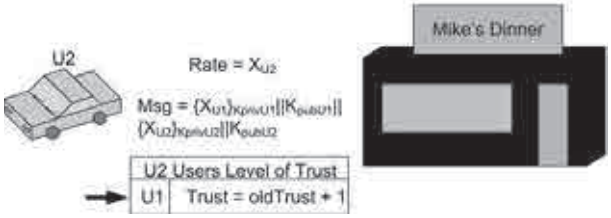
(a) Unverified POI chains organization

(b) Verified POI chains organization

Fig. 1. POI chains organization.

(a) $\boldsymbol{U}_1$ reviews ***Mike's Dinner*** and broadcasts the message.

(b) $\boldsymbol{U}_2$ queries his vehicle for a restaurant.



(c) $\boldsymbol{U}_2$ follows $\boldsymbol{U}_1$ recommendation, adds his own opinion to the chain and updates his level of trust in $\boldsymbol{U}_1$ according to how similar both reviews were.

Fig. 2. General behavior of the *Chains of Trust* protocol.

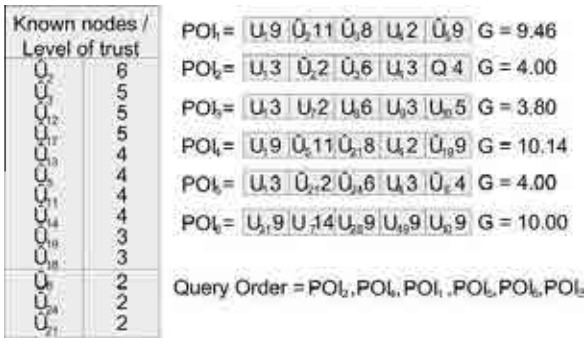


Fig. 3. User *Q* chains grading process.

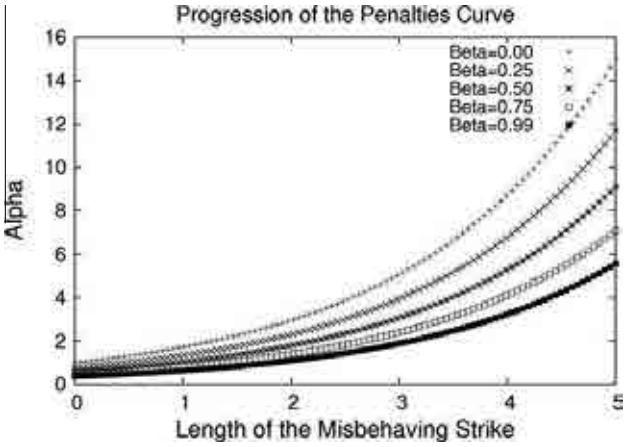


Fig. 5. Progression of the function $f(x) = \left(e^{\frac{1}{5}\ln(15)-\beta}\right)^{x}$.

the statistical uniqueness of identities. That is why the scheme uses RSA with 1024 bits long keys. The private key will be used to sign information about POIs and about the levels of trust that a particular vehicle has in the others, while the public key will be attached to that information so that the rest of the network can verify the signatures correctness. For instance, consider the scenario depicted in Fig. 2. Imagine that a user U_1 goes to a restaurant *A* and he likes it. U_1 will broadcast a message to the other users in the network saying that restaurant *A* deserves a certain rate χ, signed with his K_{priv} and attaching his K_{pub}. All the other nodes that successfully receive the message store the unverified chain for future reference. When another user U_2 queries his own vehicle for a place to have lunch the vehicle returns a list of places recommended by other users (among which is U_1's recommendation). If U_2 decides to go to *A* he will afterwards input his review into the system and if he liked it as much as U_1 his level of trust in U_1 will increase, or decrease otherwise. Regardless of how much he coincided with U_1's opinion, U_2 will append his signed review to the original, together with his K_{pub}, and broadcast the message. In this way, every time a user follows and verifies a recommendation he can update his level of trust in *n* other nodes (where *n* is the length of the chain of signatures), thus increasing the speed at which the reputation system develops.

In order to foster the development of *Chains of Trust* at an early stage vehicles could be pre-loaded with a set of POIs at the same time the application is being installed. In this way, users could benefit from the application since the very beginning, even compensating for a low initial adoption rate. In addition, those pre-loaded POIs could help users moving through a new area where they do not know anybody else, as will be described in Section 3.6.

As mentioned above, this technique does not require a *CA*, or any road side infrastructure for that matter, since the network is completely ad hoc and there is no certificate revocation process to manage. Every user generates his own pair of keys (the public key being his identity) and begins to play a part in the network by signing information. In the beginning, his identity is unknown to the rest of the users, therefore, he has to gain the others trust by telling the truth. That is how the scheme protects itself against misbehaving. If an attacker misbehaves from the start he will not be able to inflict any real damage since all the nodes join the system with the lowest level of trust, and his reviews will be mostly unnoticed. If he tries to gain some credit and then misbehaves the *Rewards and Penalties* system will recognize a misbehaving strike and punish it. Although nothing prevents the attacker from creating a new identity he will not gain anything from it, since any new identity has no credit on the network. It should be noted that the level of trust of one user in another will decrease if the second either lies to him by misbehaving or if he rates a POI significantly different than the first would. Therefore, the terms lies and disagreements shall be used indistinctively throughout the article. More details on misbehavior can be found in Section 3.8.

As far as the application's platform is concerned, we would like to elaborate on why *Chains of Trust* is specifically a VANETs application and not appropriate for other mobile platforms, e.g., smartphones, PDAs, etc. For starters, vehicles provide enough energy for the required periodic exchange information and for a fast enough processor to handle RSA encryption and decryption operations.

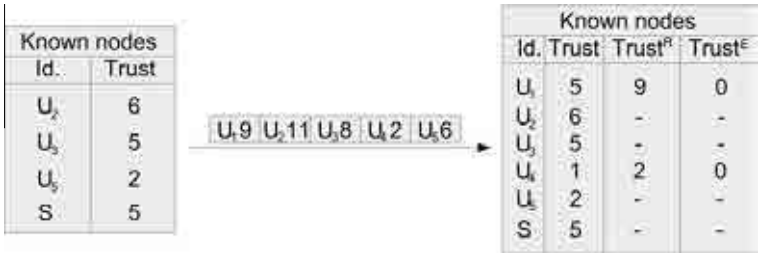


Fig. 4. *R*'s known nodes table before and after processing a Recognition Exchange message.

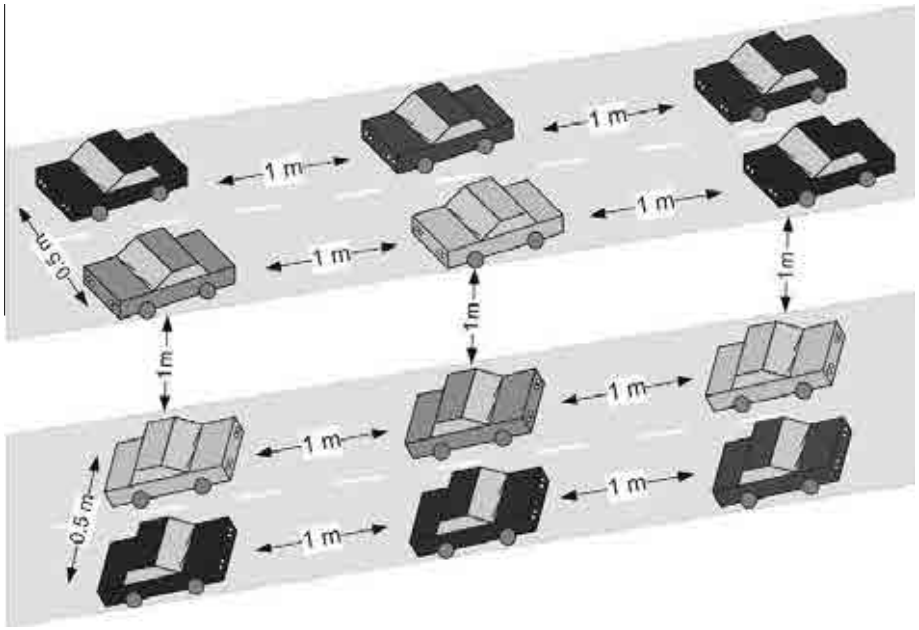


Fig. 6. Vehicle layout for the 400 nodes simulated in Ns-3.

Secondly, a larger amount of memory could be installed in order to store more data about the vehicles in the user's *Web of Trust*. Finally, vehicles allow for the installation of antennas with better gain, improving message reception and giving us the possibility to extend the transmission range. In addition, we believe that an application specifically conceived for smartphones would require a completely different solution. With 3g network access, users could connect to a remote server only when they needed to query for a POI category or submit their own POI reviews, which would mean that this remote server should have enough resources to store all the users' information. In addition, a CA would need to issue and distribute certificates to allow users to securely authenticate with the server. This a completely different scenario from our ad hoc network proposal, which requires no infrastructure (remote server or CA) and where the system knowledge is distributed among its users.

3.3. POI categories and records

Several POI categories shall be considered, and a different level of trust for each category for each user shall be kept by each vehicle, i.e., a user may be a good hotel reviewer and a terrible restaurant critic. The following is an example list of what may be considered a POI category:

- Traffic conditions.
- Gas stations.
- Grocery stores.
- Restaurants.
- Hotels.
- Bars.
- Museums.
- General entertainment.

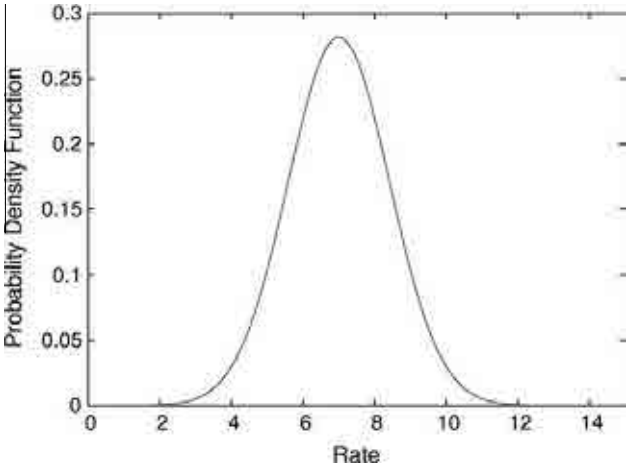


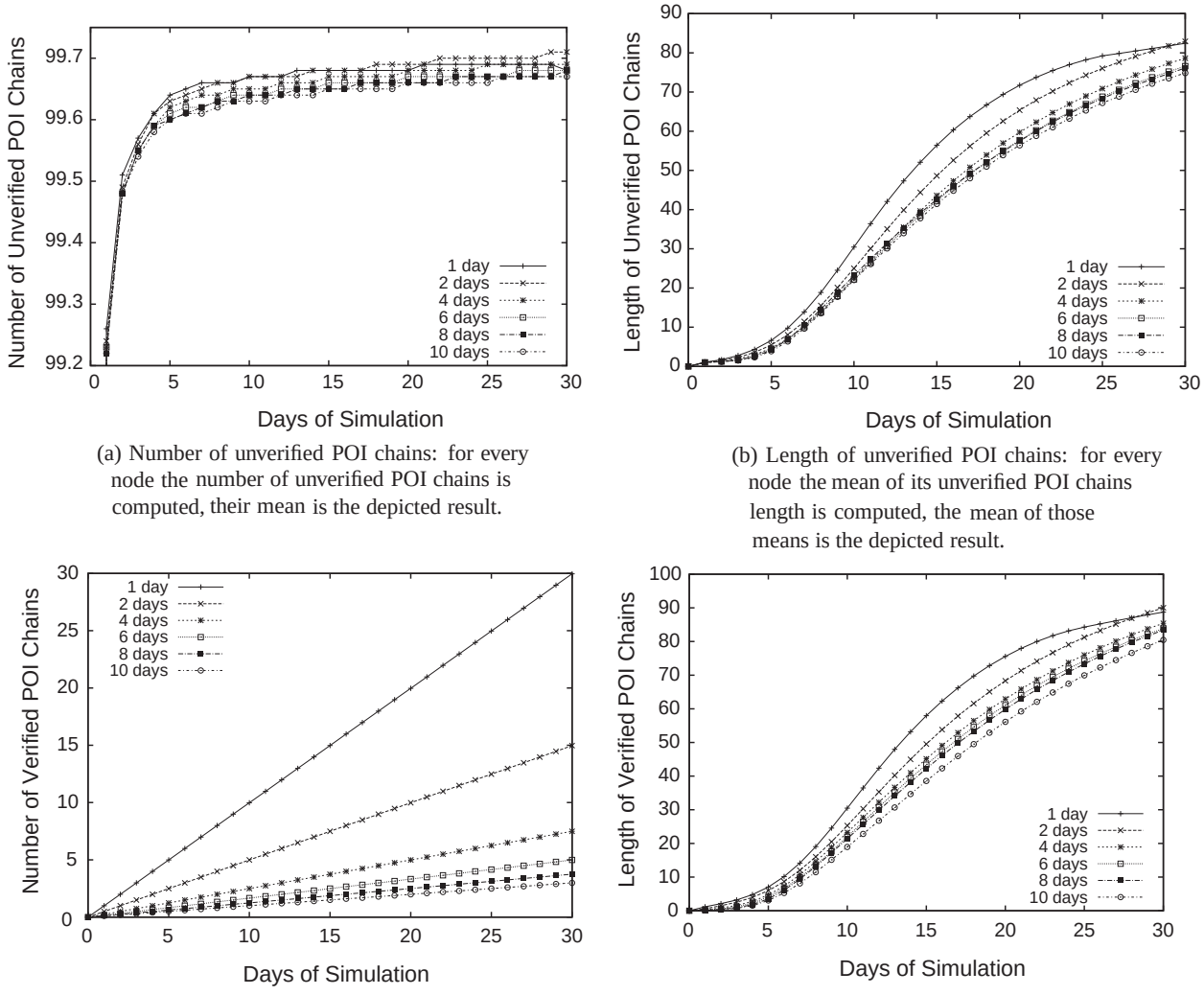
Fig. 7. User's rate distribution for the real rate $\mu = 7$ and $\sigma^2 = 2$.

For each category a validity period is defined, e.g., a hotel review may be valid for months whereas a traffic jam alert may expire within hours or even minutes. That validity period is necessary to prevent unfair punishments. For instance, if a user identifies a traffic jam and sends a message alerting the network and several hours later another vehicle passes by and sees no trace of it he should not decrease his level of trust in all the users who signed the alert message.

Before POIs can be reviewed we first need to give them a unique identifier consisting of common knowledge information:

$$Id = \{Category||POI\_Name||Postal\_Address||GPSCoords\} \quad (1)$$

The Postal Address and the GPS Coordinates fields complement each other, since it is difficult to give the Postal Address of a traffic jam or the GPS coordinates of a restau-



(a) Number of unverified POI chains: for every node the number of unverified POI chains is computed, their mean is the depicted result.

(b) Length of unverified POI chains: for every node the mean of its unverified POI chains length is computed, the mean of those means is the depicted result.

(c) Number of verified POI chains: for every node the number of verified POI chains is computed, their mean is the depicted result.

(d) Length of verified POI chains: for every node the mean of its verified POI chains length is computed, the mean of those means is the depicted result.

Fig. 8. Evolution of the length and number of unverified and verified chains.

rant (unless you position your vehicle right at the door). It should be noted that the GPS coordinates will admit a certain margin of error due to the devices positioning error.

Whenever a user wants to review a POI, he will assign a rate to it and assemble a record R with the following information:

$$R = \{Id \| Rate \| Timestamp\} \tag{2}$$

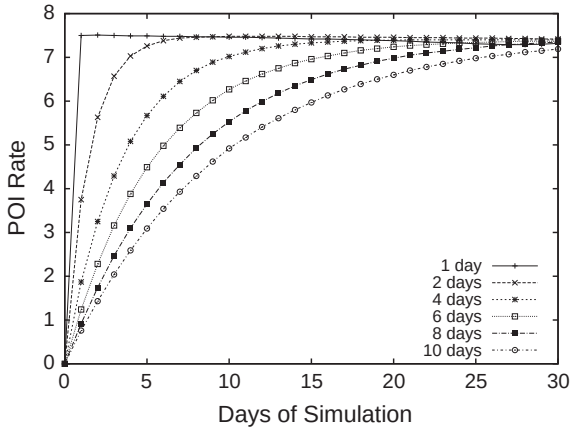
Each record has a timestamp so that users are able to keep track of the validity period per category. In addition, it could also be used to remove old entries from the trusted nodes table.

Once the record has been prepared, the sender needs to sign it (by encrypting the record's hash with his private key) and attach his public key to it. At some point in the future the vehicle will broadcast M.

$$M = \{R_1 \| \{H(R_1)\}_{K_{priv_A}} \| K_{pub_A}\} \tag{3}$$

Afterwards, when a vehicle receives a message it stores it for future use. When a user queries his vehicle for a recommendation on a POI category in a certain area the system answers with a list of received POIs, the ordering of which follows the criteria defined in Section 3.3.1. If the user follows the recommendation he will be able to write another review about the recommended POI. The idea is to keep the previous reviews and attach the latest to the group, thus forming a chain of signatures that grows until a parameter n. By keeping a chain of size n every time that a user follows a recommendation he will be able to update his level of trust in n other users. It should be noted that the new added records are a slightly modified version of the first because they contain the hash of the original POI Id, instead of the complete identifier.

$$R' = \{H(Id) \| Rate \| Timestamp\} \tag{4}$$



(e) Rate in the verified POI chains: the mean of the rates users assign to POIs.

Mean of the deviation σ																					
Simulation day	1			5			10			15			20			25			30		
Rating freq. (review/days)	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10
Num. Unver. POI chains	59.86	63.26	63.69	33.09	36.27	37.08	31.01	33.70	34.61	30.06	32.30	33.24	29.76	31.34	32.30	29.62	30.70	31.64	29.60	29.99	31.12
Length Unver. POI chains	0.30	0.17	0.14	7.21	8.55	7.94	30.84	29.78	30.50	40.39	40.25	40.42	41.11	43.32	44.13	40.74	43.08	44.44	40.78	41.76	43.10
Num. Ver. POI chains	0.00	0.36	0.30	0.14	0.83	0.66	0.30	1.17	0.95	0.47	1.44	1.16	0.62	1.67	1.34	0.79	1.87	1.50	0.94	2.05	1.64
Length Ver. POI chains	0.00	0.00	0.00	7.81	6.21	3.97	32.05	27.09	20.65	44.28	42.12	34.37	46.98	49.37	43.25	47.57	51.74	47.99	48.01	51.87	49.93
Rate Ver. POI chains	0.00	0.00	0.00	4.35	1.91	1.24	4.35	3.11	2.23	4.34	3.73	2.91	4.34	4.03	3.38	4.34	4.19	3.69	4.32	4.26	3.91

(f) Mean of the deviation table for (a), (b), (c), (d) and (e).

Fig. 8 (*continued*)

The Id field (or its hash to be more precise) needs to be included in each of the added records to prevent a security vulnerability. Imagine that the messages were shortened by removing the *Id* to decrease the transmission time and to save storage space in the vehicles. Then, only the first record of the chain would be bound to the POI. As a result, it would suffice for a misbehaving node to replace that first record with another POI *Id* and broadcast that message over the network to ruin the reputation of the other signers. A good alternative would be to use Onion Signatures (as described in [8]) to preserve the message integrity every time a new record is added. However, Onion Signatures do not take into account that a message cannot grow indefinitely and at some point new records will replace old ones which deems this scheme unfeasible since in order to preserve its integrity not a single bit of information can be discarded.

A message containing a chain of length 2 is of the form:

$$M = \{R_1 \| \{H(R_1)\}_{K_{priv_A}} \| \{R_2'\}_{K_{priv_B}} \| K_{pub_A} \| K_{pub_B}\} \tag{5}$$

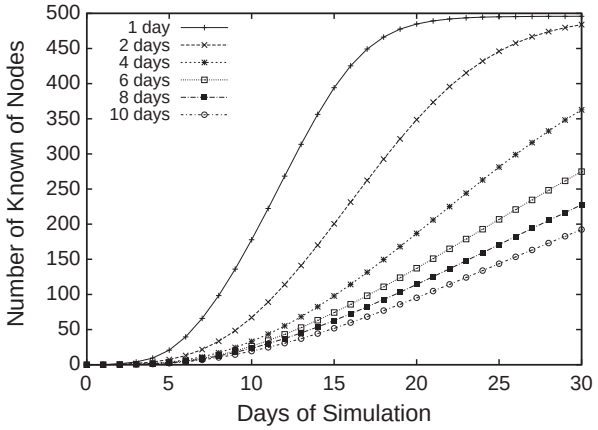
It should be noted that the added records are not hashed and then signed, but directly encrypted with the user's private key. Since R' includes the hash of the POI identifier, it already is a short message. Therefore, the use of digital signatures on it would make the hash function redundant.

3.3.1. POI chains grading

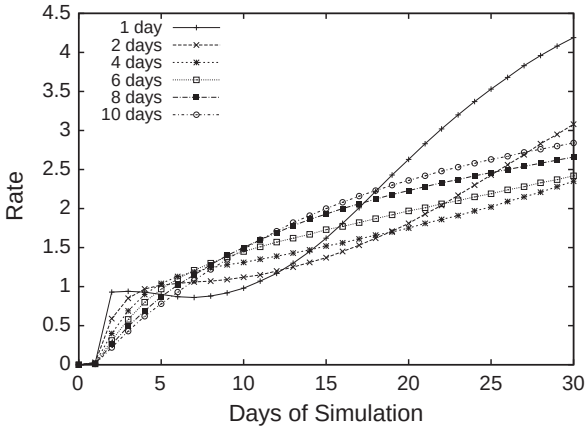
When a user Q queries his vehicle for a certain POI in its vicinity, the system needs to display all received recommendations following a certain order. In the case of verified chains that order is determined by the rate the user assigned to a POI the last time he was there. In the case of unverified chains the order is defined by the trusted (and in some cases by the most trusted) nodes in the chain. Let us define n as the number of reviews in a certain chain $POI_1, U_1, \ldots, U_n$ as the users whose POI reviews are in the chain and $\widehat{U}_1, \ldots, \widehat{U}_n$ as the subset of those nodes known by the user Q, χ_{POI_1,U_1} as the rate that U_1 gave to POI_1 and $\lambda_{\widehat{U}_i}$ as the level of trust that Q has on U_i as a POI reviewer. Then the chain grade G is defined by:

$$G = \sum_{i=0}^{n} \left(\chi_{POI_1,\ \widehat{U}_i} \cdot \frac{\lambda_{\widehat{U}_i}}{\sum_{j=0}^{n} \lambda_{\widehat{U}_j}} \right) \tag{6}$$

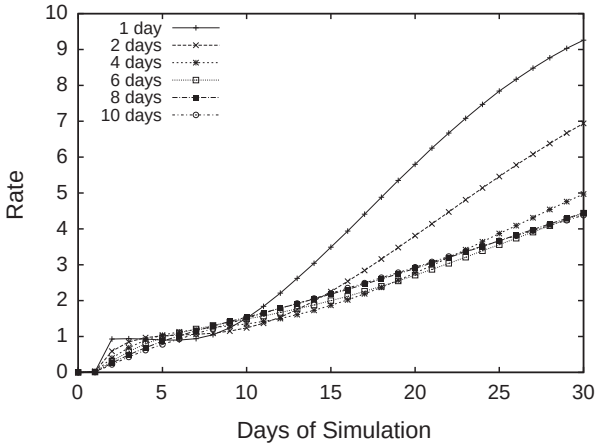
It should be noted that the rates assigned by unknown nodes are ignored as long as there is a known reviewer in the chain. Otherwise, the chain's rate is the arithmetic mean of the POI rates assigned by the unknown reviewers. Similarly, the reviews of the less trusted known nodes are ignored when there is a known node that belongs to the group of Q's k most trusted nodes. In order to prevent misbehavior only Q's most trusted users, i.e., the ones on the first k positions of the list, are considered for grading the chain. Otherwise, an attacker could create multiple low trusted identities and reduce the weight of legitimate reviews in Eq. (6) to obtain his desired result. By prioritizing the opinions of a small group of reviewers over the rest an attacker will first need to gain enough trust to belong into that group and once he starts misbehaving he will rapidly lose his influence, as described in Section 3.7.



(a) Number of known nodes: mean of the number of known nodes by every node.



(b) Rate or level of trust of the known nodes: mean of the rates users assign to other users as POI reviewers.



(c) Rate or level of trust of the 25% most trusted nodes: mean of the rates users assign to other users as POI reviewers (only for the 25% highest rated nodes).

Mean of the deviation σ																					
Simulation day	1			5			10			15			20			25			30		
Rating freq. (review/days)	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10	1/1	1/6	1/10
Num. Nodes	0.17	0.00	0.00	26.91	9.43	8.11	127.91	43.84	37.66	119.57	84.33	68.95	52.82	118.10	100.82	34.07	140.27	126.49	32.49	148.91	142.56
Nodes Rate	0.00	0.00	0.00	0.87	1.53	1.03	1.42	1.53	1.68	2.17	1.92	2.17	2.99	2.32	2.55	3.76	2.71	2.85	4.31	3.03	3.10
Nodes First 25% Rate	0.00	0.00	0.00	0.87	1.04	1.03	1.48	1.50	1.65	2.08	1.79	2.02	2.52	2.03	2.23	2.80	2.25	2.35	2.97	2.41	2.43

(d) Mean of the deviation table for (a), (b) and (c).

Fig. 9. Number of known nodes and their levels of trust progress.

If k is too small some good and trustable reviewers' opinion will never reach the top of the list, and therefore their opinion will not count as much as it should (according to their good behavior). However, if k is too large an attacker could easily gain access to the top k reviewers group and start misbehaving. The idea behind Algorithm 1 is to start with a low value and build-up. If the top k reviewers as a group gain more trust as the user reviews POIs the group can be expanded, which means more reliable information, and the user can prioritize their opinions over the rest of his trusted nodes. Otherwise, if one of the top k reviewers misbehaves then his own reputation will suffer, as described in Section 3.7, and k will decrease to expell the misbehaving node and minimize the impact of his future reviews.

Algorithm 1. How k is computed.

```
//initial value
k := 1;
```

(continued on next page)

```
//every time a POI is reviewed by a user
for every POIReview
  //compute the mean of the level of trust of the k
  most trusted users
  previousMean := computeMean(trustedNodesList, k)
  //process the POI review, update level of trust in
  other users or create a new chain if this is a
  //POI new
  processPOIReview(POIReview,trustedNodesList, k)
  //compute the mean of the level of trust of the new k
  most trusted users
  currentMean := computeMean(trustedNodesList, k)
  if currentMean > previousMean
    //the user's trust in the reviewers of the last POI
   has increased, therefore, his list of k most
    //trusted users can expand
    k := k + 1
  else if currentMean < previousMean
    //one of the reviewers in the k-top has disagreed
   with the user, k needs to be decreased to
    //prevent misbehavior
    k := k − 1
  end if
end for
```

As a result, when a user queries his vehicle, the system replies with a series of recommendations starting with verified chains, followed by unverified chains with reviews by its k most trusted reviewers, followed by unverified chains with the rest of trusted reviewers and closing with unverified chains with unknown reviewers. The chain's rate establishes its position within its category. For example, Fig. 3 depicts the grading process of several chains by user Q and the order in which they are presented to the user: POI_2 (verified chain), POI_4, POI_1 (unverified chains with the most trusted known reviewers), POI_5 (unverified chain with the rest of known reviewers) and POI_6, POI_3 (unverified chains without known reviewers).

3.4. Nodes and records

The use of user chains has to be carefully crafted in order to avoid abuse and misbehavior. Users in the network play two different roles: POI reviewers and other users reviewers. As POI reviewers, every vehicle has to store his level of trust in the other known users. As node reviewers, every vehicle needs to keep track of the nodes every other node recommends to him and their levels of trust as POI reviewer, because they impact on the level of trust the recommender deserves in that role. If a recommended node misbehaves (as POI reviewer) its recommender's reputation (as recommender) will suffer, or improve otherwise.

3.5. The information exchange

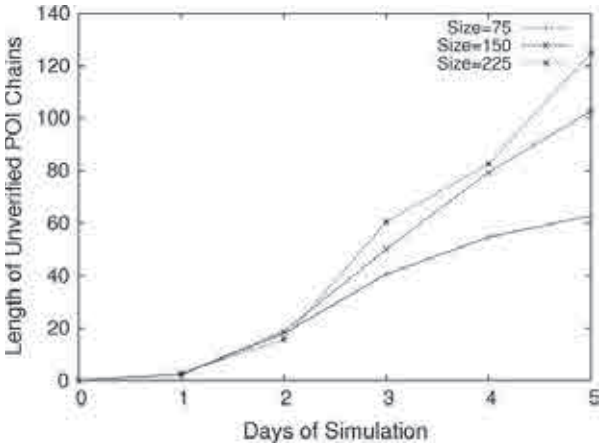
The application is designed to disseminate information about POIs among the vehicles in the network, thus the need for that information to flow from one vehicle to another. On one side there are POI chains (both verified and unverified) which represent the new information that comes into the system in the form of reviews of new POIs plus the re-evaluation of the already known. On the other, there are user chains, which are lists of known nodes and their level of trust. Basically, once two nodes know each other, besides exchanging information about POIs, they can exchange information about other users, thus increasing the speed with which the *Web of Trust* develops. The ideal way to exchange information would be for a user to issue a request for information on a certain category and its surrounding vehicles to answer it. However, it is not unusual that after having spent some time in a platoon formation a vehicle is alone or only has a few trusted vehicles in its vicinity at the time of sending the request. That is the reason why POI information should be exchanged periodically as well, and when the user needs a recommendation his vehicle still requests it to the nearby vehicles to complete what has already been gathered. As a result, the system provides the user with a satisfactory number of choices regardless of the trusted number of vehicles he has nearby when the request is sent.

Some would identify this periodic exchange of information as a tracking vulnerability. However, provided that the period between message exchanges is long enough (as explained in [1,20–22]), if an attacker plans to track a user's movements he is going to need to physically follow him, since there is no road side infrastructure to collect the messages he is going to need to be in range. Further details on the period value are given in Section 3.9.

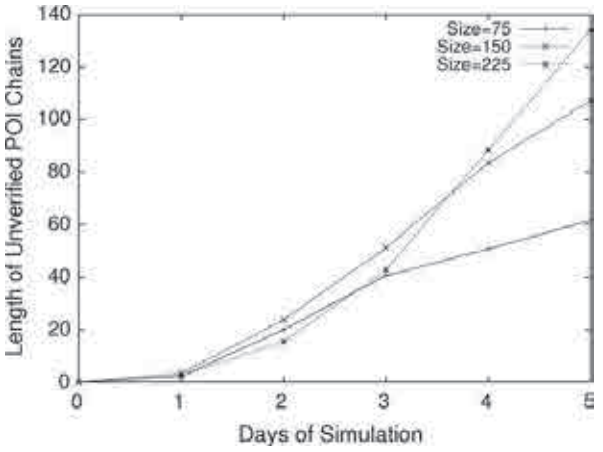
Messages will include POI review chains from different categories. A smart exchange of information is also considered, where depending on external factors some categories will be more represented on the messages than others, i.e., gathering information about restaurants will be prioritized at lunch and dinner time, about gas stations when the vehicle is running low on gas, etc.

The following three types of message exchange are considered:

1. *Requests*: if a vehicle receives a review request he will reply with several POI chains for the requested category. Preferably, reviews that he has verified himself and which have the highest rate in the category. If not enough verified chains are available, he will reply with the highest rated unverified chains (following the rating criteria described in Section 3.3.1). When the requester receives the reply he considers all the chains in the message as unverified and stores them as such. Hence, the difference between verified and unverified chains (in the response) becomes subtle: only the verified chains have the sender's signature, whereas the unverified are just being forwarded. A user will not be penalized nor rewarded for forwarding unverified chains.
2. *Periodic Exchange*: vehicles should exchange POI chains periodically with the better rated POIs in each category. Our scheme prioritizes the recommendation of which POIs another user should visit over which POIs it should not. We would like to avoid a situation where a user knows many POIs with bad reviews and only a handful with good ones.



(a) Length of unverified POI chains: for every node the mean of its unverified POI chains length is computed, the mean of those means is the depicted result.



(b) Length of verified POI chains: for every node the mean of its verified POI chains length is computed, the mean of those means is the depicted result.

Mean of the deviation σ						
Chain type	Length	Day 1	Day 2	Day 3	Day 4	Day 5
Unverified POI chains	75	2.17	22.24	29.13	26.15	23.82
	150	1.59	25.90	47.16	59.93	54.01
	225	2.09	18.80	63.11	76.52	83.49
Verified POI chains	75	1.56	22.70	31.36	28.86	24.78
	150	1.64	30.11	47.73	49.21	51.95
	225	1.08	30.79	49.64	72.80	80.65

(c) Mean of the deviation table for (a) and (b)

Fig. 10. Evolution of the lengths of unverified and verified chains.

3. *Recognition Exchange*: if during a periodic exchange, one vehicle is recognized as a trusted user (from a previous encounter) then recognizer and recognized will exchange user chains and verified POI chains, although they will be marked as unverified by the receiver. Besides, the nodes and its level of trust included in the node chains will be added to the list of the previously known nodes, as explained below.

Requests and periodic exchanges of information are of vital importance for a user that is traveling or moving through a new area. They will both provide the user with unverified chains and once he reviews one POI in one of those chains he will be able to establish a level of trust for each of the reviewers, thus staring a new *Web of Trust*.

Fig. 4 depicts a *Recognition Exchange* between a user S and a user R, in which S sends a message M with his most trusted nodes.

$$R_U = K_{pub_U} \| Level\_of\_Trust_U(as\_POI\_reviewer) \tag{7}$$

$$M = R_{U_1} \| \ldots \| R_{U_5} \| Timestamp \| \{H(R_{U_1} \| \ldots \| R_{U_5} \| Timestamp)\}_{K_{priv_S}} \| K_{pub_S} \tag{8}$$

User R adds U_1 and U_4 to the list of known nodes, with S as their recommender, λ_S^R as the level of trust R has on S as recommender and with an initial level of trust defined by the function:

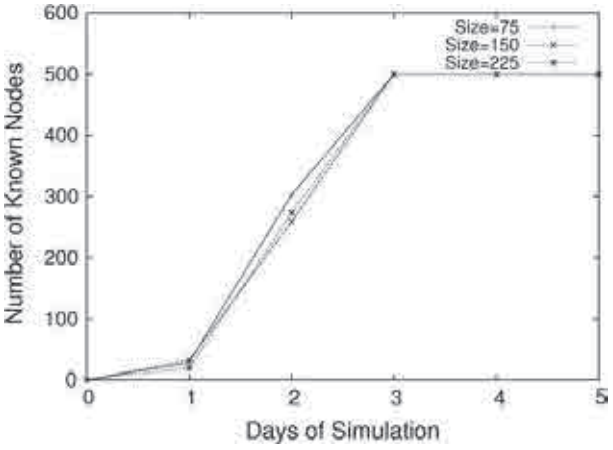
$$T(U_i) = max\left(Trust_{U_i}^E, min\left(\lambda_S^R, Trust_{U_i}^R\right)\right) \tag{9}$$

After R has had a chance to receive several reviews from U_1 and U_4, S will be rewarded or penalized, depending on how similar is the level of trust that R has on them related to what S recommended. All nodes recommended by the same user are inextricably linked, i.e., the misbehavior of one may affect the others. In order to deal with misbehavior, trust on a certain node U_i is divided in the trust recommended by another user $\left(Trust_{U_i}^R\right)$ and the level of trust result of R own experience with U_i $\left(Trust_{U_i}^E\right)$. In this way, when a node misbehaves its recommender is punished and the λ factor (Eq. (9)) is decreased for all nodes he recommended. However, $Trust_{U_i}^E$ will not be affected, and as a result nodes that have earned a reputation for themselves are no longer subject to the reputation of their recommender.

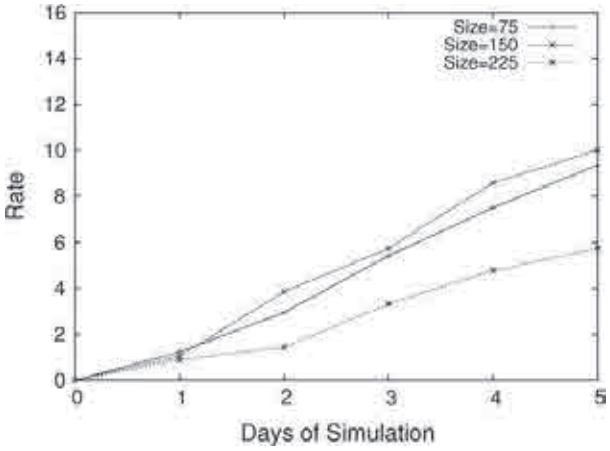
Finally, it should be noted that for every user in the recommended nodes message the receiver only processes those nodes he does not know: if a node knows another user, it means he has followed one of his recommendations and that is more important than a recommendation another user could make.

3.6. The visitor scenario

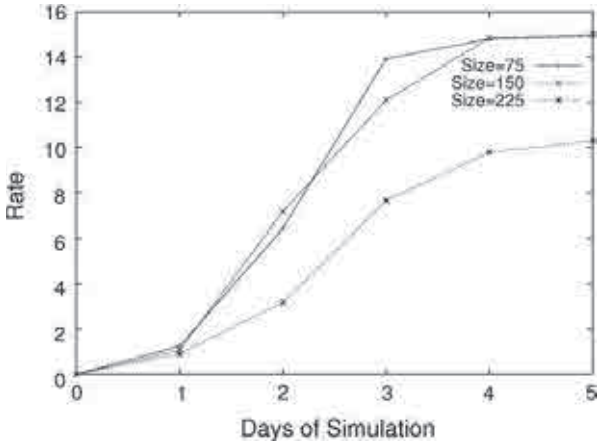
Whenever a user enters a new area and he requests a POI to the system, the system will send a request message and will present the received information together with the information received from periodic exchanges. If the user is in a completely new area it may be possible that he does not know any of the reviewers who have sent him POI recommendations for that specific region. Should that be the case, the system (when queried) will present the user a list of POIs with unverified reviews and a list of the pre-loaded POIs for that area. If the user chooses one of the POIs with unverified reviews, when he inputs his review afterwards he will update his level of trust on all the reviewers in the POI chain, thus gaining information on other users and the POIs they signed. If the user chooses



(a) Number of known nodes: mean of the number of known nodes by every node.



(b) Rate or level of trust of the known nodes: mean of the rates users assign to other users as POI reviewers.



(c) Rate or level of trust of the 25% most trusted nodes: mean of the rates users assign to other users as POI reviewers (only for the 25% highest rated nodes).

Mean of the deviation σ						
Data	Length	Day 1	Day 2	Day 3	Day 4	Day 5
Nodes rate	75	2.08	4.91	5.29	5.40	4.94
	150	2.11	3.94	4.51	5.02	4.14
	225	1.44	3.05	3.20	3.59	3.37
Nodes first 25% rate	75	2.08	6.14	1.32	0.41	0.26
	150	2.10	3.21	1.61	0.38	0.13
	225	1.44	3.88	3.40	2.91	2.59

(d) Mean of the deviation table for (a), (b) and (c)

Fig. 11. Number of known nodes and their levels of trust progress.

one of the pre-loaded POIs he will start a new review chain with his review and he will not gain information on other users. The visitor situation needs to be considered in detail, because it may closely match a tourist profile. On one hand, he will be completely new in the area and most or all POI reviewers will be unknown to him. On the other, precisely because he is a tourist he will input reviews more frequently than the average user and that will allow him to fastly develop a new *Web of Trust*.

3.7. Rewards and penalties

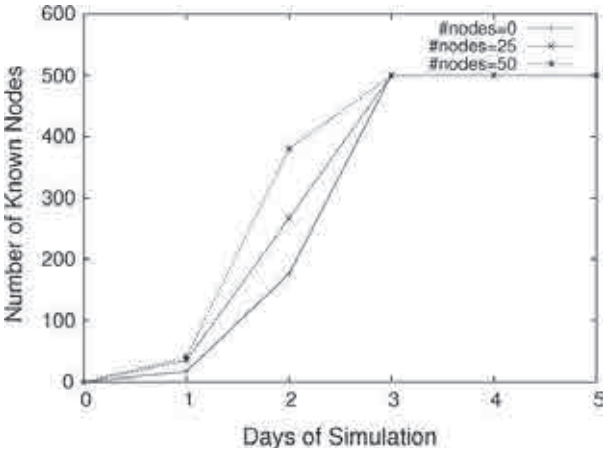
3.7.1. As POI reviewers

Whenever a user *U* receives a recommendation and follows it, he can input his own opinion in the system. Based on that, his vehicle evaluates the recommendation chain updating the levels of trust in other users depending on the similarity of their rates to *U*'s. If *U* has a positive impression of the recommended POI, all the other users in the chain that gave a positive review to the POI are rewarded; otherwise they are penalized. For this system to work, the penalty always has to be greater than the reward; otherwise, a user could cause as much damage to the system as much good he had previously done.

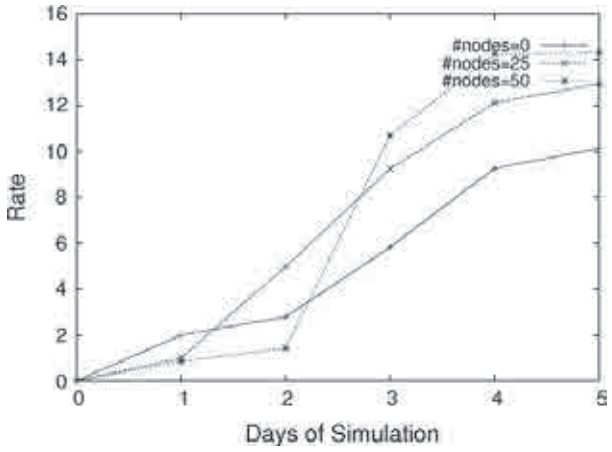
Even though it may seem like that the sole objective of this policy is the punishment of all those users that spread lies and misbehavior in the system, that is inaccurate. Misbehaving nodes is only a part of the problem, i.e., people tastes vary from individual to individual, thus so will their POI reviews. The main goal is not only to build a *Web of Trust*, but also a web of similar tastes, as previously stated.

There are several requirements the penalties system should comply with:

1. If a user *A* has received only a few messages from a user *B* and *B lies to or disagrees with* him, then his level of trust should be significantly decreased.
2. If a user *A* has received many messages from a user *B* and *B lies to or disagrees with* him, then his level of trust should be decreased, but not dramatically.



(a) Number of known nodes: mean of the number of known nodes by every node.



(b) Rate or level of trust of the 25% most trusted nodes: mean of the rates users assign to other users as POI reviewers (only for the 25% highest rated nodes).

Mean of the deviation σ						
Category	#Nodes	Day 1	Day 2	Day 3	Day 4	Day 5
Nodes first 25% rate	0	0.94	1.90	3.49	4.46	4.21
	25	0.97	4.39	2.22	2.15	1.58
	50	1.78	1.99	3.89	0.98	0.94

(c) Mean of the deviation table for (b)

Fig. 12. Number of known nodes and their levels of trust progress.

3. The system should be able to recognize misbehaving strikes, after several lies or disagreements in a row the level of trust in the misbehaving node should plummet.

A very good candidate for the penalties function is the exponential curve because it has a slow growth at the beginning and a steep increase as the rate of lies or disagreements raises, which is appropriate to deal with misbehaving strikes. It was decided that the level of trust should range from 0 to 15 and that after five consecutive bad reviews the evaluator level of trust in the evaluated should be set to the minimum. Thus, e^x was discretized from 0 to 15 into six elements (as depicted in Fig. 5) to obtain the cumulative penalization function $f(x)$, where x is the number of lies. It should be noted that the rating system could easily be modified to operate in another range of values, e.g., from 0 to 10 (which might be more human friendly). The same can be said about the number of bad reviews. What is important is that the penalization function follows the requirements described above and every time a user misbehaves the penalization is greater.

$$\alpha = e^{\frac{1}{5}ln(15)-\beta} \tag{10}$$

where

$$\beta = \frac{\#good\_reviews_{evaluated}}{\#reviews_{evaluated}} \tag{11}$$

$$f(x) = \left(e^{\frac{1}{5}ln(15)-\beta}\right)^x \tag{12}$$

The value that will be subtracted from the level of trust in the beginning of the misbehaving strike is $f(strike\_length)$. As described in requirements 1 and 2 the penalties function should take into account how many good reviews the evaluated user has sent over time, understanding by good reviews those whose rate difference with the evaluator's does not exceed a maximum value defined in the system, which is denoted by ΔOp. To that end β is included in the equation. It should be noted that for recommended nodes, as described in Section 3.5, the level of trust to be decremented shall be both $Trust^E$ and $Trust^R$.

In Algorithm 2 the pseudocode of the rewards and penalties function is presented. Consider $\chi_{U_1,A}$ as the rate user U_1 assigned to POI A. The first time that U_1 finds the difference between his rate and U_2's over a certain POI A is greater than ΔOp it marks node U_2 as misbehaving. The value of $Trust^E$ is stored as the rate at the beginning of the strike from which α^{length_strike} will be subtracted. If a user is in a misbehaving strike his level of trust will decrease faster. A misbehaving strike can be broken after the evaluator verifies *BREAK_STRIKE* good reviews from the evaluated. However, breaking the strike does not mean that the evaluated user goes back to its previous level of trust.

Algorithm 2. Rewards and penalties pseudocode.

```
if ¬misbehaving_strike then
  if |χ_{U1,A} − χ_{U2,A}| ⩽ ΔOp then
    Trust^E := Trust^E + 1
  else
    misbehaving_strike := true
    Trust^E_{pre_strike} := Trust^E
    α := e^{1/5 ln(15) − β_{U2}}
    Trust^E := Trust^E_{pre_strike} − α
    Trust^R := Trust^R_{pre_strike} − α
    strike_breakers := 0
  end if
  else
  if |χ_{U1,A} − χ_{U2,A}| ⩽ ΔOp then
    Trust^E := Trust^E + 1
```

(continued on next page)

```
    strike_breakers := strike_breakers + 1
    if strike_breakers = BREAK_STRIKE then
      misbehaving_strike := false
    end if
  else
    α := α * e^(1/5 ln(15) − β_U2)
    Trust^E := Trust^E_pre_strike − α
    strike_breakers := 0
  end if
end if
```

3.7.2. As node reviewers

As node reviewers, a very similar system to the one described in the previous section will be used. In order to be considered a good recommender in our system, the proportion of good recommendations against bad needs to be at least 5 to 1. If it is, then the user's level of trust as recommender will be increased. If it is less, it will be decreased by α^n (Eq. (10) with $\beta = 0$) where

$$n = 5 \cdot \left\lfloor \frac{\#bad\_recommendations}{\#recommendations} \right\rfloor \tag{13}$$

It should be noted that the timestamp in both types of reviews (POI and other users) allows the system to discard old information and to avoid punishing the user for events that occurred a long time ago.

Finally, as detailed in Section 3.5, by decreasing the level of trust on the user as recommender his recommended nodes level of trust becomes more dependent on $Trust^E$ and less on $Trust^R$.

3.8. Misbehavior

In this section we will elaborate on the different mechanism *Chains of Trust* implements to protect itself from the most common misbehavior or third party attacks.

- *False reviews spamming*: an attacker spreads good POI reviews (e.g., to promote his restaurant) or bad (e.g., to harm his competition). If the attacker is unknown to the rest of the users, then their level of trust in him will be 0 and as explained in Section 3.3.1 his unverified POI chain will go mostly unnoticed. On the other hand, if the attacker has previously worked on gaining a certain reputation as POI reviewer, then the penalties system described in Section 3.7 will deal with the attack. As depicted in Fig. 5, we can see that a few bad reviews are enough for a user to lose all his credit, e.g., after three bad reviews its reputation is decreased by 5.08 units. As a result, it can be concluded that the attack fails because the number of well intentioned reviews the attacker needs to send to build a reputation is much greater than the number of ill intentioned reviews he can send before he loses his reputation. It should also be noted that even if the attacker tried to use multiple identities to increase the length of the chain the same reasoning would apply and the attack would fail.
- *Nodes recommendation*: an attacker could create multiple identities, use one to recommend the others and use the latter to implement the *False reviews spamming* attack. As stated in Section 3.5 only the nodes unknown to the nodes recommendation message receiver are added into his list of known nodes and they are added with a level of trust defined by Eq. (9). If the attacker gains a good reputation as recommender and then recommends a list of his own identities, thus constructing a web of misbehaving nodes, after several incorrect messages all of its recommendees' level of trust as POI reviewers will be based on $Trust^E$. Unless the recommended node has earned a reputation for himself, his level of trust as per Eq. (9) will be 0 and it would be as if it had never been recommended, rendering the attack unsuccesful.

In addition to what has been said above, the difficulty of launching an attack on a mobile target should also be considered. Due to the lack of road side infrastructure the attacker could not rely on compromised RSUs to help him launch a global scale attack and would have to use his own resources.

3.9. Analysis of Chains of Trust scalability

The first step to determine if *Chains of Trust* can succeed in real life is to simulate the data transmission protocol for hundreds of nodes. To that end, a simulation in Ns-3 [23] was implemented defining a vehicular scenario with 400 nodes arranged in 4 lanes as depicted in Fig. 6, connected through a WAVE-DSRC 27 Mbps link with a 120 m range. WAVE-DSRC has the mechanisms to provide different user applications with different channels while reserving certain channels for safety applications, others for control and others for public safety [24]. It should be noted that our simulation uses Ns-3 *YansWifiPhyHelper* and *YansWifiChannelHelper* classes, as defined in [25].

Algorithm 3. Data transmission simulation pseudocode.

```
period := 60,120,180,240,300 s
numPackets := 100,200,300,400,500,600,700,800
for every element in numPackets
    for every element in period
      runSimulation(numPackets,period)
    end for
end for
function runSimulation(numPackets,period)
        setupWifi()
        setupVehiclesTopology()
        time := random(0.1,period)
    //schedules an event on time 'time' to send
  'numPackets' packets of a 1000 bytes
     Simulator::ScheduleEvent(time,numPackets,1000)
    end function
```

In Algorithm 3 we can see the simulation pseudo-code. Basically, the program schedules the broadcast of *numPackets* 1000 bytes packets at a randomly chosen time between the start of the simulation and its ending point, defined as *period*. For every scenario (*numPackets*|*period* combination) the number of broadcasts received by each

of the 400 simulated nodes is computed ($results_{numPackets,period}$) and compared with how many broadcasts each of those nodes would have received without packets loss ($reference_{numPackets}$), considering the mean as the scenario's result:

$$Received\ broadcasts\ \% = \sum_{node=1}^{400} \left(\frac{results^{node}_{numPackets,period}}{reference^{node}_{numPackets}} \right) \quad (14)$$

Looking at the results in Table 1 it can be seen that to ensure a delivery rate over 90% while maximizing the amount of information being transmitted 400 packets is the best option for a 120 s period. For a larger number of packets there is a drop in reception due to the MAC collisions. If a shorter period is considered there is a slight drop in performance, although the major reason against transmitting every 60 s is limiting the amount of information an attacker can collect while following a target. We believe 120 s is more secure since the attacker has to be in range twice as long, while at the same time *Chains of Trust* can produce satisfactory results, as will be showed in Section 5.

4. poiSim: the simulation tool

4.1. General description

Once *Chains of Trust* has been defined, it needs a realistic simulation tool to estimate its success in the real world. Simulation tools like Glomosim or ns-2 were discarded because in order to simulate hundreds of thousands of nodes they require a massive amount of memory. Thus, we were inclined to design our own simulation tool. Like in [26], it was decided to analyze the realistic vehicular trace produced by the Multi-Agent Traffic Simulator (MMTS) developed by K.Nagel at ETH Zurich. The MMTS is capable of simulating public and private traffic over real regional road maps of Switzerland with a high level of realism. It models the behavior of people living in the area, reproducing their movement (using vehicles) within a period of 24 h. The decision of each individual depends on the area it lives in. The individuals in the simulation are distributed over the cities and villages according to statistical data gathered by a census. Within the 24 h of simulation, all individuals choose a time to travel and the mean of transportation according to their needs and environment, e.g., one individual might take a car and go to work in the early morning, another one wakes up later and goes shopping using public transportation, etc. All in all, with over 260,000 simulated nodes or vehicles in an area of around 250 km $\times$ 260 km, this mobility trace suited the simulation needs.

According to the data in [27,28] there were 4,012,690 passenger vehicles registered in Switzerland in the year 2008, which means that poiSim simulates only 6.48% of them. Even if we took into account the number of registered vehicles which are not used, we believe the simulated adoption rate of *Chains of Trust* would still be considerably low.

In Section 3.1 it was described how the scheme relies on people's habits in order to construct a *Web of Trust*. The main goal behind designing a specific simulator is to discover if those habits suffice to ensure the application success in a real life scenario. If so, passed the first several days each node in the network should have several POI reviews as well as known nodes. *poiSim* will also be used to analyze how the system behaves when modifying several parameters, e.g., the length of POI chains, or to study how it performs when user chains are not used. It should be noted that *poiSim* is a high level simulator, i.e., it simulates *Chains of Trust* but it does not simulate a *Medium Access Control* protocol for example, it would be unfeasible to simulate wireless communication realistically for hundreds of thousands of nodes in a reasonable amount of time. That is why a separate experiment was conducted in Section 3.9. In addition, the version of *Chains of Trust* simulated by *poiSim* will be slightly different from the original scheme designed in the previous section. The differences will allow the simulator to improve its performance and will not affect the results. They will be explained in this section.

These are several of *poiSim*'s features: it simulates 259,977 nodes and 15,000 POIs. Every node stores:

- Levels of trust on 500 other vehicles.
- 100 unverified POI chains with 225 POI reviews each.
- 150 verified POI chains with 225 POI reviews each.

And for every POI:

- 5000 reviews are stored in the system.

Every POI is assigned a random value ranging from 0 to 15 to be its real rate μ. The rates the users assign to those POIs will be normally distributed around μ with variance $\sigma^2 = 2$ (as depicted in Fig. 7).

Communication wise, a range of 120 m of coverage is considered and every time a vehicle transmits all the vehicles within range receive the message. There are two kinds of simulated messages: periodic and recognition.

1. Periodic Messages: every 120 s a vehicle will broadcast a message with his 25 highest rated verified POI chains, adding unverified POI chains to complete the message if necessary.
2. Recognition Messages: every time a vehicle recognizes another as a trusted user it will send his 25 highest rated verified POI reviews and his 25 most trusted nodes, together with his level of trust in them. Unverified POI chains may be included as well to complete the message if necessary.

It should be noted that *poiSim* does not include request messages, as the original scheme did. The reason is that their implementation would not have changed the simulation experiments, since they are seldom used in respect with periodic messages (every 120 s). As a result, the quality of the system is measured by the number of nodes known to every user and the number and length of verified POI chains every user stores at the end of every simulated day. Due to computational limitations, it was decided not

to simulate user chains as explained in Sections 3.4, 3.5 and 3.7.2, since their simulation would have required the execution of Algorithm 4 for each of the 260,000 simulated nodes and each of the 225 reviewers that unverified and verified chains store. Had user chains not been removed, they would have added a large overhead on the simulation, thus extending the simulation execution time to weeks or even months. As a result, *poiSim* does not simulate misbehavior (since it would not be possible to penalize bad recommenders), hence there is no need to keep track of the recommender-recommended relation and its rewards and penalties policy. We believe misbehavior attempts will be dealt with the mechanisms described in Section 3.8 and will not affect the overall performance. User chains still exist and are transmitted in recognition messages, but the recommender will not be rewarded nor penalized for it. In addition, recommended nodes will be added to the known nodes list with a trust level of 1. In this way, it can be established if the scheme performs satisfactorily or not, and if it does it can be safely assumed that the implementation of user chains will be an improvement, since recommended nodes will be added to the known nodes list with the level of trust with which they were recommended (always greater or equal than 1). It should also be noted that our simulation only contains one POI category, which is enough for the desired testing purposes.

Algorithm 4. User chains algorithm.

```
//every time a POI is reviewed by the user
for every POIReview
//for every reviewer in the POI's unverified chain
  for every unverifiedChain.reviewer
  //if the user and the reviewer's opinion are too different
  if |unverifiedChain.reviewer.rate − user.rate| ⩽ ΔOp
    //update the level of trust in all the nodes recommended by the reviewer's recommender
    reviewLevelOfTrust(unverifiedChain.reviewer.recommender)
  end if
end for
end for
```

In a nutshell, *poiSim* processes each line of the MMTS trace, which contains a $node_{ID}$ and its corresponding x, y, z, t coordinates and updates the vehicles position. On every update it ensures that the vehicles send a periodic message every 120 s, which is a long enough period to avoid causing a tracking vulnerability, and a recognition message when needed. In addition, once a day at most each user reviews a randomly chosen POI from his unverified POI chains, or a completely random POI if there are no unverified POI chains available, as described in Algorithm 5.

Algorithm 5. POI review algorithm

```
if node.reviewedPOIToday() = false then
  node.setReviewedPOIToday(true)
  if random(0,1) = 1then
    if node.unverifiedReviewsTable.isEmpty() = false
    then
      reviewPOIUnverifiedTable ()
    else
      reviewPOIRandom ()
    end if
  end if
end if
```

In order to better study the system, to observe how the POI reviews are exchanged between users, how users build a better reputation for themselves and the effect of several configuration parameters on the simulation, such as the chain length or the number of user reviews in a user reviews message, the 24 h vehicular trace is replayed to obtain a multiple days scenario. It should be remarked that the only common element in every simulated day will be the MMTS trace, because the POIs being reviewed are randomized, and hence will be different in every run.

4.2. Message formats

In this section the message formats and sizes for the simulation will be defined according to the results presented in Section 3.9. Considering the following format for a periodic message M as defined in Section 3.3:

$$R = \{\underbrace{Id}_{88\text{bytes}} \| \underbrace{Rate}_{1\text{byte}} \| \underbrace{Timestamp}_{8\text{bytes}}\} \tag{15}$$

$$R' = \{\underbrace{H(Id)}_{8\text{bytes}} \| \underbrace{Rate}_{1\text{byte}} \| \underbrace{Timestamp}_{8\text{bytes}}\} \tag{16}$$

$$M = \{\underbrace{R_1}_{97\text{bytes}} \| \underbrace{\{H(R_1)\}}_{K_{priv_{Node\ 1}}\ 17\text{bytes}} \| \underbrace{\{R'_2\}}_{K_{priv_{Node\ 2}}\ 17\text{bytes}} \| \dots \| \underbrace{\{R'_n\}}_{K_{priv_{Node\ n}}\ 17\text{bytes}} \| \underbrace{K_{pub_{Node\ 1}}\}}_{128\text{bytes}} \| \dots \| \| \underbrace{K_{pub_{Node\ n}}\}}_{128\text{bytes}} \tag{17}$$

Taking into account that the total amount of information has to be approximately 400,000 bytes, information about 25 POIs will be sent, each containing 107 user's reviews adding up to a total of 390,300 bytes. It should be noted that periodic messages are fragmented in a 1000 bytes packets including certain redundancy, so that if a packet is lost the rest of the message can still be read.

Recognition messages will also contain an user reviews message M':

$$R_{Node\ i} = \{\underbrace{K_{pub_{Node\ i}}}_{128\text{bytes}} \| \underbrace{Level\_of\_Trust_{Node\ i}(as\_POI\_reviewer)}_{1\text{byte}}\} \tag{18}$$

The different simulated reviewing rates determine how fast the POI rate converges to that value.

$$M = \{\underbrace{R_1}_{97\text{bytes}} \| \underbrace{\{H(R_1)\}_{K_{priv_{Node\ 1}}}}_{17\text{bytes}} \| \underbrace{\{R'_2\}_{K_{priv_{Node\ 2}}}}_{17\text{bytes}} \| \ldots \| \underbrace{\{R'_n\}_{K_{priv_{Node\ n}}}}_{17\text{bytes}} \| \underbrace{K_{pub_{Node\ 1}}}_{128\text{bytes}} \} \| \ldots \| \| \underbrace{K_{pub_{Node\ n}}}_{128\text{bytes}} \} \tag{19}$$

Considering that M' contains information about 25 users the message size amounts to 3.371 bytes.

5. Experimental results

5.1. How will Chains of Trust behave in a realistic scenario?

Chains of Trust is designed so that every vehicle is pre-loaded with a selection of a 100 POIs to provide information to users that have arrived to a new area (as described in Section 3.6). However, a user will not transmit a pre-loaded POI unless he visits it for himself, at which point he starts a new chain with his review and can be transmitted. Therefore, the pre-load will not impact in the results presented in this section.

The first test should reveal if the scheme is feasible in a realistic scenario. The simulation will be executed for different reviewing rates, i.e., every user will input a review into the system once a day on average (1/1), once every two days (1/2), once every four days (1/4) and so on until a review is input once every 10 days (1/10).

In Fig. 8a–b the evolution of the number and length of unverified POI chains can be seen. After the first 5 days of simulation the number of unverified chains and its length (number of POIs it contains) is very similar regardless of the reviewing rate. The fact that the average number of unverified chains is over 90 and its length is almost 5 (considering a reviewing rate 1/6) means that there has been interaction between the users and some have already started to build a better reputation in the network. Considering the results after 30 days of simulation it can be seen that they do not differ significantly.

As far as verified chains are concerned in Fig. 8c the direct relation between the reviewing rate and the number of verified chains the nodes store can be observed, which is logical considering that every time a POI is reviewed its unverified chain moves on to the verified state. In the first 5 days of the simulation, the number of verified chains for reviewing rates 1/4 and 1/6 is 1.24 and 0.83 respectively. Similarly, Fig. 8d shows that the progression of the length of verified chains is very close to the unverified depicted in Fig. 8b. Regarding the rate assigned to the POIs in the verified chains, in Fig. 8e it can be observed that the rate of the reviewed POIs varies until it stabilizes around 7, which is expected since the randomly chosen rates are distributed around that value, as described in Section 4.1.

The measure of the system success is given by how many users every user knows and what level of trust he has assigned to them. In Fig. 9a it can be observed that after the first 5 days of simulation every user has several other users in his known nodes list, going from 20.76 users on average for a review rate 1/1 to 2.11 users for a review rate 1/10. As expected, lower reviewing frequencies result in a lower number of known nodes. If a middle ground scenario is considered, review rates 1/4 and 1/6 yield 3.85 and 3.05 known users respectively. Results improve significantly after the first ten days of simulation, where reviewing rates 1/4 and 1/6 result in every node knowing on average 33.24 and 26.37 nodes, respectively.

Regarding the rate or level of trust a user assigns to his known users, in Fig. 9b it can be seen that after the first 5 days of simulation for all reviewing rates the average level of trust is almost 1. As the simulation progresses, the level of trust may oscillate (as it can be seen for reviewing rate 1/1) due to the randomness of the simulation, although on the long run a larger number of chains are reviewed and the level of trust increases due to the higher proportion of good reviews. After the first 10 days, considering review rates 1/4 and 1/6 result in levels of trust of 1.31 and 1.45 respectively. In Fig. 9c we can see represented the level of trust in the 25% most trusted nodes each user has. After the first 15 days it can be seen how its progression differs from Fig. 9b, ending the simulation with a level of trust slightly over 4 for review rates 1/4 and 1/6.

We believe this first experiment has proven that the system will in all likelihood succeed in effectively disseminating POIs information and building a *Web of Trust* among users in a real life scenario. Considering moderate reviewing rates of 1/4 and 1/6 we can see that just after the first 5 days of simulation every user has on average more than 90 unverified chains containing five user reviews, almost 1 verified chain with five reviews and more than three trusted nodes with trust levels over 1. It should also be noted that results significantly improve after 10 days of simulation. Therefore, it can be concluded that although the system will produce results from the very start, depending on the reviewing rate the it may need from 5 to 10 days (in the worst case scenario) to fully develop a *Web of Trust*. Users, however, will be able to take advantage of the application from the start by using as well the collection of pre-loaded POIs.

5.2. Chain size experiments

The length of POI chains is of paramount importance in the system because every time an unverified POI chain is reviewed the reviewer updates his level of trust in all its signers. Hence, the longer the chain the better the system should work. Naturally, the messages cannot be allowed to grow indefinitely because vehicles do not have an infinite amount of memory and the messages exchanged between vehicles should be relatively short due to wireless communication limitations. In Section 5.1, *poiSim* was configured to allow chains up to a length of 225 reviews, but we would like to observe how does the system behave with shorter chains and verify if there is a certain frontier value where the benefits of increasing the length begin to decrease. Thus, the simulator was executed with POI chains of 75, 150 and 225 reviews. In addition, for this experiment every user will input a new review in the system every 1800 s. Certainly, it is not very likely that users will input one new POI review every half an hour. However, once we have stablished the validity of the system, we would like to modify the reviewing rate to study the system in the long run. In Fig. 10a and b, for both unverified and verified POI chains there is a slight difference between using length 150 or 225 after 5 days of simulation. In a 5 days simulation both lengths are high enough to not be a limitation, but in larger runs we would definitely see a bigger difference in the length of chains the vehicles store. Therefore, from *Chains of Trust*'s point of view chains should be as long as the wireless communication between vehicles permits.

Fig. 11a shows a similar growth for the number of known nodes with the three simulated chain lengths. That is because after the first day of simulation most of this growth is a product of the exchange of recognition messages (which do not depend on the chain length). As far as the rates are concerned, in Fig. 11b and c there is a certain variation attributed to the randomness of the simulation, rather than to the chain length. It should be noted that user opinions of POIs are normally distributed with a mean μ that we termed its real rate. As a result, the mean of the rates of 75 reviewers should not differ much from the mean of the rates of 225.

In Fig. 11b and c the levels of trust progress as the simulation advances, although it provides conclusive evidence that longer chains do not lead to more trustworthy nodes. Therefore, in a scenario where recognition messages do not play such an important role on conveying nodes information, POI chains assume that responsibility. Mainly, because every time a chain is verified all the reviewers levels of trust are updated in the verifier. As a result, the length of a POI chain should only be limited by physical requirements such as the size of the message to be transmitted.

5.3. POI vs nodes experiments

The purpose of this experiment is to discern how much of the system performance can be attributed to the exchange of recognition messages, or in other words, how is the system performance affected when node reviews are not exchanged. To that end, the simulation was executed with 0, 25 and 50 node reviews per transmitted message and with a reviewing rate of a new review every 1800 s.

In Fig. 12a the results of those simulations are plotted. As expected, the average number of users known by every user increases as the number of nodes in the message increases as well. However, it should be noted that the maximum number of nodes to be stored (500) is reached in the three cases during the third day of the simulation. Therefore, the exchange of node reviews does not represent a dramatic improvement in that aspect. On the other hand, Fig. 12b shows that the rates of the 25% highest rated nodes improve as a result of increasing the number of nodes in the message. This leads to the conclusion that recognition messages are not critic to the system performance, although they do provide a considerable improvement.

6. Conclusions

This article presents *Chains of Trust*, a new POI information dissemination scheme that builds a reputation system based on people's traffic patterns. From the results presented in Section 5 several conclusions can be drawn. First and foremost, *Chains of Trust* performs satisfactorily in a realistic scenario by rapidly building a *Web of Trust* among its users, even for low reviewing frequencies. Secondly, the length of POI chains is relevant in terms of the number of nodes a user gains information of when verifying a POI review. However, regardless of the length, the mean of the known nodes level of trust remained similar, hence indicating that it does not help to improve the trustworthiness of those nodes. Finally, user chains do help improve the development of the *Web of Trust* once a primary structure of known nodes has been established.

Acknowledgements

This work was partially supported by the EuroNF NoE and by Spanish Grants TIN2010-21378-C02-01 and 2009-SGR-1167.

References

[1] M. Raya, J.-P. Hubaux, The security of vehicular ad hoc networks, in: SASN '05: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM, New York, NY, USA, 2005, pp. 11–21.
[2] D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, W. Schulz, Cartalk 2000: safe and comfortable driving based upon inter-vehicle-communication, Intell. Veh. Symp., vol. 2, IEEE, 2002, pp. 545–550.
[3] C.-H. Yeh, Y.-M. Huang, T.-I. Wang, H.-H. Chen, DESCV – a secure wireless communication scheme for vehicle ad hoc networking, Mob. Netw. Appl. 14 (5) (2009) 611–624.
[4] P.D. Dawoud, D.S. Dawoud, R. Peplow, A proposal for secure vehicular communications, in: ICIS '09: Proceedings of the 2nd International Conference on Interaction Sciences, ACM, New York, NY, USA, 2009, pp. 1026–1032.
[5] K.P. Laberteaux, J.J. Haas, Y.-C. Hu, Security certificate revocation list distribution for vanet, in: VANET '08: Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking, ACM, New York, NY, USA, 2008, pp. 88–89.
[6] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, A. Lioy, Efficient and robust pseudonymous authentication in vanet, in: VANET '07: Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks, ACM, New York, NY, USA, 2007, pp. 19–28.

[7] F. Picconi, N. Ravi, M. Gruteser, L. Iftode, Probabilistic validation of aggregated data in vehicular ad-hoc networks, in: VANET '06: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, ACM, New York, NY, USA, 2006, pp. 76–85.
[8] M. Raya, A. Aziz, J.-P. Hubaux, Efficient secure aggregation in vanets, in: VANET '06: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, ACM, New York, NY, USA, 2006, pp. 67–75.
[9] T. Moore, M. Raya, J. Clulow, P. Papadimitratos, R. Anderson, J.-P. Hubaux, Fast exclusion of errant devices from vehicular networks, in: 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 2008, pp. 135–143.
[10] P. Kamat, A. Baliga, W. Trappe, An identity-based security framework for vanets, in: VANET '06: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, ACM, New York, NY, USA, 2006, pp. 94–95.
[11] A. Wasef, X.S. Shen, Rep: location privacy for vanets using random encryption periods, Mob. Netw. Appl. 15 (1) (2010) 172–185.
[12] S.-B. Lee, G. Pan, J.-S. Park, M. Gerla, S. Lu, Secure incentives for commercial ad dissemination in vehicular networks, in: MobiHoc '07: Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM, New York, NY, USA, 2007, pp. 150–159.
[13] Y. Zhang, J. Zhao, G. Cao, Roadcast: a popularity aware content sharing scheme in VANETS, SIGMOBILE Mob. Comput. Commun. Rev. 13 (4) (2009) 1–14.
[14] Y. Sun, R. Lu, X. Lin, X. Shen, J. Su, A secure and efficient revocation scheme for anonymous vehicular communications, in: 2010 IEEE International Conference on Communications (ICC), 2010, pp. 1–6. doi:10.1109/ICC.2010.5502130.
[15] A. Studer, E. Shi, F. Bai, A. Perrig, Tacking together efficient authentication, revocation, and privacy in VANETS, in: 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 2009, pp. 1–9. doi:10.1109/SAHCN.2009.5168976.
[16] M. Nowatkowski, H. Owen, Certificate revocation list distribution in VANETS using most pieces broadcast, in: Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon), 2010, pp. 238–241. doi:10.1109/SECON.2010.5453881.
[17] B.K. Chaurasia, S. Verma, Maximizing anonymity of a vehicle through pseudonym updation, in: WICON '08: Proceedings of the 4th Annual International Conference on Wireless Internet, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2008, pp. 1–6.
[18] M. Gerlach, F.Güttler, Privacy in vanets using changing pseudonyms – ideal and real (poster presentation), in: Proceedings of 65th Vehicular Technology Conference VTC2007, 2007.
[19] F. Dotzer, Privacy issues in vehicular ad hoc networks, in: Privacy Enhancing Technologies, Lecture Notes in Computer Science, vol. 3856, Springer, Berlin/Heidelberg, 2006, pp. 197–209.
[20] L. Huang, K. Matsuura, H. Yamane, K. Sezaki, Enhancing wireless location privacy using silent period, in: Wireless Communications and Networking Conference, 2005 IEEE, vol. 2, 2005, pp. 1187–1192.
[21] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, K. Sezaki, Caravan: providing location privacy for vanet, in: in Embedded Security in Cars (ESCAR, 2005.
[22] B.K. Chaurasia, S. Verma, Maximizing anonymity of a vehicle through pseudonym updation, in: WICON '08: Proceedings of the 4th Annual International Conference on Wireless Internet, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2008, pp. 1–6.
[23] T.R. Henderson, S. Roy, S. Floyd, G.F. Riley, NS-3 project goals, in: Proceeding from the 2006 Workshop on NS-2: the IP Network Simulator, WNS2 '06, ACM, New York, NY, USA, 2006. <http://doi.acm.org/10.1145/1190455.1190468>.
[24] R.A. Uzcategui, G. Acosta-Marum, Wave: a tutorial, Commun. Mag., IEEE 47 (5) (2009) 126–133.
[25] M. Lacage, T.R. Henderson, Yet another network simulator, in: Proceeding from the 2006 Workshop on NS-2: the IP Network Simulator, WNS2 '06, ACM, New York, NY, USA, 2006. <http://doi.acm.org/10.1145/1190455.1190467>.
[26] V. Naumov, R. Baumann, T. Gross, An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces, in: ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc, ACM Press, 2006, pp. 108–119.
[27] Energy, transport and environment indicators – eurostat pocketbooks. eurostat 2010. <http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-DK-10-001/EN/KS-DK-10-001-EN.PDF>.
[28] Swiss confederation – population size and population composition. <http://www.bfs.admin.ch/bfs/portal/fr/index/themen/01/02/blank/key/bevoelkerungsstand.html>

David Antolino Rivas received his BS in Computer Science from the Technical University of Catalonia (UPC) in 2008. In 2008 he joined an MS program on Security, Cryptology and Coding of Information Systems held by the Grenoble Institute of Technology (INPG – ENSIMAG) and the Joseph Fourier University (UJF) as part of the Socrates/Erasmus program. In 2010 he received his MS from the Computer Architecture Department at the Technical University of Catalonia (UPC) where he is currently pursuing his PhD.
His research interests include network security, wireless networks, and vehicular networks.

Manel Guerrero-Zapata is an Assistant Professor in the Computer Architecture Department (DAC) at the Technical University of Catalonia (UPC).
His research interests include network security, wireless networks, and routing protocols. He is the author of Secure Ad hoc On-Demand Distance Vector (SAODV) routing protocol and of Simple Ad hoc Key Management (SAKM) scheme.
He received his PhD, MS and BS in Computer Science from the Technical University of Catalonia (UPC) in 2006, 1999 and 1997 respectively.
From 1998 to 2003 he worked at the Nokia Research Centre in Helsinki (first as Assistant Research Engineer, then as Research Scientist, and finally as Senior Research Scientist). From 2003 to 2005 he worked as an assistant professor at the Universitat Pompeu Fabra (UPF) in Barcelona.

Simulation

Simulation: Transactions of the Society for Modeling and Simulation International
88(11) 1390–1404

DOI: 10.1177/0037549712456440
sim.sagepub.com

Simulation of points of interest distribution in vehicular networks

David Antolino Rivas and Manel Guerrero-Zapata

Abstract

Over the last few years *Vehicular Ad-hoc Networks* (VANETs) have become a major research topic. Security mechanisms such as *Public Key Infrastructure* have been customized to provide privacy, authentication, integrity and non-repudiation to vehicle communications. Once the security foundations were established, different applications were built on top: intelligent driving systems, parking spot finders, peer-to-peer content, advertisements distribution, etc. In order to verify the feasibility of those applications in the VANET environment, simulation tools such as *ns-2* or *Glomosim* are used, basing their mobility model on non-uniform distributions. The major difficulty for those simulations resides in the complexity of correctly characterizing vehicular mobility at both macroscopic and microscopic levels. This article leaves the generation of mobility traces to simulators such as *Multi-Agent Traffic Simulator* or *VanetMobiSim* and focuses on the implementation of the network application simulator *poiSim*. *poiSim* simulates *Chains of Trust*, a secure points of interest distribution protocol for vehicular networks. This article discusses how, by using customized application simulators, we can obtain more realistic results than by using general network simulators such as *ns-2*. *poiSim* processes a 24-hour mobility trace produced by a *Multi-Agent Traffic Simulator* with over 260,000 nodes, which realistically simulates public and private traffic over regional maps of Switzerland. The result is a *Chains of Trust* simulation, which accurately portrays reality and can be executed in a personal computer. Finally, it should be noted that *poiSim* could easily be modified to simulate other protocols in vehicular networks.

Keywords

simulation, Vehicular Ad-hoc Networks, vehicular, network

1. Introduction

In the near future, *Vehicular Ad-hoc Networks* (VANETs) will change the way we drive. Vehicles equipped with wireless communication devices, also known as *On Board Units* (OBUs), will be able to communicate among themselves and with *Road Side Units* (RSUs). RSUs will compose the road-side infrastructure that will connect the vehicular network to a central system or to the Internet.

With the massive deployment of wireless technologies, the automotive industry will open a wide range of possibilities for drivers and passengers alike: theoretically, anything from finding out the road conditions ahead to watching a movie through streaming should be possible. So, different requirements will lead to the deployment of different kinds of applications over the network. In Raya and Hubaux[1] and Reichardt et al.,[2] applications are classified based on the service they provide.

1. **Safety-related applications**.
   (a) *Traffic information messages*: used to disseminate traffic conditions over an area; they affect public safety only indirectly (they are not time critical).
   (b) *General safety-related messages*: used by public safety applications, such as cooperative driving and collision avoidance (in order to prevent traffic accidents time is certainly an issue; at least they should satisfy an upper bound delay in delivering the information).
   (c) *Liability-related messages*: they are only exchanged in liability-related situations such as accidents. The senders' identities should be kept hidden from the other users in the network and only revealed to the law enforcement authorities (time is not an issue).

Department of Computer Architecture, Polytechnic University of Catalonia, Spain

Corresponding author:
David Antolino Rivas, Department of Computer Architecture, Polytechnic University of Catalonia, UPC-AC D6-212 Campus Nord, C. Jordi Girona 1-3., 08034 Barcelona, Spain.
Email: antolino@ac.upc.edu

2. **Other applications (some examples)**.
   (a) *Toll applications*: electronic toll collection systems such as *AutoPASS* in Norway allow drivers to continue driving without having to stop at tolls.
   (b) *TV and other multimedia content*: used to provide users with entertainment and information (movies, newspapers, etc.).
   (c) *Advertisements*: businesses along the road (such as gas stations and restaurants) could advertise themselves to drivers before they reach their location, giving them enough time to compare different offers.

As far as safety applications are concerned, the integrity and the non-repudiation of the transmitted messages has to be ensured, albeit maintaining at the same time the user's privacy. For instance, a traffic information application needs to make every user accountable for the traffic events he reports, otherwise a misbehaving user would be able to report false events (e.g., traffic jams, accidents, etc.) and redirect traffic to his own benefit. Other applications, for example, multimedia content distribution, may also need to encrypt their messages to avoid eavesdropping from non-registered users. The use of *Public Key Infrastructure* (PKI) will fulfill most security requirements.

PKI is a cryptographic technique that enables users to securely communicate on an insecure public network and reliably verify a user's identity via digital signatures. In a nutshell, every user receives a digital certificate with a pair of keys (public and private), issued and signed by the *Certification Authority* (CA), which uniquely identify him in the network. The CA is responsible for storing and revoking all issued certificates. It should be noted that every user's public key is accessible to everybody else. In this way, if *Alice* wants to send a message to *Bob*, which only *Bob* can read, she encrypts the message with *Bob*'s public key. In addition, if she also wants to make sure that *Bob* knows the message was from her, she signs it with her own private key and appends her certificate with her public key. When *Bob* receives the message, he will first check that *Alice*'s certificate is valid (i.e., not expired and not revoked by the CA). Then, he will verify *Alice*'s signature using her public key and decrypt her message using his own private key.

In order to study the behavior of applications in VANET scenarios, extensive research has been performed in mobility and network simulation fields. Vehicular traffic simulators can be classified in macroscopic and microscopic simulators. The macroscopic perspective considers system parameters as traffic density (number of vehicles per km per lane) or traffic flow (e.g., number of vehicles per hour crossing an intersection) to compute road capacity and the traffic distribution in the road net. In contrast, microscopic simulators determine the movement of each vehicle that participates in the road traffic.

As far as network simulators are concerned, there is a wide variety of available options: *ns-2*,[3–6] *GloMoSim*,[7–10] *OPNET*,[11,12] etc. They are essential tools to simulate network aspects such as communications, routing protocols and wireless propagation models. However, as far as we know, they are not able to handle the simulation of hundreds of thousands of nodes, unlike our simulation tool *poiSim*.

poiSim simulates a secure *points of interest* (POIs) distribution application named *Chains of Trust*.[13] It should be noted that almost anything that could be of interest to a driver could be considered a POI: a museum, a restaurant, a traffic jam, etc. Briefly summarized, in *Chains of Trust* every user or vehicle creates its own pair of public and private keys (of length L), and is responsible for its private key securing; the protocol does not require a CA or any road-side infrastructure. When users visit POIs they evaluate them and input their reviews into the system. The private key is used to sign those POI reviews, whereas the public key is attached to the transmitted information so that the rest of the network can verify the signatures.

In most research articles,[14–17] the authors are aware of network simulator limitations and simulate a low number of vehicles (of the order of a hundred), moving randomly or following a statistical distribution. What we intend to show in this article is that VANET application simulation should be divided into two layers: the first will deal with network-specific aspects such as the Medium Access Control (MAC) layer, which can be simulated by network simulators such as *ns-2* with a comparatively small number of nodes (of the order of a hundred) without affecting the general results, and the second will be application specific, which can be simulated by *poiSim* with a large number of nodes (of the order of hundreds of thousands) while using a realistic mobility trace. We believe that this approach will yield more accurate and realistic results than directly using a network simulator to simulate the application and the network-specific behavior.

The remainder of this work is organized as follows. In Section 2, vehicular traffic, network simulators and the application being simulated, *Chains of Trust*, are further explained. Section 3 introduces the application simulator *poiSim*, followed by a detailed description in Section 4. Finally, the article closes with the *poiSim* simulation results in Section 5 and the conclusions that can be drawn from them in Section 6.

2. Related work

This section introduces certain topics that are required to better understand *poiSim*: vehicular traffic simulators, network simulators and the application *Chains of Trust*.

Every VANET simulation needs to take into account how it characterizes vehicular mobility. It is a key aspect

of the simulation because only by working with realistic descriptions of vehicular mobility can the simulation results be trusted. Providing realistic vehicular mobility descriptions is the responsibility of vehicular traffic simulators.

As far as network simulators are concerned, they allow us to model and simulate our own networks. Usually, they can simulate wired and wireless communications, different signal propagation models, different routing protocols, etc. They are introduced to compare them against *poiSim* and show how it can obtain more realistic results than them.

Finally, this section also explains *Chains of Trust*, since the application-specific layer of *poiSim* has been tailored to its simulation.

2.1. Vehicular traffic simulators

The main goal of vehicular traffic simulators is to realistically portray how vehicles behave on the road. The *Multi-agent Microscopic Traffic Simulator* (MMTS) developed at ETH Zurich[18] is capable of simulating public and private traffic over real regional road maps of Switzerland with a high-level realism (vehicular traces are publicly available from http://www.lst.inf.ethz.ch/research/ad-hoc). The MMTS models the behavior of people living in the area, reproducing their movement (using vehicles) within a period of 24 hours. The decision of each individual depends on the area it lives in. The individuals in the simulation are distributed over the cities and villages according to statistical data gathered by a census. Within the 24 hours of simulation, all individuals choose a time to travel and the mean of transportation according to their needs and environment. For example, one person may take a car early in the morning to go to work while another goes shopping using public transportation in the afternoon.

The street network that is used in the MMTS was originally developed for the Swiss regional planning authority. The major attributes of each road segment are type, length, speed and capacity. The simulation's result is a 24-hour detailed car traffic trace with almost 260,000 vehicles involved, with more than 25,000,000 recorded vehicles direction/speed changes in an area of around 250 km $\times$ 260 km.

Simulation of Urban Mobility (SUMO)[19,20] is another example of microscopic traffic simulation. It simulates how a given traffic demand, which consists of single vehicles, moves through a given road network. The simulation addresses a large set of traffic management topics. It is purely microscopic: each vehicle is modeled explicitly, has its own route and moves individually through the network. It should also be noted that it can extract road topologies from maps obtained from the TIGER database.[21,22]

In *VanetMobiSim*[23] the authors take into consideration multiple factors to produce detailed vehicular movement traces, for example, obstacles, vehicles characteristics, human driving patterns, intersection management, etc. According to the authors, *VanetMobiSim* combines a macroscopic and microscopic approach to produce more realistic results. It should also be noted that, like SUMO, it can extract road topologies from maps obtained from the TIGER database. The authors include as well some interesting results regarding the execution time on an average computer (Intel Core2 Duo at 2.2 GHz with 2 GB of random-access memory (RAM)). *VanetMobiSim* can simulate 5000 vehicles in a 2 km $\times$ 2 km area in over 30 minutes.

For *poiSim* we decided to use the MMTS traces, mainly because they contain 24 hours of over 260,000 vehicles moving over a real map of Switzerland and because they are publicly available for download.

2.2. Network simulators

Harri et al.[24] divide network simulators between commercial based and open source. In the first group we may find *OPNET, QualNet*[25,26] and *OMNet++*.[27,28] They all contain a large number of network protocols for wired and wireless networks. In the second group we may find *ns-2, ns-3*[29] and *GloMoSim* as the most representative network simulators. *ns-2* and *ns-3* are discrete event simulators targeted at networking research, which provide substantial support for simulation of transmission control protocol (TCP), routing and multicast protocols over wired and wireless (local and satellite) networks. *ns-3* is more efficient than *ns-2* and offers new features to help program simulations, although there is still an ongoing effort to port all protocols supported by ns-2. *GloMoSim* has basically the same functionality as *ns-2*, although it simulates fewer protocols due to the smaller *GloMoSim* support community.

The main drawback of the simulators described above is that, to the best of our knowledge, none of them are able to handle simulations of the order of hundreds of thousands of nodes, and therefore process the MMTS traces. Naumov et al.[18] are aware of this limitation and select smaller regions from the trace to run their simulations with *ns-2*. Similarly, Ding et al.,[14] Patwardhan et al.,[15] Dhurandher et al.[16] and Lo and Tsai[17] limit their simulations to several hundred nodes.

Precisely, we designed our own simulation tool to overcome this major limitation and to be able to process the entire MMTS trace.

2.3. Chains of Trust: the application being simulated

2.3.1. Scheme overview.

Chains of Trust is a technique for secure dissemination of POIs information over VANETs, which we presented in Rivas and Guerrero-Zapata.[13] One of its main advantages is that it does not require a CA or any road-side infrastructure. It relies on a reputation system or *Web of Trust* based on human driving patterns.

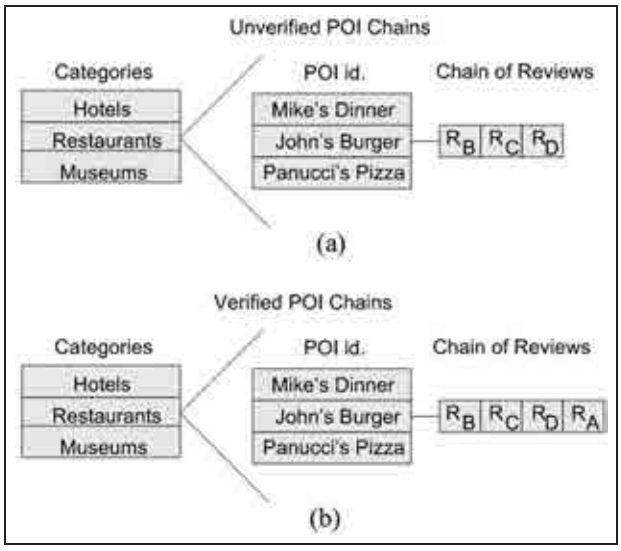


Figure 1. Point of Interest (POI) chains organization. (a) Unverified POI chains organization. (b) Verified POI chains organization.

In the reputation system, every vehicle needs to store information about other vehicles and POIs (whether received from other users or reviewed by him). Every node in the network shall store the following.

- POI chains: these are a series of reviews of the same POI from different users. Whenever a user queries his vehicle for a POI recommendation, verified and unverified POI chains are returned as a result. As depicted in Figure 1, POI chains can be divided into the following.
  - Unverified POI chains: these contain POI reviews that the user has received from other users but which he has not yet been able to verify (by visiting and rating the POI himself), for example, a traffic jam alert or the review of a new restaurant. When the user queries the vehicle for a POI recommendation a selection of unverified chains, ordered by how much the user trusts the reviewers in the chains, is returned.
  - Verified POI chains: once the user has a chance to check if there really is a traffic jam or how good the recommended restaurant is, he evaluates the reviewers in the unverified chain and updates his level of trust in them depending on how truthful they were and marks the chain as verified. Verified chains are an essential part of the exchange of information between users, as will be explained in Section 2.3.3.
- Trust levels in other users (per category): every node needs to remember how much he trusts other users based on the verification of previous reviews. Besides, nodes not only share information about POIs, but also information about other nodes.
- Information about the latest messages from every user, both about POIs and nodes, should be stored for misbehavior detection.

Every user or vehicle will create its own pair of public and private keys (of length L) and will be responsible for its securing (making a CA unnecessary). It should be noted that the public key *Kpub* is also the user identifier; therefore, L should be long enough to ensure the statistical uniqueness of identities. The private key will be used to sign POI information and levels of trust that a particular user has in the others, while the public key will be attached to that information so that the rest of the network can verify the signature's correctness. For instance, consider the scenario depicted in Figure 2. A user *U*1 goes to a hotel *Hotel 1* and he likes it. *U*1 will broadcast a message to the other users in the network saying that *Hotel 1* deserves a certain rate χ, signed with his *Kpriv* and attaching his *Kpub*. All the other nodes store the unverified chain for

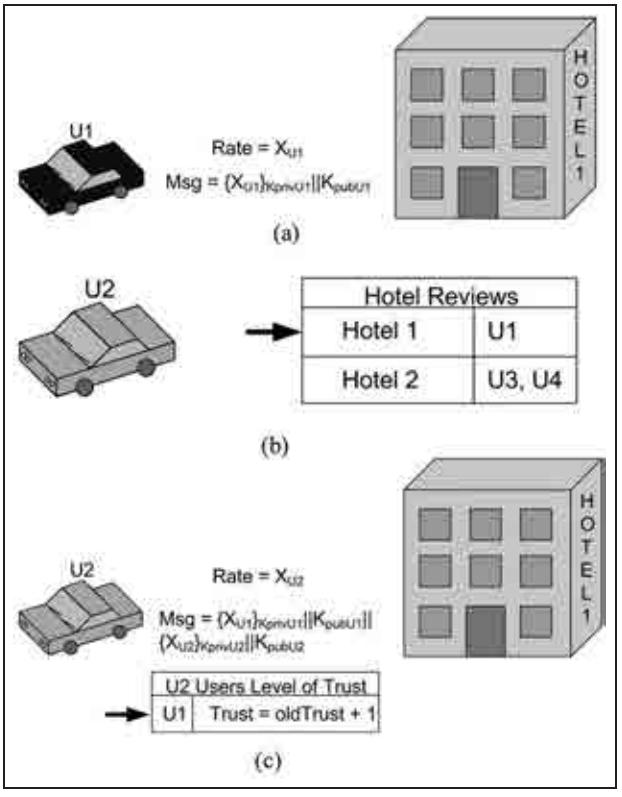


Figure 2. General behavior of the *Chains of Trust* protocol. (a) *U*1 reviews *Hotel 1* and broadcasts the message. (b) *U*2 queries his vehicle for a hotel. (c) *U*2 follows *U*1 recommendation, adds his own opinion to the chain and updates his level of trust in *U*1 according to how similar both reviews were.

future reference. When another user $U2$ queries his own vehicle for a place to spend the night, the vehicle returns a list of places recommended by other users (among which is $U1$'s recommendation). If $U2$ decides to go to *Hotel 1* and likes it as much as $U1$, his level of trust in $U1$ will increase, or decrease otherwise. Regardless of how much he coincides with $U1$'s opinion, $U2$ will append his signed review to the original, together with his *Kpub*, and broadcast the message. In this way, every time a user follows and verifies a recommendation he can update his level of trust in n other nodes (where n is the length of the chain of signatures), thus increasing the speed at which the reputation system develops.

2.3.2. POI categories and records. Several POI categories shall be considered, and a different level of trust for each category for each user shall be kept by each vehicle, that is, a user may be a good hotel reviewer and a terrible restaurant critic. The following are examples of what may be considered a POI category: traffic conditions, gas stations, grocery stores, restaurants, etc.

For each category a validity period is defined, for example, a hotel review may be valid for months, whereas a traffic jam alert may expire within hours. That validity period is necessary to prevent unfair punishments. For instance, if a user identifies a traffic jam and sends a message alerting the network and several hours later another vehicle passes by and sees no trace of it he should not decrease his level of trust in all the users who signed the alert message.

Before POIs can be reviewed they must first be given a unique identifier consisting of common knowledge information:

$$Id = \{Category||POI\_Name|| \\ Postal\_Address||GPSCoords\} \quad (1)$$

The Postal_Address and the GPSCoords fields complement each other, since it is difficult to give the postal address of a traffic jam (although possible using road markers) or the GPS coordinates of a restaurant (unless you position your vehicle right at the door). It should be noted that the GPS coordinates will admit a certain margin of error due to the device's positioning error.

Whenever a user reviews a POI, he assigns a rate to it and assembles a record R with the following information:

$$R = \{Id||Rate||Timestamp\} \quad (2)$$

Each record has a timestamp so that users are able to keep track of the validity period per category. In addition, it could also be used to remove old entries from the trusted nodes table. Once the record has been prepared, the sender needs to sign it (by encrypting the record's hash with his private key) and attach his public key to it. At some point in the future the vehicle will broadcast M:

$$M = \{R1||\{H(R1)\}KprivA||KpubA\} \quad (3)$$

Afterwards, when a vehicle receives a message it stores it for future use. When a user queries his vehicle for a recommendation on a POI category in a certain area, the system answers with a list of received POIs. If the user follows the recommendation he will be able to write another review about the recommended POI. The idea is to keep the previous reviews and attach the latest to the group, thus forming a chain of signatures that grows until a parameter n. By keeping a chain of size n, every time a user follows a recommendation he will be able to update his level of trust in n other users. It should be noted that the new added records are a slightly modified version of the first because they contain the hash of the original POI *Id*, instead of the complete identifier:

$$R' = \{H(Id)||Rate||Timestamp\} \quad (4)$$

A message containing a chain of length 2 is of the form:

$$M = \{R1||\{H(R1)\}Kpriv_A||\{R'2\}Kpriv_B||Kpub_A||Kpub_B\} \quad (5)$$

2.3.3. The information exchange. The application is designed to disseminate information about POIs among the vehicles in the network, thus the need for that information to flow from one vehicle to another. On one side there are POI chains (both verified and unverified) that represent the new information that comes into the system in the form of reviews of new POIs plus the re-evaluation of the already known. On the other, there are user chains, which are lists of known nodes and their level of trust. Basically, once two nodes know each other, besides exchanging information about POIs, they can exchange information about other users, thus increasing the speed at which the *Web of Trust* develops.

The following two types of message exchange are considered in *poiSim*:

1. Periodic Exchange: vehicles should periodically exchange (every 120 seconds) POI chains with the better rated POIs in each category;
2. Recognition Exchange: if during a periodic exchange, one vehicle is recognized as a trusted user (from a previous encounter) then recognizer and recognized will exchange user chains and verified POI chains, although they will be marked as unverified by the receiver.

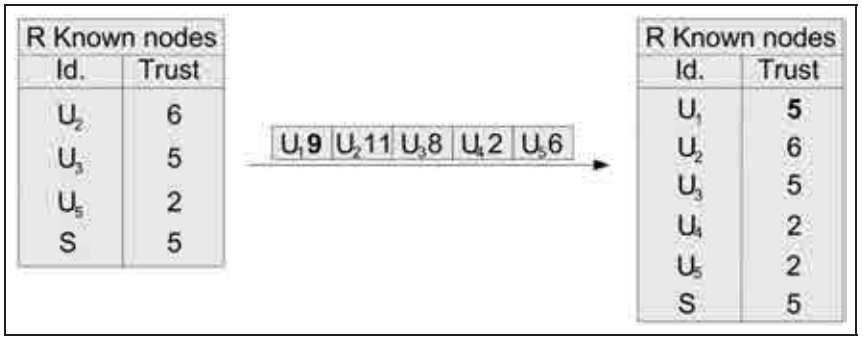


Figure 3. R's known nodes table before and after processing a Recognition Exchange message.

In addition, the nodes and their level of trust included in the node chains will be added to the list of the previously known nodes, as explained below.

Figure 3 depicts a *Recognition Exchange* between a user S and a user R, in which S sends a message M with his most trusted nodes:

$$RU = Kpub_U || Level\ of\ Trust_U(as\ POI\ reviewer) \quad (6)$$

$$M = R_{U1} || \ldots || R_{U5} || Timestamp || \{H(R_{U1} || \ldots || R_{U5} || Timestamp)\} Kpriv_S || Kpub_S \quad (7)$$

User R adds U_1 and U_4 to the list of known nodes with the level of trust S recommended, provided that it is not greater than the level of trust R has on S itself.

Due to the information exchange, users will tend to have the same reputation in groups of users with the same taste. For instance, consider a user who is a good fast food critic but a terrible shellfish restaurant reviewer. In the long run, this user will have a good reputation among other users who like fast food and a bad reputation among users that like shellfish. The *Recognition Exchange* accelerates that process by allowing users to send and receive recommendations of other users as POI reviewers. In the previous example, the fast food reviewer will be recommended in *Recognition Exchanges* between fast food enthusiasts, which will make it more likely that another user follows one of his recommendations and therefore increases his level of trust in him. On the other hand, he will never be recommended between users that like shellfish, which will make it less probable that another user follows one of his reviews.

Finally, it should be noted that whenever a user receives a *Recognition Exchange* message he will only process recommendations for users he does not know yet: if one user knows another, it means he has followed one of his POI recommendations and that is more important than any recommendation he could receive from other users.

2.3.4. Rewards and penalties. In order to build a reputation system from the POI and users' recommendations discussed in the previous section, a rewards and penalties policy is required. Whenever a user U receives a recommendation and follows it, he can input his own opinion in the system. Based on that, his vehicle evaluates the recommendation chain updating the levels of trust in other users depending on the similarity of their rates to U's. If U has a positive impression of the recommended POI, all the other users in the chain that gave a positive review to the POI are rewarded; otherwise they are penalized. For this system to succeed the penalty always has to be greater than the reward; otherwise, a user could cause as much damage to the system as good he had previously done.

A very good candidate for the penalties function is the exponential curve because it has a slow growth at the beginning and a steep increase as the rate of lies or disagreements raises, which is appropriate to deal with misbehaving strikes. By definition, the level of trust ranges from 0 to 15 and after five consecutive bad reviews the evaluator level of trust in the evaluated reviewer should be set to the minimum . Thus, e^x was discretized from 0 to 15 into six elements (as depicted in Figure 4) to obtain the cumulative penalization function $f(x)$, where x is the number of lies:

$$\alpha = e^{(\ln(15)-\beta)/5} \quad (8)$$

where

$$\beta = \#good\_reviews_{evaluated} / \#reviews_{evaluated} \quad (9)$$

$$f(x) = (e^{(\ln(15)-\beta)/5})^x \quad (10)$$

The value that will be subtracted from the level of trust in the beginning of the misbehaving strike is $f(strike\_length)$. The penalties function should take into account how many good reviews the evaluated user has sent over time, understanding by good reviews those whose rate difference with the evaluator's does not exceed a maximum value defined in the system, which is denoted by ΔOp. To that end β is included in the equation.

In Algorithm 1 the pseudo-code of the rewards and penalties function is presented. Consider $\chi_{U1,\,A}$ as the rate user U_1 assigned to POI A. The first time that U_1 finds the difference between his rate and U_2's over a certain POI A

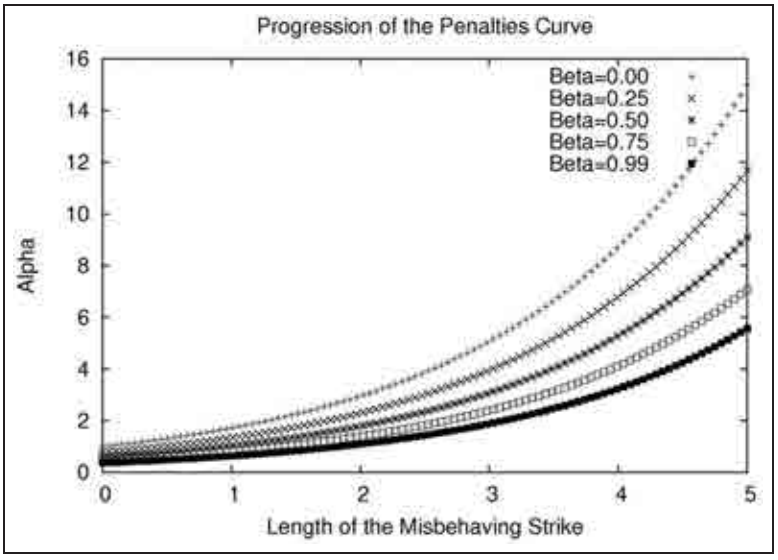


Figure 4. Progression of the function $f(x) = (e^{(\ln(15) - \beta) / 5})^x$.

is greater than ΔOp, it marks node U_2 as misbehaving. The value of *Trust* is stored as the rate at the beginning of the strike from which α^{length_strike} will be subtracted. If a user is in a misbehaving strike his level of trust will decrease faster. A misbehaving strike can be broken after the evaluator verifies *BREAK_STRIKE* good reviews from the evaluated reviewers. However, breaking the strike does not mean that the evaluated user goes back to its previous level of trust.

Algorithm 1. Rewards and penalties pseudo-code.

```
if ¬ misbehaving_strike then
   if |χU1 ,A − χU2 ,A| ≤ ΔOp then
      Trust := Trust + 1
   else
      misbehaving_strike := true
      Trust_pre_strike := Trust
      α = e^(ln(15)/5 − βU2)
      Trust := Trust_pre_strike − α
      strike_breakers := 0
   end if
else
   if |χ_U1,A − χ_U2,A| ≤ ΔOp then
      Trust := Trust + 1
      strike_breakers := strike_breakers + 1
      if strike_breakers = BREAK_STRIKE then
         misbehaving_strike := false
      end if
   else
      α = α* e^(ln(15)/5 − βU2)
      Trust := Trust_pre_strike − α
      strike_breakers := 0
   end if
end if
```

3. *poiSim*: the simulation tool

Once *Chains of Trust* had been defined, it needed a realistic simulation tool to demonstrate that it performed satisfactorily in a realistic scenario. Simulation tools such as *Glomosim* or *ns-2* were discarded to simulate the application, because in order to simulate hundreds of thousands of nodes they require a massive amount of memory. Thus, we were inclined to design our own simulation tool. Like in Naumov et al.[18] it was decided to analyze the realistic vehicular trace produced by the MMTS developed by K Nagel at ETH Zurich.

In Section 2.3.1, it was described how the scheme relies on people's habits to construct a *Web of Trust*. The main goal behind designing a specific simulator is to discover if those habits suffice to ensure the application's success in a real-life scenario. The quality of the system is measured by the number of nodes known to every user and the number and length of the verified POI chains every user stores at the end of every simulated day. It should also be noted that our simulation only contains one POI category, which is enough for the desired testing purposes.

poiSim is a high-level simulator, that is, it simulates the application but it does not simulate a MAC protocol; it would be unfeasible to simulate wireless communication realistically for hundreds of thousands of nodes if we want *poiSim* to be lightweight enough to run on a desktop computer. In Table 1 we can see the features of *poiSim* compared to the network simulator *ns-3*. *ns-3* is better prepared to simulate the network specifics, while *poiSim* is better suited to simulate the application. What we propose is a two-tier approach: use a network simulator such as *ns-3* to test the communication layer with a small-scale simulation (hundreds of nodes) in combination with *poiSim* to

Table 1. Comparison of features in *ns-3* and *poiSim*.

ns-3–poiSim comparison		
Features	*ns-3*	*poiSim*
MAC layer	✓	×
Packet collision/noise simulation	✓	×
Energy simulation	✓	×
Wireless propagation models	✓	×
Routing protocols	✓	×
Processing a third party mobility trace	×	✓
Magnitude of the simulation (in nodes)	≈ 100	≈ 260,000

MAC: Medium Access Control

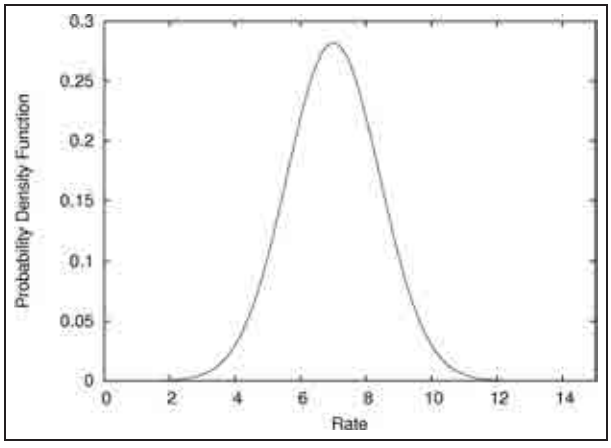


Figure 5. User's rate distribution for the real rate $\mu = 7$ and $\sigma^2 = 2$.

simulate the application in a full scale (hundreds of thousands of nodes).

Communication wise, every node is considered to be equipped with a Wireless Access in Vehicular Environments–Dedicated Short Range Communications (WAVE-DSRC) 27 Mbps link with a 120 meters range and, every time a vehicle transmits, all vehicles within range will receive the message. It should be noted that the simulation of communication-related aspects is outside the scope of this article and was already treated in Rivas and Guerrero-Zapata.[13] For instance, in Rivas and Guerrero-Zapata[13] we prepared an experiment to determine what was the lowest packet delivery rate possible in the network due to the MAC protocol, that is, if vehicles would be able to communicate in a traffic jam. The results showed that in a four-lane road scenario with 400 vehicles separated by 1 meter the packet delivery rate was over 90%.

poiSim considers two kinds of simulated messages: periodic and recognition.

1. Periodic messages: every 120 seconds a vehicle will broadcast a message with his 25 highest rated verified POI chains, adding unverified POI chains to complete the message.
2. Recognition messages: every time a vehicle recognizes another as a trusted user it will send his 25 highest rated verified POI reviews and his 25 most trusted nodes, together with his level of trust on them.

The following is a list of several of *poiSim*'s features:

- simulates 259977 nodes and 15000 POIs.
- every node stores:
  - levels of trust on 500 other vehicles;
  - 100 unverified POI chains with 225 POI reviews each;
  - 150 verified POI chains with 225 POI reviews each.
- for every POI:
  - 5000 reviews are stored in the system.

Every POI is assigned a random value ranging from 0 to 15 to be its real rate μ. The rates the users assign to those POIs will be normally distributed around μ with variance $\sigma^2 = 2$ (as depicted in Figure 5).

In a nutshell, *poiSim* processes each line of the MMTS trace, which contains a $node_{ID}$ and its corresponding *x, y, z, t* coordinates and updates the vehicle's position. On every update it ensures that the vehicles send a periodic message every 120 seconds, which is a long enough period to avoid causing a tracking vulnerability, and a recognition message when needed. In addition, every Δt seconds each user reviews a randomly chosen POI from his unverified POI chains, or a completely random POI if there are no unverified POI chains available.

4. Inside the simulator architecture

This section intends to give a detailed explanation on the design of *poiSim*. Firstly, it explains the motivations of the simulator; for instance, why we decided to prioritize memory over processing optimizations. Secondly, it presents the memory structures used by the simulator and how they were optimized to minimize the amount of memory required. Then, this section explains how *poiSim* processes the MMTS trace and simulates *Chains of Trust*. Finally, the hardware requirements to execute *poiSim* are presented.

4.1. General overview

The simulator has been designed with efficiency in mind, with the emphasis placed on memory rather than on reducing the computing time. The reason for this order in

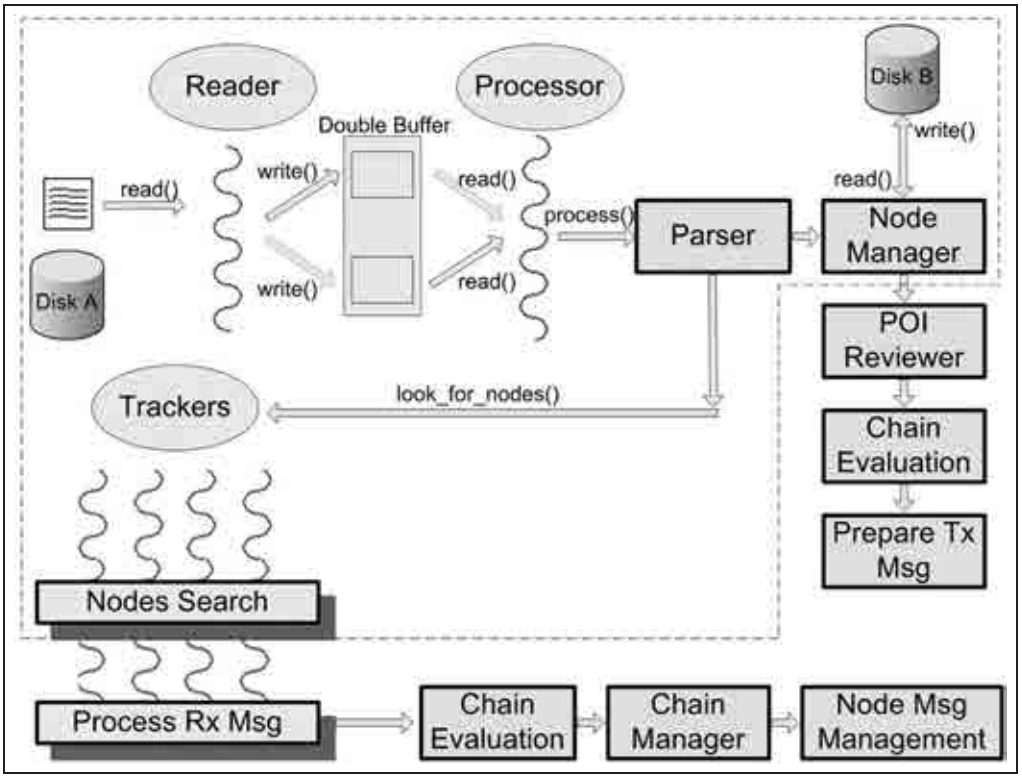


Figure 6. System processes map.

priorities is simple if illustrated with an example. Every simulated node must have a unique identifier and a level of trust, the first ranging from 1 to 260,000 and the second from 0 to 15. In order to store the $node_{ID}$, the simulator is going to use a 4-byte integer, since 2 bytes fall short, and to store the level of trust a single byte will suffice. However, when memory alignment is taken into consideration that single byte turns into 4 (or even 8, depending on the architecture). As a result, every node is now 8 bytes long. Looking at the bigger picture, every node stores the level of trust of 500 other nodes ($8 \times 500 = 4000$ bytes) and over 260,000 nodes are simulated ($260{,}000 \times 4000 = 1{,}040{,}000{,}000$ bytes). Had both fields been stored in the same 4-byte integer it would have been possible to save half that space. It is of paramount importance to grasp the magnifying effect of changes deep in the structure of the simulator. Certainly, by using the same region of memory for both fields every time they are accessed, an additional operation will need to be performed to separate them, which will increase the access time; the alternative, however, is not being able to run the simulation with average computational resources.

Figure 6 provides a clear depiction of *poiSim*'s logical components and processes. The components inside the selected area make up the simulator generic layer, which can be reused to simulate different applications. The components outside are application dependent, and therefore would need to be re-implemented when simulating a different application. More specifically, to simulate a different application with *poiSim* one would only need to re-implement the node's behavior every time its position changes and how it sends and receives messages.

Basically, there is a thread that reads the mobility trace from a disk, block by block, and places it in a double buffer from which another thread feeds on. Those blocks are processed line by line, which are of the form $node_{ID}$, *x, y, z, time, command*. During that processing, the *command* dictates if a node is created, destroyed or updated. Besides, based on the *time*, the simulator checks if the node should review a POI or prepare a message to be transmitted. If the node needs to send a message, a group of threads is notified to look for nodes in range and process the received message, if any are found.

All of these processes will be extensively detailed in the following sections. However, before any further explanations, it should be remarked that even though memory management was our first priority, we were also able to take full advantage of the multi core central processing unit (CPU) at our disposal, by dividing tasks into independent sub-tasks and implementing them in multiple threads so that they could be parallelized.

4.2. Memory snapshot

The mobility trace being used largely determined the memory structures depicted in Figure 7, and that is the

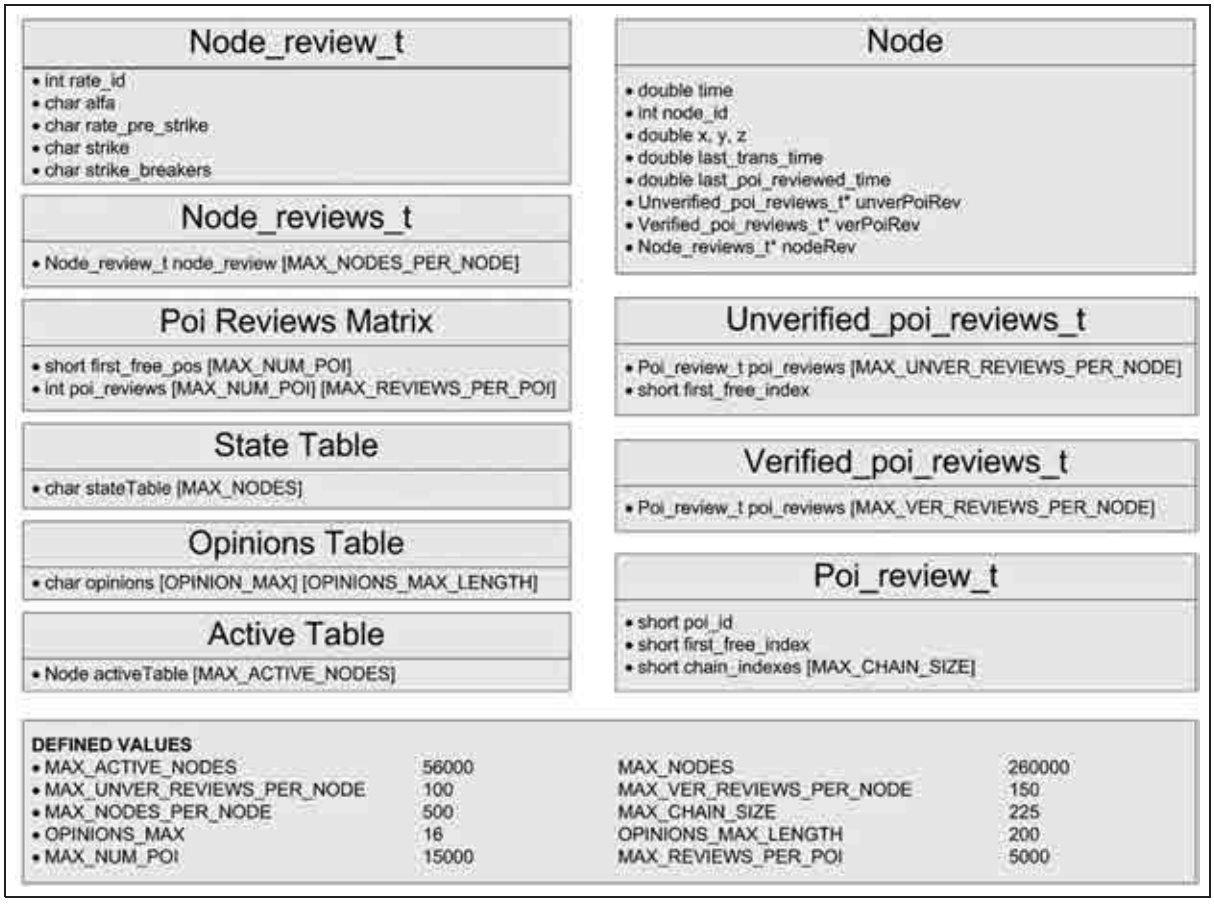


Figure 7. Memory map.

reason why its understanding was so important. That trace describes the traffic patterns of 259,978 vehicles over 24 hours. In that period of time trips began and were ended, hence not all vehicles were traveling at the same time. Several tests were performed to study the trace and it was concluded that 55,197 is the maximum number of vehicles traveling at the same time. As a result, the simulator is designed to store an active vehicle's information in memory while the rest is kept on disk. It should be noted that the exact number of nodes allocated in memory is slightly larger (56,000) to account as well for the simulator's internal operations, so that vehicles can be moved in and out of memory as required by the trace without dragging down the performance.

The second improvement is derived from carefully examining the goal of the application and it affects on the way POIs are stored in chains. The main objective is the dissemination of POIs information over the network, and that information translates into POI chains that in their turn are an aggregate of POI reviews. In other words, many nodes will have common parts of POI chains, that is, repeated POI reviews since what the *Chains of Trust* is trying to accomplish leads to the repetition of information. Therefore, a matrix is designed to hold every review ever created in the system (*Poi Reviews Matrix* in Figure 7) and instead of storing the reviews in the nodes, they only store the indexes to the matrix. This allows the system to save half as many bytes for every repeated review.

In addition, the system is designed to avoid the extra bytes lost to memory alignment, when possible, by grouping pieces of information together. This technique was used in the *Poi Reviews Matrix* to store a user identifier and the rate he assigned to a POI and in the *Node_reviews_t* table of every node to store levels of trust and user identifiers. We would like to remark that the identifiers and the rates (or levels of trust) are fields that, had they not been grouped together, they would have been accessed sequentially. Therefore, the extra selection operations are compensated by one less access to memory. It should also be noted that the *Node_reviews_t* table always has to be ordered by the level of trust (so that the most trusted nodes can be easily found and sent in recognition messages), hence the importance of allocating the rate in the first byte and the identifier in the lower three. Given

Table 2. Size of the memory structures used by *poiSim*.

Memory analysis	
Structure	Size
Node	80 bytes
Unverified_poi_reviews_t	45,402 bytes
Verified_poi_reviews_t	68,100 bytes
Active Table (56,000) nodes	6.58 GB
Poi Reviews Matrix	300 MB
Opinion Table	3200 bytes
State Table	56,000 bytes
Simulation data produced in each simulation (260,000 nodes)	31 GB

two values a and b, $a > b$ if and only if *a.highest byte* > *b.highest byte*. Hence, the ordering operation can be performed disregarding the fact that those bytes contain different bits of information.

The result of those optimizations is displayed in Table 2. It should be noted that the simulation uses over 6.9 GB of main memory (between *Active Table* and *Poi Reviews Matrix*) and produces a volume of 31 GB of data in disk at the end of the simulation, which contains the state of each individual vehicle when it is not traveling.

As far as memory initialization is concerned, it is performed at the beginning of the simulation, even before it starts to process the mobility trace. Most of the memory is allocated dynamically (unverified, verified POI tables and nodes tables), while the rest of the system is stored in static memory. However, nodes are not allocated and freed every time they are created and destroyed. The *Active Table* allocates dynamic regions when it is created and until the simulation finishes it does not free them, mainly to avoid memory fragmentation.

4.3. Processing the trace

Once the memory has been allocated, the simulator can begin reading the trace. The file is read in blocks of 8192 bytes by a thread that copies them into a double buffer. On the other side of the buffer another thread feeds on those blocks and processes them, as described in Listing 1. The idea is to minimize the wait of the *Processor* thread on retrieving the data from disk by having another thread perform the task, while at the same time keeping them both synchronized so that every block is processed. To that end the double buffer is protected with what in pthreads notation are called *Condition Variables*, which is a combination of signals and mutexes: before writing or reading a block from or to the memory structure each thread tries to acquire a lock; if unsuccessful it blocks until the current lock owner sends a signal to indicate that the lock has been released.

Listing 1. Code excerpt to illustrate how the double buffer works.

```
/* Memory definitions */
#define DISK_BLOCK_SIZE 8192
struct disk_double_buffer{
char bufferA[ DISK_BLOCK_SIZE] ;
char bufferB[ DISK_BLOCK_SIZE] ;
short dataReadyA;
short dataReadyB;
} ;

struct disk_double_buffer exchange_buffer;
pthread_mutex_t bufferA_mutex, bufferB_mutex;
pthread_cond_t bufferA_cond, bufferB_cond;

/* readerThread.c - fills the buffer with blocks */
res = fread(block,1,DISK_BLOCK_SIZE,fd);

if(res > 0)
{
     pthread_mutex_lock(&bufferA_mutex);
     if(exchange_buffer.dataReadyA != 0)
     {
        pthread_cond_wait(&buffer A_cond,&buffer A_
        mut ex);
     }
     memcpy(exchange_ buffer.bufferA,block,res);
     exchange_buffer.dataReadyA = res;
     pthread_cond_signal(&bufferA_cond);
     pthread_mutex_ unlock(&bufferA_mutex);
}
/* managerThread.c - processes the buffer */

pthread_mutex_lock(&bufferA_mutex);
if(exchange_buffer.dataReadyA == 0)
{
      pthread_cond_wait(&bufferA_cond,&bufferA_
      mutex);
}
                ...
processBlock(exchange_buffer.bufferA,
exchange_buffer.dataReadyA);
exchange_buffer.dataReadyA = 0;
pthread_cond_signal(&bufferA_cond);
pthread_mutex_unlock(&bufferA_mutex);
```

The *Processor* thread reads the block line by line, translating each and every line into a simulated step. Each of those steps indicates to the simulator that one of the following events has occurred: a trip has begun, a vehicle's position has changed or a trip has come to an end. In the *Node Management* phase, the *Processor* becomes responsible for the interpretation of those instructions, that is, it has to create and destroy nodes as the trace dictates, bringing them from memory to the *Active Nodes* table and back to memory once the trip finishes, besides updating their position when needed.

To speed up the process of looking for nodes in the table, a dictionary was implemented using the $node_{ID}$ mod *MAX ACT IVE NODES* as key. As in any other dictionary, the idea is to check if the *key* position is empty: otherwise move forward to the next one and retry. Notice that, as depicted in Figure 6, the trace and the nodes are stored in separate disks in order to minimize the access latency.

While updating the position and the time of the vehicle, the simulator checks if the user has to review one of his POIs, and if so the thread enters the *POI Reviewer* phase. In this step of the simulation it has to select a *POIID* to be reviewed, which can either be accomplished by randomly choosing one of the unverified chains stored in the node or by randomly generating an identifier if no chains are available. Should that last option be the case things simplify considerably, as detailed below.

- Random POI: this needs to create a review in the *Poi Reviews Matrix* and a new verified chain in which to store that review.
- Unverified POI chain: this needs to create a review in the *Poi Reviews Matrix* too, although in this case this is just the beginning of the process. In the *Chain Evaluation* phase it compares that review with the other reviews in the chain and increases or decreases the node's level of trust on the reviewers based on how much their opinions or rates differ. This rewards and penalties policy follows the process previously described in Section 2.3.4. Notice that whenever the nodes' level of trust is modified, the *Node_reviews_t* table needs to be reordered, which as described in Section 4.2 can be done disregarding the fact that two pieces of information are stored in that region of memory.

Finally, the *Processor* thread verifies if it is time for the user to transmit information to the network. If so, it prepares the messages; otherwise, the processing of that parsed line finishes here. *poiSim* simulates two kinds of messages.

1. Periodic messages are made of 25 POI chains; the highest rated among the verified POI chains the node stores. Should there not be enough, unverified chains will be selected.
2. Recognition messages are made of the highest-rated 25 POI chains and 25 node reviews. Like in periodic messages, verified POI chains can be complemented with unverified chains.

Both messages will be prepared and, depending on the situation, the receiving node will select one or the other.

Once finished with the preparations, the *Processor* thread signals the *Trackers* threads to wake up. The *Active Table*, where all the active nodes are stored, is partitioned into four equal portions (one for each thread) and processed by the *Trackers*, which search for nodes in range. When a node is found, the thread processes one message or the other depending on if the receiver previously knew the sender. This is why it was of paramount importance that everything was prepared beforehand; had it not been done that way, each time a vehicle in range was found its thread would have had to look for the information instead of processing it directly from the message.

Since the messages contain different kinds of information, different paths will be followed when processing them.

- POI chains: when a POI chain is received it is marked as unverified and the thread looks for its POI_{ID} in the node's tables. If it is not found then the received chain is stored in the unverified table. If it is found and the POI chain has not yet been verified, then both chains are merged. Otherwise, if it receives a chain for a POI that it has already reviewed, then it reviews the received chain assigning rewards and penalties to the reviewers, just as was done in the *POI Reviewer* phase, and merges the chains, storing them in the verified table (*Chain Management*).
- Node reviews: a node review is a $node_{ID}$ and a level of trust assigned to that node by the sender. The receiver of the message treats those reviews as if they were his own with two conditions:

  1. the recommended level of trust for a certain node can never be greater than the level of trust the receiver has in the sender;
  2. the recommended level of trust is always decreased by 1, to signify one link in the chain of trust.

Finally, when the *Trackers* have finished processing all active nodes they signal back the *Processor* and the cycle can begin again.

4.4. Hardware requirements

poiSim was executed on a PC running 64 bits Linux Fedora 12, with the following hardware specifications)

- Quad Core CPU Q6600 at 2.40 GHz, with 128 kB of L1 cache and 8 MB of L2 cache;
- 8 GB of DDR2 RAM memory at 887 MHz with latencies 5-5-5-15 (tCL-tRCD-tRP-tRAS)
- 32 GB solid state disk (SSD) to store the operating system (OS) and the mobility trace:
  – 64 MB onboard cache;
  – read maximum performance: up to 210 MB/s;
  – write maximum performance: up to 75 MB/s.
- 96 GB SSD to store the simulation data:

– read maximum performance: up to 285 MB/s;
– write maximum performance: up to 275 MB/s;
– sustained write performance: up to 250 MB/s.

With this hardware, a simulation of the 24-hour vehicular trace lasts approximately 100 minutes.

5. *poiSim* results: how will *Chains of Trust* behave in a realistic scenario?

In order to show the validity of both *poiSim* and *Chains of Trust* we need to study the simulation results. In this test, *poiSim* will be executed for different reviewing rates, that is, every user will input a new review into the system once a day on average (1/1), once every two days (1/2), once every four days (1/4) and so on until a review is input once every 10 days (1/10). The objective of this experiment is to discover how often a user should input a new review for the application to succeed.

The measure of the system success will be given by how many users every user knows and what level of trust he has assigned to them. In Figure 8(a) it can be observed that after the first five days of simulation every user has several other users in his known nodes list, going from 20.76 users on average for a review rate 1/1 to 2.11 users for a review rate 1/10. As expected, lower reviewing frequencies result in a lower number of known nodes. If a middle ground scenario is considered, review rates 1/4 and 1/6 yield 3.85 and 3.05 known users, respectively. Results improve significantly after the first 10 days of simulation, where reviewing rates 1/4 and 1/6 result in every node knowing on average 33.24 and 26.37 nodes, respectively.

Regarding the rate or level of trust a user assigns to his known users, in Figure 8(b) it can be seen that after the first five days of simulation for all reviewing rates the average level of trust is almost 1. As the simulation progresses, the level of trust may oscillate (as can be seen for reviewing rate 1/1) due to the randomness of the simulation, although on the long run a larger number of chains are reviewed and the level of trust increases due to the higher proportion of good reviews. After the first 10 days, review rates 1/4 and 1/6 result in levels of trust of 1.31 and 1.45, respectively.

We believe this experiment has shown that the system will in all likelihood succeed in effectively disseminating POIs information and building a *Web of Trust* among users in a real-life scenario. Considering moderate reviewing rates of 1/4 and 1/6, we can see that just after the first five days of simulation every user has on average more than three trusted nodes with trust levels over 1. It should also be noted that results significantly improve after 10 days of simulation. Therefore, it can be concluded that although the system will produce results from the very start, depending on the reviewing rate it may need from 5 to 10 days (in the worst case scenario) to fully develop a *Web of Trust*.

In addition, these results also show the correctness of *poiSim*, since the more often users input their reviews, the higher are the average number of known nodes per user and their levels of trust.

6. Conclusions

This article presents *poiSim*, a lightweight simulator for a POIs dissemination application in VANETs. It is capable of simulating a 24-hour trace containing almost 260,000 vehicles in approximately 100 minutes. A feat that, to the best of our knowledge, none of the available state-of-the-art simulators (*OPNET, QualNet, OMNet++, ns-2, ns-3* or *GloMoSim*) are able to achieve.

We believe that it has been demonstrated in the article that our two-tier approach based on using a network simulator to test the communication layer with a small-scale simulation (hundreds of nodes) in combination with *poiSim* to simulate the application in a full scale (hundreds of thousands of nodes) yields more realistic results than using a network simulator to test both the network and the application. Whenever a simulation of a large number of nodes is required, the implementation of a customized tool is strongly encouraged as opposed to the use of general network simulators, like is done in most research articles (e.g., Ding et al.,[14] Patwardhan et al.,[15] Dhurandher et al.,[16] Lo and Tsai[17]), always keeping in mind an accurate design and a rigorous memory management strategy.

It should be noted that *poiSim* follows a two-layer customization design. On one hand it has been customized for the *Chains of Trust* application, so that every vehicle stores the required information. On the other, it has been customized to optimally process the MMTS trace. As a result, to simulate a different application we would just need to modify the information the vehicles store and the information being transmitted, while keeping the trace processing and node management layer intact.

7. Future work

In the future, we would like to improve *poiSim* by simulating more than one POI category. This will allow us to modify *Chains of Trust* to implement a smart exchange of information by prioritizing certain POI categories in the *Periodic Exchange* messages. For example, gas stations would be exchanged more often when the vehicle is running low on gas or restaurants when lunch and dinner time are near.

Finally, we would also like to implement different models of user misbehavior; for instance, a restaurant owner trying to spread bad reviews of his competition or a malicious user trying to lower the reputation of another user.

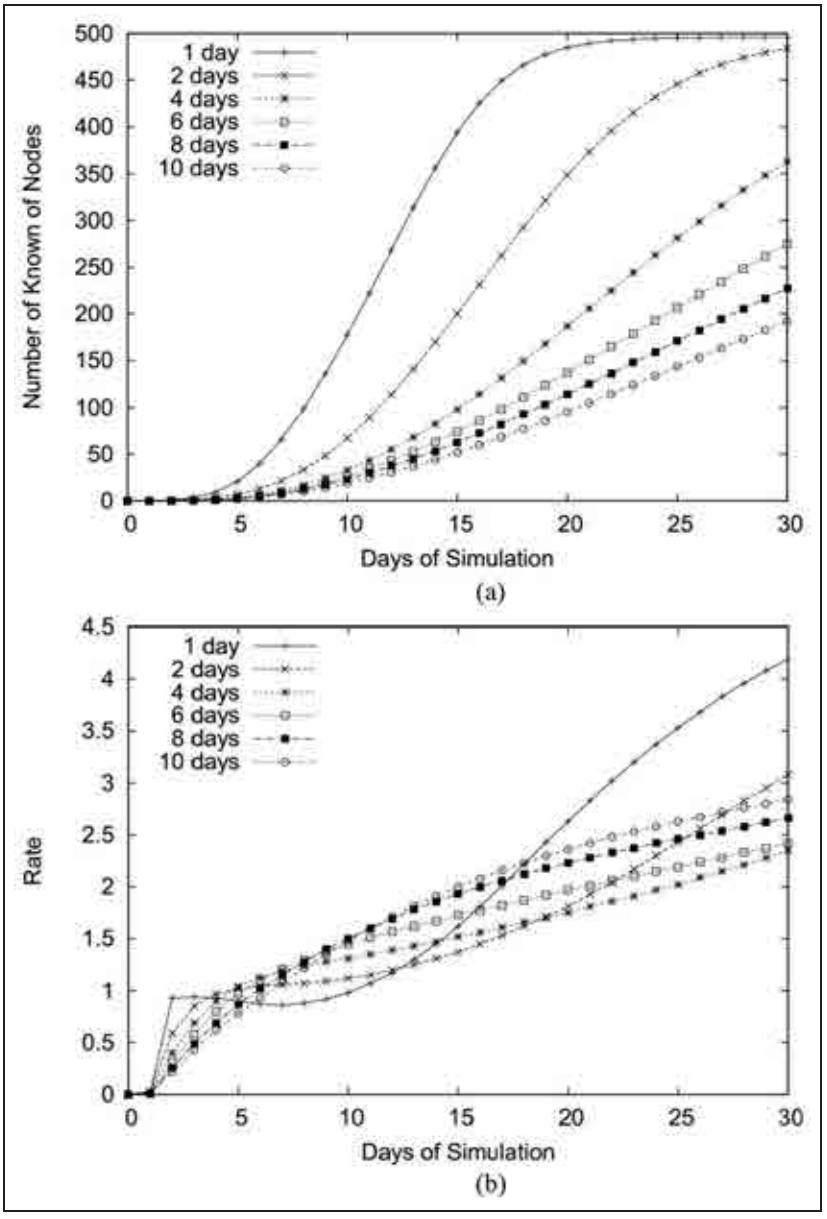


Figure 8. Number of known nodes and their levels of trust progress. (a) Number of known nodes: mean of the number of known nodes by every node. (b) Rate or level of trust of the known nodes: mean of the rate users assign to other users as Point of Interest reviewers.

Funding

This work was partially supported by the EuroNF NoE and by Spanish grants TIN2010-21378-C02-01 and 2009-SGR-1167.

References

1. Raya M and Hubaux J-P. The security of vehicular ad hoc networks. In: *proceedings of the 3rd ACM workshop on security of ad hoc and sensor networks (SASN '05)*, ACM, New York, 2005, pp.11–21.
2. Reichardt D, Miglietta M, Moretti L, et al. Cartalk 2000: safe and comfortable driving based upon inter-vehicle-communication. In: *intelligent vehicle symposium, 2002*, IEEE, Vol. 2, 2002, pp.545–550.
3. Ns-2. The network simulator, http://isi.edu/nsnam/ns/ (accessed 28 July 2012).
4. Chen Q, Schmidt-Eisenlohr F, Jiang D, et al. Overhaul of IEEE 802.11 modeling and simulation in ns-2. In: *proceedings of the 10th ACM symposium on modeling, analysis, and simulation of wireless and mobile systems (MSWiM '07)*, ACM, New York, 2007, pp.159–168.
5. Chen J, Wang C-C, Tsai FC-D, et al. The design and implementation of wimax module for ns-2 simulator. In: *proceedings of the 2006 workshop on ns-2: the IP network simulator (WNS2 '06)*, ACM, New York, 2006.

6. Mahrenholz D and Ivanov S. Real-time network emulation with ns-2. In: *eighth IEEE international symposium on distributed simulation and real-time applications, 2004 (DS-RT 2004)*, 2004, pp.29–36.
7. GloMoSim. Global mobile information systems simulation library, http://pcl.cs.ucla.edu/projects/glomosim/ (accessed 28 July 2012).
8. Zeng X, Bagrodia R and Gerla M. Glomosim: a library for parallel simulation of large-scale wireless networks. *SIGSIM Simul Dig* 1998; 28: 154–161.
9. Royer E, Melliar-Smith P and Moser L. An analysis of the optimum node density for ad hoc mobile networks. In: *IEEE international conference on communications, 2001 (ICC 2001)*, Vol. 3, 2001, pp.857–861.
10. Xu K, Hong X and Gerla M. An ad hoc network with mobile backbones. In: *IEEE international conference on communications, 2002 (ICC 2002)*, Vol. 5, 2002, pp.3138–3143.
11. The OPNET modeler. High definition IT performance management, http://www.opnet.com (accessed 28 July 2012).
12. Chang X. Network simulations with OPNET. In: *simulation conference proceedings, 1999*, Winter, Vol. 1, 1999, pp.307–314.
13. Rivas DA and Guerrero-Zapata M. Chains of trust in vehicular networks: a secure points of interest dissemination strategy, *Ad Hoc Networks 10(6)*, http://www.sciencedirect.com/science/article/pii/S1570870512000285 (2012, accessed 28 July 2012).
14. Ding Q, Li X, Jiang M, et al. Reputation management in vehicular ad hoc networks. In: *2010 international conference on multimedia technology (ICMT)*, 2010, pp.1–5.
15. Patwardhan A, Joshi A, Finin T, et al. A data intensive reputation management scheme for vehicular ad hoc networks. In: *third annual international conference on mobile and ubiquitous systems: networking services*, 2006, pp.1–8.
16. Dhurandher S, Obaidat M, Jaiswal A, et al. Securing vehicular networks: a reputation and plausibility checks-based approach. In: *GLOBECOM workshops (GC Wkshps)*, IEEE, 2010, pp.1550–1554.
17. Lo N-W and Tsai H-C. A reputation system for traffic safety event on vehicular ad hoc networks. *EURASIP J Wirel Commun Netw* 2009; 9: 1–9.
18. Naumov V, Baumann R and Gross T. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In: *proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '06)*, ACM, New York, 2006, pp.108–119.
19. Simulation of urban mobility. SUMO, http://sumo.sourceforge.net (accessed 28 July 2012).
20. Piórkowski M, Raya M, Lugo AL, et al. Trans: realistic joint traffic and network simulator for VANETs. *SIGMOBILE Mob Comput Commun Rev* 2008; 12: 31–33.
21. U.S. Census Bureau. Topologically integrated geographic encoding and referencing (TIGER) system, http://www.census.gov/geo/www/tiger/ (accessed 28 July 2012).
22. Chang X. Network simulations with OPNET. In: *simulation conference proceedings*, Winter, Vol. 1, 1999, pp.307–314.
23. Härri J, Fiore M, Filali F, et al. Vehicular mobility simulation with vanetmobisim. *Simulation* 2011; 87: 275–300.
24. Harri J, Filali F and Bonnet C. Mobility models for vehicular ad hoc networks: a survey and taxonomy. *IEEE Commun Surv Tutorials* 2009; 11: 19–41.
25. QualNet developer. Mission readiness, http://www.scalable-networks.com (accessed 28 July 2012).
26. Roy S, Saha D, Bandyopadhyay S, et al. A network-aware MAC and routing protocol for effective load balancing in ad hoc wireless networks with directional antenna. In: *proceedings of the 4th ACM international symposium on mobile ad hoc networking & computing (MobiHoc '03)*, ACM, New York, 2003, pp.88–97.
27. OmNet. Homepage, http://www.omnetpp.org/ (accessed 28 July 2012).
28. Köpke A, Swigulski M, Wessel K, et al. Simulating wireless and mobile networks in OMNET++ the mixim vision. In: *proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops (Simutools '08)*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, Belgium, 2008, pp.71: 1–71: 8.
29. Henderson TR, Roy S, Floyd S, et al. ns-3 project goals. In: *proceedings of the 2006 workshop on ns-2: the IP network simulator (WNS2 '06)*, ACM, New York, 2006.

Author biographies

David Antolino Rivas received his BS in computer science from the Technical University of Catalonia (UPC) in 2008. In 2008 he joined a MS program on security, cryptology and coding of information systems held by the Grenoble Institute of Technology (INPG – ENSIMAG) and the Joseph Fourier University (UJF) as part of the Socrates/Erasmus program. In 2010 he received his MS from the computer architecture department at the UPC, where he is currently pursuing his PhD. His research interests include network security, wireless networks and vehicular networks.

Manel Guerrero-Zapata is an assistant professor in the computer architecture department (DAC) at the UPC. His research interests include network security, wireless networks, and routing protocols. He is the author of the secure ad hoc on-demand distance vector (SAODV) routing protocol and of simple ad hoc key management (SAKM) scheme. He received his PhD, MS and BS in computer science from the UPC in 2006, 1999 and 1997, respectively. From 1998 to 2003 he worked at the Nokia Research Centre in Helsinki (first as assistant research engineer, then as research scientist and finally as senior research scientist). From 2003 to 2005 he worked as an assistant professor at the Universitat Pompeu Fabra (UPF) in Barcelona.

Anonymous Chains of Trust in Vehicular Networks: Preserving Users Privacy in a Reputation System

David Antolino Rivas*, Manel Guerrero Zapata

Department of Computer Architecture, Polytechnic University of Catalonia, Barcelona 08034, Spain
Telephone: (+34) 934054059

Abstract

This article describes a scheme which, to the best of our knowledge, is the first one to provide anonymity in a reputation system for nodes in a mobile wireless network. The presented solution specifically targets Vehicular Ad hoc Networks where vehicle users share information -opinions or recommendations- about Points of Interest -such as restaurants, hotels, etc. The mechanism used to achieve anonymity allows a user to effectively borrow the identity of another user who trusts him by asking him to issue a recommendation on his behalf. The results show that for moderate frequencies of Points of Interest reviewing on behalf of other users the development of the reputation system remains unaffected.

Keywords: Security, VANETs, Chains, Trust, POI, Reputation

1. Introduction

We live in a world that produces massive amounts of information every day and in order to thrive we need to process them and make the best decisions we can. We rely on friends and family to deal with this complex problem, i.e., whether we are trying to decide where to go for dinner or making a career choice we rely on the experience of other people to help us make a good decision.

This concept lies at the foundation of reputation systems. Since it is not possible to experience everything first hand, a user of a reputation system shares his own knowledge with other

*Corresponding author.
Email address: antolino@ac.upc.edu (David Antolino Rivas)

system users and relies on some of them, preferably ones with a good reputation, to help him make decisions.

A user's reputation will grow with every good decision he helps others make. Naturally, people have different tastes so what may be a good recommendation for somebody may not be so good for somebody else. This leads to the creation of groups of users that trust each other because they have a similar taste, what is called a *Web of Trust*. On the other hand, entities with too different views will recognize each other as not trustable and disregard each other's recommendations. Since what is being shared is subjective information, two people may trust each other today and have different views tomorrow. In addition, they may not trust each other in one area of expertise and at the same time they may share similar views on others.

Reputation systems are increasingly being used nowadays. They are a very good way to bring some order into the chaos that can be a network of users sharing information.

They can be found almost everywhere, in P2P networks, in movie rating websites, in sites like eBay or YouTube, etc. They can be as simple as the one used by eBay -in which after each pair of users conducts a transaction they rate each other and a user's reputation is the count of positive and negative ratings- or they can be extremely complex ones.

Reputation systems, however, are vulnerable to several kinds of attacks [1, 2], one of the most serious being the breach of users privacy. By definition, in a reputation system every user has an identity to which all the opinions he makes public can be traced to. For this reason, an attacker with the appropriate tools should be able to profile all the users in the system: knowing which restaurants they go to, the books they like, having an accurate idea of the area the users live in and even mapping their online identity to their real one.

This article presents a solution to preserve users privacy in reputation systems. In particular, we will apply this solution to the reputation system for *Vehicular Ad-hoc Networks* (VANETs) *Chains of Trust* (CoT), which we developed in [3], although it may well apply to any reputation system.

The remainder of this work is organized as follows. Section 2 gives an overview of reputation systems in VANETs, including CoT. Section 3 explains *Anonymous Chains of Trust* in detail, what is its purpose and how it differs from CoT. Finally, section 4 presents the experiments we have prepared and their results and section 5 closes with the conclusions that can be drawn from them.

2. Related work

This article introduces a *Point of Interest* (POI) -such as restaurants, hotels, museums, etc.- information dissemination technique for VANETs based on a reputation system. To the best of our knowledge, it is the first one to build a reputation scheme that preserves user privacy, mostly due to the fact that users generate and manage their own identities and there is not a central entity where all the network information is stored.

Nevertheless, there are other works that consider the use of reputation systems for different purposes not always keeping in mind users right to privacy.

2.1. Reputation systems

The authors in [4] propose a reputation system to manage traffic warning events while preventing the spread of false information. In their proposal, users/vehicles are divided into different categories according to their proximity to a traffic event and play different roles: (i) *Event Reporter* (ER) is the vehicle that witnesses an event, (ii) *Event Observer* (EO) is any node within one hop distance of an ER and (iii) *Event Participant* (EP) is any node beyond the one hop distance from the ER. Whenever an ER witnesses an event he assigns a *local trust* to it based on the information gathered by the vehicle's sensors. If that value is greater than a certain threshold then he transmits that information to all neighbors in one hop (EOs). When an EO receives a traffic event from an ER he stores it and observes the behavior of the ER. If the ER's behavior matches a model related to the traffic event reported the EO sends this message withing a certain ΔT time, which is enough for him to receive information from other EOs and EPs. At the end of the process every event is assigned a *global trust* based on the ER's behavior and on the *global trust* information sent by other nodes weighted by their role in the event. It should be noted that EPs will base their *global trust* solely on the information gathered from EOs and other EPs since they cannot directly observe ER.

The authors, however, do not take security into account. In their simulation scenario they consider a single vehicle forwarding false messages, which is not realistic since an attacker could easily report the same false event several times with different identities and successfully spread false messages. In addition, their system considers events reputation but not ER's; every role has a fixed reputation or weight assigned to it, which is what is taken into consideration when

computing the event's *global trust*. As a result, there is no way to decrease the trust deserved by an ER who always reports false events.

Similarly, the authors in [5] introduce a scheme to report traffic events in VANETs that respects user privacy by using groups and offers security through trust and reputation. Their idea is to use group membership to provide individual users with privacy outside of the group while the *Group Manager* (GM) is responsible for adding new vehicles and evicting attackers or misbehaving members. A GM is identified by a certificate issued by a *Certification Authority* (CA). Every group has a reputation in the network and every user contributes to it by sending group messages reporting traffic events. The GM has to be able to identify the real identity of the sender of a group message in order to protect the group's reputation against repudiation attacks. The regular flow of events is as follows:

1. Users periodically exchange messages with information of the state of the road.
2. Each receiver verifies that the message has a valid signature from the sender's group.
3. Each receiver computes how much he can trust the message based on the group's reputation and act accordingly, i.e., if he receives a trusted traffic jam alert he will take another route.
4. After taking a decision, the receiver vehicle may be able to know the real state of the road through direct observation. In that case he will update his level of trust on the group or groups that sent him information about this event. False messages are collected and eventually reported to the CA, which forwards them to the responsible GM to take appropriate measures.

Even if the authors do not mention it in their article, we believe that users require access to the CA every time they receive a group message because if a GM is revoked they need to be able to check if his identity is in the *Revocation List* (RL). In addition, the authors do not mention any mechanism for the group members to see if their GM is misbehaving by not evicting misbehaving nodes. We believe too much trust is placed on GMs, which could disclose the group members' identity to third-parties. Moreover, in the event of a traffic jam, only those vehicles which do not heed the warning and have the opportunity to make a direct observation will know the truth. If all users believed an attacker's warning he would be able to completely redirect the traffic on a road and he would not be punished because no other vehicles would be able to directly observe the event. Finally, it should be noted that there is no security mechanism to prevent a user *A* from

lying about an event reported by another user *B*, which would make *B*'s group manager punish *B*.

In [6], the authors propose a general information scheme (not necessarily directed towards traffic events) where every user is not only responsible for the events that he reports but also for the information that he forwards. In this scheme every vehicle is uniquely identified through the use of cryptographically self-generated addresses [7] and the authors assume that their scheme is immune to Sybil attacks. Information can be sent by anchored sources (trusted by default) and by mobile devices (whose level of trust is determined by a reputation system). Mobile devices are accountable for verifying data before propagating it. Therefore, whenever a user receives a message or segment he checks if it was originated at one of his trusted sources, if so that information is automatically trusted. If the segment is received from a source classified as malicious (by reputation) it is immediately discarded. Every time a segment from an unknown source is received a *verification session* starts. If the number of received segments from unknown nodes reporting the same event reaches a threshold value all reporting nodes are promoted to trusted. Similarly, if an unknown user reports the same event that a trusted user, he becomes trusted as well.

We believe that this scheme fails to protect the users' privacy since they always use the same identity (an attacker could easily profile their routes). The authors do not take into consideration that even a trusted originator of an event may be interested in spreading false information at some point. In addition, their proposal heavily relies on anchored resources that only distribute reliable information, which may not be realistic. Finally, the idea of only forwarding information after it has been verified is not without its risks considering the ephemeral nature of a VANET.

The authors in [8] present another solution to distribute safety related information by broadcasting events (traffic jams, accidents and vehicles braking) which uses a reputation system to detect and isolate malicious nodes. Their algorithm is divided into the following phases:

1. *Neighbor discovery*: whenever a node S needs to forward an event received from one of its neighbors, it sends a neighbor discovery request to which its surrounding nodes reply with their identities. Each of the receivers R of that discovery request will check in its trusted nodes table if it trusts S and respond to the discovery request only if it does. If the identity of S is unknown to R then R adds S to its trusted nodes table with a trust level $(MAX\_TRUST - MIN\_TRUST)/2$. Similarly, when S receives the discovery responses

it will update its trusted nodes table following the same criteria.

2. *Data dispatching*: once a node has discovered its neighbors it broadcasts the event information.
3. *Decision making and trust updating*: packets reporting events beyond a certain distance *d* are discarded (far away events are considered irrelevant). The next step is to see if the node itself is in the detection range of the event: if it is, the node will be able to judge if this event is true or false and update his trust on the reporting node accordingly; if it is not in range, it collects information from other neighbors for a time *t* and only if the number of reporters exceeds a certain threshold the event is considered true (either way, after *t* expires the level of trust on the reporters will be updated accordingly).
4. *Neighbor monitoring*: the authors assume that a genuine packet will always be broadcast, whereas false information will be unicast towards a certain node. Based on that, nodes should monitor the network observing its neighbors behavior.

The authors are not clear on whether they use *Public Key Infrastructure* with a CA. If they assign to unknown nodes levels of trust greater than what misbehaving nodes have, it will always be better for an attacker to change his identity once he is discovered. A CA would be able to prevent that by linking the identity to the vehicle's license plate, for example. However, if they use a CA vehicles need to be in permanent connection with it to receive updates on the *Certificate Revocation List* (CRL), which requires a heavy road-side infrastructure. In addition, using always the same identity introduces a tracking vulnerability for the users.

In [9] the authors present a scheme to distribute traffic events information. They define a two tier approach: vehicle sensors first have to detect an event a certain number of times T_S before reporting it to the driver and if they have not detected the event for themselves, they need to receive the event warning from T_V vehicles before trusting it. Every time an event is detected T_S times a message including how many times the vehicle's sensors have detected the event and the identity of vehicles detecting it is send to the vehicle's neighbors. The receiving nodes will use this value and the number of vehicles that detected the event to determine if it is true or not.

We believe that the major problem with this scheme is that it does not address security at all. The authors do not consider the possibility of misbehaving nodes (intentionally or just due to the usual degradation of components). In addition, this solution is an event reputation system, but not a user reputation system, which means that the system has no memory over previous

events recommended by a certain user and therefore all users can be equally trusted, which is a unrealistic assumption.

[10] presents a solution to manage a reputation system in the early stages of VANETs. The authors consider a scenario where the density of *smart* vehicles equipped with wireless communications is too low to allow for *Vehicle to Vehicle* (V2V) communication. As a result, their scheme relies on the distribution of *Road-side Units* (RSUs) to handle the reputation scheme. Ideally, vehicles will always follow the same route (to work places, schools, superstores, etc.) and therefore be periodically in contact with the same RSUs. Depending on the desired deployment cost, the authors distinguish between two different designs:

- Isolated RSUs: if RSUs are not directly connected to each other, they need smart vehicles to forward their messages. This format of communication is called *Delay Tolerant Network* (DTN) [11]. In a nutshell, every vehicle is assigned an *Agent RSU* which keeps track of its reputation and provides the vehicle with a certificate with its updated reputation. The other RSUs will monitor the vehicle behavior, i.e., forwards messages between RSUs, correctly reports traffic accidents to the RSUs, etc. Each RSU will use smart vehicles to forward this information to the vehicle's *Agent RSU* so that it can update the vehicle's reputation.

- Internet-accessible RSUs: in this scenario there is no need to distinguish between the *Agent RSU* and the others. Since they are all communicated, a vehicle can obtain its reputation update from any of them.

The authors also take into account the possibility that a user might take a different route which does not pass by any of his usual RSUs, e.g., he goes to work from Monday to Friday but Saturday and Sunday he drives to a different location. The solution they propose is to increment the validity period of the reputation certificate, so that on Friday the user receives a certificate valid until Monday.

We believe this is an interesting approach to the initial stage of a VANET. However, there are several drawbacks. For instance, road condition alerts will not be delivered immediately upon detection because there is no V2V communication. Secondly, the authors consider a scenario where a user takes an alternative route, although they need to plan ahead so that the RSU can give him an extended reputation certificate. In our opinion, this is not realistic since people are only predictable up to a certain point. Finally, the Internet-accessible RSUs model brings out

the problem of having a network of connected devices which register every move made by every user, thus posing a threat to user privacy.

2.2. Chains of Trust

In [3] we introduced CoT, a secure POI distribution strategy and reputation system for VANETs. In a nutshell, users issue reviews of POIs and broadcast them to the network. The receiving users store them for future use so that when they need information about a certain POI category, e.g., restaurants, museums, traffic events, etc., they can choose one recommendation issued by another node (preferably one they already trust). Whenever they follow one of these recommendations, they issue their own review of the POI and the system updates the level of trust on the recommender(s) depending on how similar their reviews were. In addition, users who trust each other not only exchange information about POIs, but also about other users, i.e., which ones are the most trustable.

User reviews are structured in POI chains, as depicted in Fig. 1. Fig. 1a shows user *A*'s unverified POI chain for *John's Burger*, with the reviews received from users *B*, *C*, and *D*. When *A* visits *John's Burger* and inputs his review into the system that chain turns into verified and *A*'s review is appended to the rest (Fig. 1b).

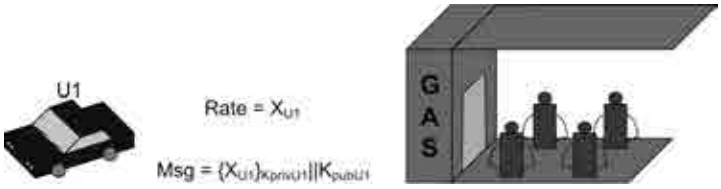
Security wise, the system uses asymmetric cryptography (1,024 bits-RSA). Every user or vehicle creates its own pair of public and private keys and is responsible for its securing. The private key is used to sign POIs information as well as the levels of trust that one user has in the others, while the public key (which serves as user identifier) is attached to that information so that the rest of the network can verify the signatures correctness.

Going into detail, consider the scenario depicted in Fig. 2 where a user U_1 goes to a gas station *A*. U_1 will input his review into the system and the system will create a chain of recommendations of length 1 for that POI. U_1 will broadcast a message containing this chain (verified chain) to the other users in the network saying that gas station *A* deserves a certain rate χ, signed with his private key K_{priv} and attaching his public key K_{pub}. All the other nodes that successfully receive the message store the chain (unverified chain) for future reference. When another user U_2 queries his own vehicle for a place to refuel the system returns a list of places recommended by other users (among which is U_1's recommendation). If U_2 decides to go to *A* he will afterwards input his review into the system, which will cause the unverified chain to turn into verified, and if his review is similar to U_1's his level of trust in U_1 will increase, or decrease otherwise.

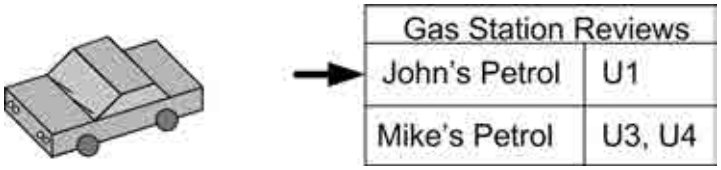
(a) Unverified POI chains organization

(b) Verified POI chains organization

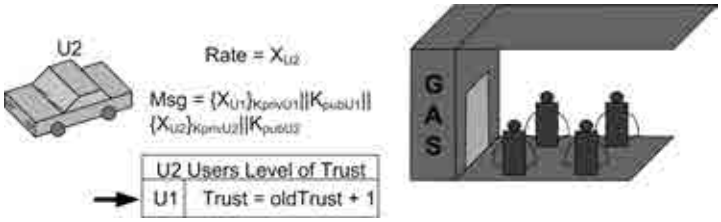
Figure 1: POI chains organization.



(a) U_1 reviews *Mike's Petrol* and broadcasts the message.



(b) U_2 queries his vehicle for a gas station.



(c) U_2 follows U_1 recommendation, adds his own opinion to the chain and updates his level of trust in U_1 according to how similar both reviews were.

Figure 2: General behavior of the *Chains of Trust* protocol.

Regardless of how much he coincides with U_1's opinion, U_2 will append his own signed review to the original (thus increasing the length of the chain of recommendations), together with his K_{pub}, and broadcast the message. In this way, every time a user follows and verifies a recommendation he can update his level of trust in *n* other nodes (where *n* is the length of the chain of signatures-recommendations), thus increasing the speed at which the reputation system develops. In addition, in order to discourage misbehavior, it is easier for a user to lose his reputation because of a series of misbehaviors than to increase it by issuing faithful reviews. Otherwise, a well behaved user could use his reputation to do as much damage to the network as good he had done previously.

Every user will store in a list the identity of the nodes he trusts and what level of trust he has in them. In this way, nodes will be divided between unknown, trusted and most trusted. The most trusted nodes is a group of *m* nodes with the highest reputation. Every user's *Most Trusted Group* (MTG) assists him in different decision making processes, e.g., if a user queries the system for a place to have dinner the first recommendations will belong to the MTG, followed by the rest of trusted nodes and closing with recommendations made by unknown nodes. This makes it more difficult for an attacker to influence any user's behavior since he first needs to gain access to the MTG, and once he misbehaves he will immediately be expelled from it.

As far as communications are concerned, CoT considers three different mechanisms for the exchange of information between the nodes in the network: (i) whenever a user needs information on a certain POI category, his vehicle queries its neighbors, (ii) vehicles periodically exchange POI chains (verified and unverified) with the better rated POIs in each category (every 120 seconds) and (iii) if during a periodic exchange, one vehicle is recognized as a trusted user (from a previous encounter) then recognizer and recognized will exchange information about their levels of trust in other users, i.e., user chains.

CoT completely relies on an ad-hoc network, and therefore requires no road-side infrastructure. This helps protect user privacy, since the reputation system knowledge is distributed between all users. Nonetheless, an attacker could position himself at very frequented crossroads and given sufficient time he could gather a large amount of data containing thousands of recommendations from thousands of users. This information could be used to profile users, deduce their habits and try to link their public key with their real identity, all of which would compromise the users' privacy.

3. Anonymous Chains of Trust

3.1. General overview

In this article we propose a new mechanism based on CoT to preserve users privacy based on identity borrowing. In a reputation system, if two nodes trust each other it is because they both have had similar views or opinions on the information they have shared in the past. Particularly in CoT, if two users trust each other it is because they have similarly rated POIs in a given category and therefore have similar tastes. Since their rates for a certain POI category are similar, one user *A* should be able to ask another user *B*, who he trusts, to issue a review message with a certain rate for a certain POI with *B*'s own identity, much like if he had reviewed the POI himself. For all intents and purposes, *A* will be borrowing *B*'s identity for that single review.

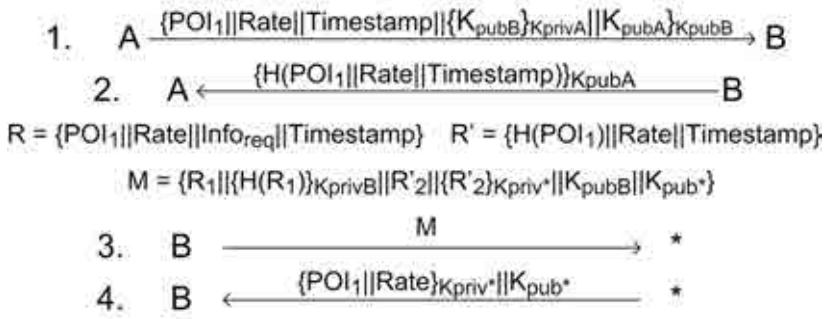


Figure 3: *Anonymous Chains of Trust.*

Fig. 3 depicts in detail how the system works. Steps 1 and 2 are the first part of the protocol where user A requests user B to issue a POI review on his behalf. The second part, steps 3 and 4, allows B to determine how reviewing that POI on behalf of A affects his reputation.

User *A* reviews POI_1 and the system decides to request another node to issue the public review on his behalf. How often the system asks a user to review a POI on behalf of another is based on the system parameter α. Whenever *A* meets one of his trusted nodes (*B*) he sends a review request containing the POI identifier (POI_1), the rate he assigns to that POI and a timestamp, everything encrypted with *B*'s public key (as we can see in step 1). In addition, he signs B's K_{pub} with his own K_{priv} so that B can verify A's identity. In step 2, if *B* recognizes *A* as one of his trusted users he acknowledges the reception of the message by sending the hash *H* of the received message encrypted with *A*'s public key K_{pub_A}. Should that acknowledgement

not reach *A* the system on *A*'s vehicle will request the review to another trusted user. Once *B* has accepted to review POI_1 on *A*'s behalf, he will include this review in the list of messages he transmits periodically.

In step 3, *B* prepares a periodic message *M* containing a chain of reviews of length 2 for POI_1, which includes the review R_1 he is issuing on behalf of *A* and another review R_2 he has received for that same POI from another user. For the sake of clarity, in this example *M* contains information about just one POI, i.e., one POI chain, although in reality periodic messages may include several concatenated chains for different POIs. Once that information is compiled, *B*'s vehicle broadcasts it to the network. It should be noted that *R*, which is the first element of a chain of POI recommendations, contains a field named $Info_{req}$. This bit-field will be set to let the message receivers know that *B* would like to receive their reviews of that POI. In step 4, the receivers of *M* reply with their own rate for the requested POI. *B* will store this information, and once he has gathered enough data he will evaluate the review *A* sent to him and adjust his level of trust in *A* accordingly (as explained in section 3.2).

It should be noted that nodes that receive the information request will reply with their own reviews, with reviews from trusted nodes or with reviews they have issued on behalf of other nodes. If they were only allowed to reply with their own reviews, an attacker would only need to broadcast a POI request for multiple POIs and gather all the information to profile the users.

Incidentally, in order to minimize the repetition of information in *M*, the POI identifier is only used in the first review of a chain of recommendations (*R*), while the rest use instead the hash *H* of that identifier (R').

The idea behind this scheme is that if enough users request their trusted fellows to review POIs on their behalf, then a user's individual identity is hidden by the identities of all the users he trusts. As a result, even an all-knowing attacker will not be able to profile individual users because he will have no way of knowing the identity of the real POI reviewers. This concept of privacy is somewhat similar to what group signatures provide [12, 13, 14], although without the overhead of specifically creating and managing a group.

Generally speaking, in a group signature scheme every user is part of a group, either preset or dynamically created, and every group has a group manager in charge of making public the information gathered by group members. In addition, the group manager needs to monitor the group members for misbehavior and evict them from the group if they misbehave.

3.2. Evaluation of identity borrowing

As seen in section 3.1, user B needs a mechanism to determine the impact that reviewing a POI on behalf of A has on his reputation. Whenever a user reviews a POI on behalf of somebody else he sets the $Info_{req}$ bit in the chain of reviews for that POI in the periodic message. After having gathered n reviews from other users (or if the time passed since he issued the review reaches a certain value $T_{evaluation}$) B evaluates A's review.

Let us define n as the number of reviews sent by different users regarding a certain POI POI_1, $U_1, ..., U_n$ as the users who sent their POI review and $\hat{U}_1, ..., \hat{U}_n$ as the subset of those nodes known by the user B, χ_{POI_1,U_1} as the rate that U_1 gave to POI_1 and $\lambda_{\hat{U}_i}$ as the level of trust that B has on $\hat{U}_i$ as a POI reviewer. Then the POI consensual grade G is defined by:

$$G = \sum_{i=1}^{n} \left(\chi_{POI_1,\hat{U}_i} \cdot \frac{\lambda_{\hat{U}_i}}{\sum_{j=1}^{n} \lambda_{\hat{U}_j}} \right) \tag{1}$$

It should be noted that the rates assigned by unknown nodes are ignored as long as there is a known reviewer in the chain. Otherwise, the chain's rate is the arithmetic mean of the POI rates assigned by the unknown reviewers. Similarly, the reviews of the less trusted known nodes are ignored when there is a known node that belongs to the group of B's MTG.

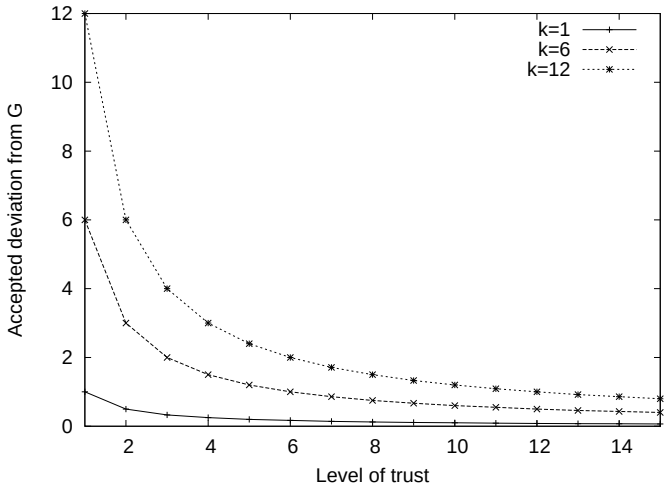


Figure 4: *Progression of k/λ_A for different values of k.*

Once B knows the value of G, he expects the rate A sent in his review of POI_1 to be:

$$G - k/\lambda_A \leq \chi_{POI_1,A} \leq G + k/\lambda_A \tag{2}$$

where k is a parameter defined by each user depending on how strict he wants to be when lending his reputation. k can take any value considering that $G + k/\lambda_A \leq 15$, 15 being the maximum value for a node's reputation in the system as well as the maximum rate for a POI, and $G - k/\lambda_A \geq 0$, 0 being the minimum. If $\chi_{POI_1,A}$ falls outside the limits defined by (2) then B will stop transmitting A's review and the level of trust B has on A, i.e. λ_A, will decrease by half its value.

It should be noted that too high values of k will allow misbehaving users to take advantage of the system and ruin the reviewer's reputation in the network. On the other hand, too low values will in all likelihood unfairly decrease the level of trust B has in A. Regardless of the value assigned to k, in Fig. 4 we can see that the allowed deviation from G decreases for high levels of trust between users. This responds to the fact that users with high levels of trust assign the most similar rates to the same POIs, and that should still be true when a user is lending his identity.

In the same way that B needs to make sure that A is not lying to him, A needs to know if B is really transmitting a review on his behalf. To that end A examines the periodic messages he receives looking for a chain of recommendations for the requested POI POI_1. If he does not find it after a certain time $T_{request}$, A will request the review of POI_1 to another of his trusted nodes. The level of trust that A has in B does not need to be decreased because A's reputation in the network was not damaged by B's inaction.

3.3. Scalability analysis

In [3] we determined with a ns-3 [15] simulation that in a 400 vehicles scenario such as the one depicted in Fig. 5, every user can broadcast 400 packets of a 1,000 bytes every 120 seconds yielding a 91.5% rate of successfully received packets. It should be noted that in our system every node broadcasts periodic messages to be received by all nodes within 1 hop distance.

The periodic message used in CoT has been modified to include the changes described in section 3.1 with the goal of achieving a reception rate still over 90%. Considering the following format for a periodic message M as defined in Fig. 3:

$$R = \{\underbrace{POI_{Id}}_{88\text{ bytes}} \| \underbrace{Rate}_{1\text{ byte}} \| \underbrace{Info_{req}}_{1\text{ bit}} \| \underbrace{Timestamp}_{8\text{ bytes}}\} \tag{3}$$

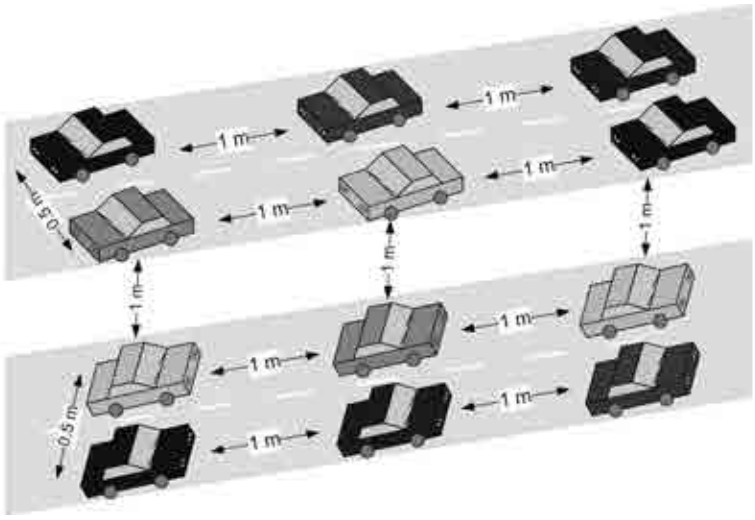


Figure 5: Vehicle layout for the 400 nodes scalability scenario.

$$R' = \{\underbrace{H(POI_{Id})}_{8\text{ bytes}} \| \underbrace{Rate}_{1\text{ byte}} \| \underbrace{Timestamp}_{8\text{ bytes}}\} \tag{4}$$

$$M = \{\underbrace{R_1}_{97\text{ bytes}} \| \underbrace{\{H(R_1)\}_{K_{priv_{Node\ 1}}}}_{17\text{ bytes}} \| \underbrace{\{R'_2\}_{K_{priv_{Node\ 2}}}}_{17\text{ bytes}}$$
$$\|...\| \underbrace{\{R'_n\}_{K_{priv_{Node\ n}}}}_{17\text{ bytes}} \| \underbrace{K_{pub_{Node\ 1}}}_{128\text{ bytes}} \|...\|\| \underbrace{K_{pub_{Node\ n}}}_{128\text{ bytes}}\} \tag{5}$$

Taking into account that the total amount of information has to be approximately 400,000 bytes, information about 25 POIs will be sent, each containing 107 user's reviews adding up to a total of 390,303.125 bytes. It should be noted that periodic messages are fragmented in a 1,000 bytes packets including certain redundancy, so that if a packet is lost the rest of the message can still be read.

In addition, in order to avoid flooding the network when users reply to a POI information request, it will only be allowed to set the $Info_{req}$ bit for a maximum of 5 POIs in a message M.

$$POI_{resp} = \{\underbrace{\{POI_{Id} \| Rate \| Timestamp\}_{K_{priv_S}}}_{97\text{ bytes}} \| \underbrace{K_{pub_S}}_{128\text{ bytes}}\} \tag{6}$$

In the best case scenario every user will have information of all 5 POIs and reply with POI_{resp} (a 1,125 bytes message).

4. Experiments

Once the system has been defined we need to determine how it will perform in a realistic scenario. To that end, we re-designed our simulation tool *poiSim* [3] to simulate *Anonymous Chains of Trust*.

poiSim is a vehicular application simulator which, like in [16], analyzes the realistic vehicular trace produced by the *Multi-Agent Traffic Simulator* (MMTS) developed by K.Nagel at ETH Zurich. The MMTS is capable of simulating public and private traffic over real regional road maps of Switzerland with a high level of realism. It models the behavior of people living in the area, reproducing their movement (using vehicles) within a period of 24 hours. The decision of each individual depends on the area it lives in. The individuals in the simulation are distributed over the cities and villages according to statistical data gathered by a census. Within the 24 hours of simulation, all individuals choose a time to travel and the mean of transportation according to their needs and environment, e.g., one individual might take a car and go to work in the early morning, another one wakes up later and goes shopping using public transportation, etc. All in all, with over 260,000 simulated nodes or vehicles in an area of around 250 km x 260 km, this mobility trace suited the simulation needs.

In order to better study the system, to observe how the POI reviews are exchanged between users, how users build a better reputation for themselves and the effect of different values of α, the 24 hours vehicular trace is replayed to obtain a multiple days scenario. It should be remarked that the only common element in every simulated day will be the MMTS trace, because the POIs being reviewed are randomized, and hence will be different in every run.

In *Anonymous Chains of Trust* whenever a user reviews a POI the system needs to choose between: (i) broadcasting that review, i.e., making it public, and (ii) waiting until the user's vehicle recognizes a trusted node and asking him to review that POI on his behalf. As explained in section 3.1, this decision depends on the system parameter α. In the early stages of the reputation system deployment, that delay can hamper the development of the *Web of Trust* between users. Determining the degree to which the system deployment is affected is our main goal.

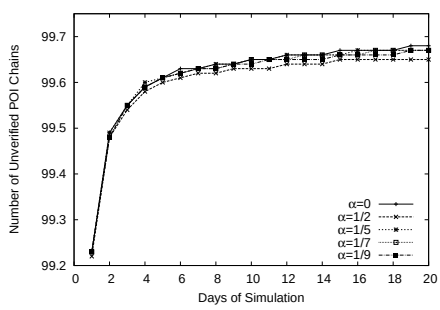
In this experiment, every user inputs a new review into the system every 5 days and we study different values for α: a user requests another user to review a POI on his behalf once every 2 reviews ($\alpha = 1/2$), 1 review of every 5 ($\alpha = 1/5$), 1 review of every 7 ($\alpha = 1/7$), 1 review of every 9 ($\alpha = 1/9$) and a control sample where users do not review POIs on behalf of other

users ($\alpha = 0$). The reviews or rates users assign to POIs range from 0 to 15 and follow a normal distribution with mean 7 and $\sigma = 2$. The evaluation of user misbehavior is outside the scope of this simulation. It should be noted that the real measure of the system performance is given by how many users every user knows and how much he trusts them, because (i) the more users he knows the more information he has to choose a truthful recommendation from and (ii) the more users he knows the more users he can ask to review a POI on his behalf and make his identity harder to discover.

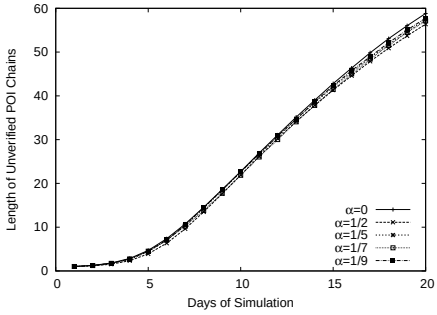
In Figs. 6a and 6b the evolution of the number and length of unverified POI chains can be seen. After the first 5 days of simulation the number of unverified chains and its length is very similar regardless of the reviewing rate. The fact that the average number of unverified chains is over 90 (the simulator can store up to 100) and its length is approximately 5 (considering any of the α's) means that there has been interaction between the users and some have already started to build a better reputation in the network. Moreover, the results after 20 days of simulation do not differ significantly.

As far as verified chains are concerned, in Fig. 6c the direct relation between the reviewing rate and the number of verified chains the nodes store can be observed. After 20 days of simulation, the difference between a $\alpha = 1/5$ and the control group with $\alpha = 0$ is almost 1, increasing to almost 2.5 for $\alpha = 1/2$. Overall, the more often a user request another user to review a POI on his behalf the lower his number of verified chains will be, which is logical considering that higher request frequencies introduce a greater delay to information transmission. Fig. 6d shows the mean of the length of verified POI chains. It is very similar to 6b, which is natural considering that every time a POI is reviewed its unverified chain moves on to the verified state. Regarding the rate assigned to the POIs in the verified chains, in Fig. 6e we can observe that the rate of the reviewed POIs varies until it stabilizes around 7, which is expected since the randomly chosen rates are distributed around that value, as previously described in this section. The different simulated values for α determine how fast the POI rate converges to 7.

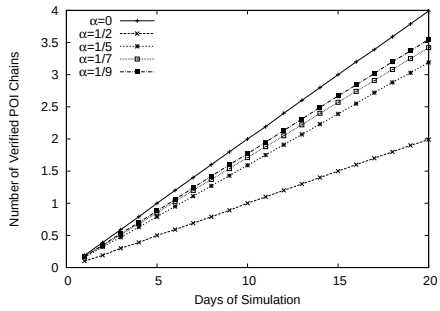
Figs 7a and 7b present the user reputation results. In 7a we can see that after 20 days of simulation, nodes in the control group ($\alpha = 0$) know on average 160 users, while nodes with $\alpha = 1/5$ know approximately 130 users and nodes with $\alpha = 1/2$ know slightly under 100. Regarding the level of trust in those users depicted in Fig. 7b, we can say that they are very similar regardless of the value of α, the maximum difference shown by $\alpha = 0$ and $\alpha = 1/2$.



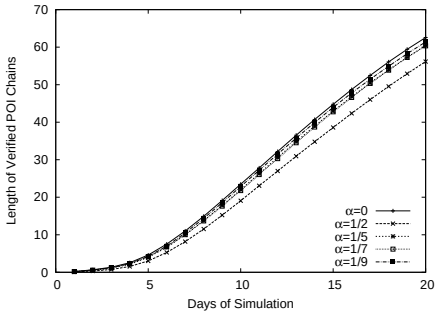
(a) Number of unverified POI chains: for every node the number of unverified POI chains is computed, their mean is the depicted result.



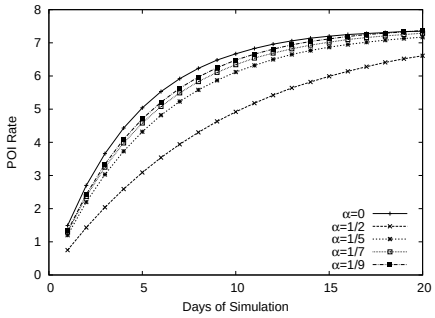
(b) Length of unverified POI chains: for every node the mean of its unverified POI chains length is computed, the mean of those means is the depicted result.



(c) Number of verified POI chains: for every node the number of verified POI chains is computed, their mean is the depicted result.

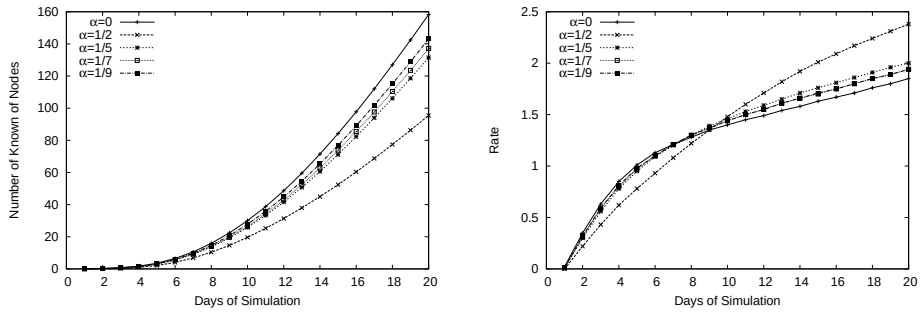


(d) Length of verified POI chains: for every node the mean of its verified POI chains length is computed, the mean of those means is the depicted result.



(e) Rate in the verified POI chains: the mean of the rates users assign to POIs.

Figure 6: Evolution of the length and number of unverified and verified chains.



(a) Number of known nodes: mean of the number of known nodes by every node.

(b) Rate or level of trust of the known nodes: mean of the rates users assign to other users as POI reviewers.

Figure 7: Number of known nodes and their levels of trust progress.

5. Conclusions

In this article we have presented a mechanism to preserve users privacy in a reputation system. By allowing users to borrow each other's identities an attacker can never be sure of who was the real reviewer behind a given POI recommendation. In other words, users that trust each other form a virtual group where any user can use anybody else's identity, thus hiding behind the group. Moreover, this technique should be transparent to the user reputation, since identity borrowing can only occur between users that trust each other, which by definition implies that their reviews for a given POI category are very similar and therefore interchangeable.

The results of our simulation tell us that regardless of the value of α we have used (how often a user reviews POI on behalf of another) the length of unverified and verified chains and their rates remains very similar. Regarding the number of users known by every node and his level of trust in them we have shown that even if the known number of users is slightly lower for $\alpha = 1/5$ the difference when compared to the control group is not significant and does not constraint the development of the reputation system. When we compare the control group with $\alpha = 0$ we can start to see a decrease in the system performance (it has a fewer number of verified chains and knows less nodes).

Privacy wise, the fact that after just 10 days of simulation every user knows about 20 other users which he trusts with a rate of approximately 1.5 tells us that an attacker trying to profile a user will have to guess which of the 20 trusted nodes he relies on really issued the review. This problem becomes increasingly harder as the days go by. For instance, after 20 days of simulation

an attacker would have to find the real reviewer from a group of approximately 120 users.

All in all, the results show that reviewing POIs on behalf of other users with a moderate frequency has hardly an impact on the system performance while their privacy is protected. However, in a scenario where users review as many POIs on behalf of others as they do for themselves the results point to the fact that borrowing identities to preserve user privacy poses a constraint on how fast the reputation system develops.

Acknowledgment

This work was partially supported by the EuroNF NoE and by Spanish grants TIN2010-21378-C02-01 and 2009-SGR-1167.

References

[1] A. Jøsang, J. Golbeck, Challenges for robust trust and reputation systems., in: Proceedings of the 5th International Workshop on Security and Trust Management (STM 2009), 2009.

[2] K. Hoffman, D. Zage, C. Nita-Rotaru, A survey of attack and defense techniques for reputation systems, ACM Comput. Surv. 42 (2009) 1:1–1:31. doi:http://doi.acm.org/10.1145/1592451.1592452.

[3] D. A. Rivas, M. Guerrero-Zapata, Chains of trust in vehicular networks: A secure points of interest dissemination strategy, Ad Hoc Networks 10 (6) (2012) 1115 – 1133. doi:10.1016/j.adhoc.2012.02.011.

[4] Q. Ding, X. Li, M. Jiang, X. Zhou, Reputation management in vehicular ad hoc networks, in: Multimedia Technology (ICMT), 2010 International Conference on, 2010, pp. 1–5. doi:10.1109/ICMULT.2010.5632149.

[5] A. Tajeddine, A. Kayssi, A. Chehab, A privacy-preserving trust model for vanets, in: Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, 2010, pp. 832–837. doi:10.1109/CIT.2010.157.

[6] A. Patwardhan, A. Joshi, T. Finin, Y. Yesha, A data intensive reputation management scheme for vehicular ad hoc networks, in: Mobile and Ubiquitous Systems: Networking Services, 2006 Third Annual International Conference on, 2006, pp. 1–8. doi:10.1109/MOBIQ.2006.340422.

[7] G. Montenegro, C. Castelluccia, Statistically unique and cryptographically verifiable (sucv) identifiers and addresses, in: In Proceedings of the 9th Annual Network and Distributed System Security Symposium (NDSS, 2002.

[8] S. Dhurandher, M. Obaidat, A. Jaiswal, A. Tiwari, A. Tyagi, Securing vehicular networks: A reputation and plausibility checks-based approach, in: GLOBECOM Workshops (GC Wkshps), 2010 IEEE, 2010, pp. 1550–1554. doi:10.1109/GLOCOMW.2010.5700199.

[9] N.-W. Lo, H.-C. Tsai, A reputation system for traffic safety event on vehicular ad hoc networks, EURASIP J. Wirel. Commun. Netw. 2009 (2009) 9:1–9:2. doi:http://dx.doi.org/10.1155/2009/125348.

[10] S. Park, B. Aslam, C. Zou, Long-term reputation system for vehicular networking based on vehicle's daily commute routine, in: Consumer Communications and Networking Conference (CCNC), 2011 IEEE, 2011, pp. 436–441. doi:10.1109/CCNC.2011.5766507.

[11] K. Fall, A delay-tolerant network architecture for challenged internets, in: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '03, ACM, New York, NY, USA, 2003, pp. 27–34. doi:10.1145/863955.863960.

[12] M. Raya, A. Aziz, J.-P. Hubaux, Efficient secure aggregation in vanets, in: VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks, ACM, New York, NY, USA, 2006, pp. 67–75.

[13] X. Lin, X. Sun, P.-H. Ho, X. Shen, Gsis: A secure and privacy-preserving protocol for vehicular communications, Vehicular Technology, IEEE Transactions on 56 (6) (2007) 3442–3456.

[14] E. van Heyst D. Chaum, Group signatures, Advances in Cryptology 1981 - 1997 - EUROCRYPT '91, Springer-Heidelberg 1440 (1999) 127–133.

[15] T. R. Henderson, S. Roy, S. Floyd, G. F. Riley, ns-3 project goals, in: Proceeding from the 2006 workshop on ns-2: the IP network simulator, WNS2 '06, ACM, New York, NY, USA, 2006. doi:http://doi.acm.org/10.1145/1190455.1190468.

[16] V. Naumov, R. Baumann, T. Gross, An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces, in: MobiHoc '06: Proceedings of the 7th ACM international symposium on Mo-

bile ad hoc networking and computing, MobiHoc '06, ACM, New York, NY, USA, 2006, pp. 108–119. doi:http://doi.acm.org/10.1145/1132905.1132918.

Visual Light Communication in Vehicular Ad-hoc Networks

David Antolino Rivas*, Manel Guerrero Zapata

Department of Computer Architecture, Polytechnic University of Catalonia, Barcelona 08034, Spain
Telephone: (+34) 934054059

Abstract

Vehicular Ad-hoc Networks (VANETs) portray a world where every vehicle is equipped with the means to communicate with each other, thus providing the perfect framework for the development of applications that improve our driving experience, e.g., safety, content distribution, liability in case of an accident, etc. The vast majority of these applications rely on the IEEE 802.11p/WAVE standard (Wireless Access in Vehicular Environments) or some other form of radio communication. This poses a security risk if we consider how vulnerable radio transmission is to intentional jamming and natural interferences since an attacker could easily block all radio communication in a certain area if its transmitter is powerful enough.

Visual Light Communication (VLC), however, is resilient to jamming over a wide area because it relies on visible light to transmit information. Therefore, VLC could be a perfect complement to radio communication whenever the signal to noise ratio was too low. VLC research is currently gaining momentum and although experiments have only been conducted indoors, experts recognize the potential that this technology has in the open air.

This article is the first to realistically consider VLC as a form of communication for VANETs, both as an alternative to radio waves and as an addition to it. Whenever the wireless physical medium becomes too populated or in case of an emergency VLC could be used to transmit information. In order to show its feasibility, several experiments comparing the performance of VLC and WAVE have been performed. In these experiments our simulation tool processes a 24 hours mobility trace with over 260,000 nodes produced by a *Multi-Agent Traffic Simulator*, which realistically simulates public and private traffic over regional maps of Switzerland. The results show that VLC performs satisfactorily in a realistic scenario.

Keywords: VLC, Visual, Light, Communication, VANET, Vehicle

1. Introduction

In the near future, *Vehicular Ad-hoc Networks* (VANETs) will change the way we drive. Vehicles will be able to communicate among them and with road-side infrastructure which will connect them to other networks,e.g., the Internet. Many applications for VANETs have been proposed: content distribution [1, 2], advertisements [3], finding a parking spot [4], safety and emergency applications [5], etc.

It is important to notice that most of these applications have one thing in common: the use of radio communication. For that reason, the standard WAVE-DSRC, which defines wireless vehicular communication, was designed. According to [6], WAVE-DSRC has the mechanisms to provide different user applications with different communication channels while reserving certain channels for safety applications, others for protocol operation and others for public safety. WAVE-DSRC can transmit in short range (i.e., less than 100m) at data rates between 1-54Mbps and at a range of less than 1000m at data rates between 3-27Mbps. Radio communication, however, is inherently vulnerable to jamming attacks: anyone with a powerful enough radio device can transmit in the same channel used by vehicles and distort communication over a wide area (the radius of which depends on the power of the radio device), thus causing a *Denial of Service* (DoS). The impact of such an attack ranges from a minor inconvenience for content distribution applications to a potential car accident for safety applications.

Recent research has begun to focus on *Visual Light Communication* (VLC) [7, 8] as an alternative form of communication. In VLC, the communication takes places between a *Light Emitting Diode* (LED) used as a transmitter and photodiode that acts as a receiver. In the past few years, there has been significant progress in this area, e.g., in [7] the authors were able to reach a transmission speed of a 100Mbps in indoor conditions. Extensive research still needs to be conducted before the technology becomes available to the general public. Efforts in that direction are backed by the recently created IEEE 802.15.7 Visible Light Communication Task Group [9] and the Visible Light Communications Consortium [10].

LED illumination is becoming widespread for indoor lightning due to its lower power consumption compared to the regular light bulb. In addition, it is also becoming increasingly popular in the automotive industry for indicator, tail and even headlights, as well as being used in traf-

*Corresponding author.
Email address: antolino@ac.upc.edu (David Antolino Rivas)

fic lights and signs. By the time VLC technology is mature enough to be used outdoors, LED illumination will be widespread and a great range of possibilities will open for VANETs.

In this article, we define a 24 hour scenario where a 260,000 vehicles VANET uses VLC (as explained in section 3) and simulate it with a modified version of our own simulation tool [11, 12]. The vehicles mobility is determined by the trace produced by the realistic *Multi-Agent Traffic Simulator* (MMTS) developed by K.Nagel at ETH Zurich [13]. We will compare the performance of VLC and WAVE (considering their different ranges of transmission) under certain situations in order to determine if VLC could be a valid form of communication in VANETs.

The remainder of this work is organized as follows. In sections 2.1 and 2.2 a background in VANETs and VLC is given respectively. Section 3 describes in detail what is our proposal for VANET communication and subsections 3.1, 3.2, 3.3 present the simulation results. Finally, the article closes with the conclusions drawn from those results.

2. Related work

2.1. Vehicular Ad-hoc Networks

Vehicles equipped with wireless communication devices, also known as *On Board Units* (OBUs), will be able to communicate among themselves and with *Road Side Units* (RSUs). RSUs will compose the backbone of the roadside infrastructure which will connect the vehicular network to a central system or to the Internet.

With the massive deployment of wireless technologies, the automotive industry will open a wide range of possibilities for drivers and passengers alike: theoretically, anything from finding out the road conditions ahead to watching a movie through streaming should be possible. So different requirements will lead to the deployment of different kinds of applications over the network. In [14] and [15] applications are classified based on the service they provide:

1. **Safety related applications:**
   (a) *Traffic information messages*: used to disseminate traffic conditions over an area; they affect public safety only indirectly (they are not time-critical).
   (b) *General safety-related messages*: used by public safety applications such as cooperative driving and collision avoidance (in order to prevent traffic accidents time is certainly an issue; at least they should satisfy an upper bound delay in delivering the information).

(c) *Liability-related messages*: they are only exchanged in liability-related situations such as accidents. The senders' identities should be kept hidden from the other users in the network and only revealed to the law authorities (time is not an issue).

2. **Other applications (some examples):**
   (a) *Toll applications*: electronic toll collection systems like *AutoPASS* in Norway allow drivers to continue driving without having to stop at tolls.
   (b) *TV and other multimedia content*: used to provide users with entertainment and information (movies, newspapers, etc.).
   (c) *Advertisements*: businesses along the road (such as gas-stations and restaurants) could advertise themselves to drivers before they reached their location, giving them enough time to compare different offers.

In [16] the authors present a scheme to distribute traffic events information. They define a two tier approach: vehicle sensors first have to detect an event a certain number of times T_S before reporting it to the driver and if they have not detected the event for themselves, they need to receive the event warning from T_V vehicles before trusting it. Every time an event is detected T_S times a message including how many times the vehicle's sensors have detected the event and the identity of vehicles detecting it is send to the vehicle's neighbors. The receiving nodes will use this value and the number of vehicles that detected the event to determine if it is true or not.

As far as safety applications are concerned, the integrity and the non-repudiation of the transmitted messages has to be ensured, albeit maintaining at the same time the user's privacy. For instance, a traffic information application needs to make every user accountable for the traffic events he reports, otherwise a misbehaving user would be able to report false events (e.g., traffic jams, accidents, etc.) and redirect traffic to his own benefit. Other applications, e.g., multimedia content distribution, may also need to encrypt their messages to avoid eavesdropping from non-registered users. The use of *Public Key Infrastructure* (PKI) will fulfill most security requirements.

In [2] the authors present Roadcast, a popularity aware P2P content sharing scheme. Their technique relies on the idea that by ensuring that popular data is widely shared with other vehicles the overall query delay can be improved. If users request popular data, which is densely disseminated in the network, their queries can be answered in much shorter time than a request for rare data, because the chance of meeting another vehicle with that particular piece of infor-

mation is much higher. In the opportunistic and unreliable VANET, the authors expect users to be more willing to receive data which approximately matches their request with a short delay than waiting for a longer time to receive exactly what they requested. Thus the need to forward the popular information with higher priority.

In [11] we introduced *Chains of Trust*, a secure *Points of Interest* (POI) distribution strategy and reputation system for VANETs. In a nutshell, users issue reviews of POIs and broadcasts them to the network. The receiving users store them for future use so that when they need information about a certain POI category they can choose one recommendation issued by another node (preferably one they already trust). Whenever they follow one of these recommendations, they issue their own review of the POI and the system updates the level of trust on the recommender(s) depending on how similar their reviews were. In addition, users who trust each other not only exchange information about POIs, but also about other users, i.e., which ones are the most trustable.

Regarding advertisements distribution, in [3] the authors describe a protocol based on a virtual cash scheme where the following actors are involved:

- *Certification Authority* (CA): every vehicle is loaded with a pair of keys (public and private) issued by a CA and with the CA's public key.
- *Vehicular Authority*: entity that approves every advertisement to be loaded in an *Ad Distribution Point*.
- *Ad Distribution Point*: broadcasts advertisements to the vehicles passing by.
- *Virtual Cashiers*: users are rewarded with virtual cash for forwarding advertisements. They sign each other receipts to prove the message forwarding. Later on, that cash can be exchanged for other services at the *Cashiers*.
- *Road Side Units* (RSU): provide a link to the CA for keys revocation purposes.

It should be noted that even though VANET applications may differ on their goal or their design, almost all of them (if not all) rely on the use of some sort of radio communication.

2.2. Visual Light Communication

The predecessor of modern VLC was the photophone invented by Alexander Graham Bell and Charles Sumner Tainter [17]. The device consisted of a transmitter which modulated a

light beam with a person's voice and a parabolic receiver on the other end which converted the light back into sound. The transmitter used a mirror which vibrated with voice, thus alternating between convex and concave forms and dispersing and focusing the light. The receiver had selenium cells at its focal point, which made possible to convert the light back into voice due to its photovoltaic properties (its electrical resistance is higher when in the dark and lower when exposed to light). The invention was successfully tested over a distance of approximately 213m using plain sunlight as their light source.

VLC uses visible light, with a wavelength between ~400nm (750THz) and ~700nm (428THz), to transmit information. It is based on the usage of a white LED emitter and a photodiode as a receiver.

The authors in [8], classify white LEDs into two types: (i) devices that use separate red-green-blue (RGB) emitters and (ii) blue emitters used in combination with a phosphor that emits yellow light. The former has a greater bandwidth while the latter has lower complexity.

As far as data rate is concerned, in [8] the authors present their results building a VLC link between an emitter and a receiver using a pre-equalized 45MHz bandwidth white LED, reaching a speed of 80Mbps with *On-Off Keying Nonreturn-to-zero* (NRZ-OOK) over a link of 10cm (a distance which could be extended by using an array of transmitters, according to the authors). Similarly, in [7] the authors present their experiment using post-equalization, which reaches 100Mbps over a 10cm link, although the range could also be extended by using an array of transmitters.

The Visible Light Communications Consortium shows in [18] a wide variety of applications for VLC:

- a prototype which transmits sounds through RGB lights, where each RGB light has the sound of a different instrument: guitar, keyboard, etc.
- usage in restricted areas like aircrafts and hospitals.
- in a supermarket, product information could be acquired by the visible light receiver installed on the shopping cart
- indoor navigation systems
- wireless LANs

As the technology matures it will be possible to extend optical wireless networks to the outdoors. For instance, in [19, 20, 21, 22] the authors use lasers to transmit information and, in particular, to solve a problem commonly referred to as the *first/last mile problem* [19, 22]. In the early days of optical fiber deployment, the fiber connected a telecommunication company's different switching stations while consumers connected to those stations through twisted-pair wiring, which in effect limited the network access rates. Optical wireless proposed to bridge this gap and connect consumers directly to their closest switching station with a laser link, thus improving data rates and minimizing deployment costs.

In our view, in the next decade we will see vehicles transmitting information with their headlights or receiving information from traffic signs, as envisioned in [18, 23, 24]. However, there are several aspects that need to be addressed first, like the low transmission speed over a long link (speed rapidly decreases as the distance increases, from 100Mbps in a 2 meters link to 115Kbps for approximately 5 meters [18]) and how to transmit in movement. In addition, in order to succeed in the open air it must overcome the interferences caused by meteorological conditions (e.g., fog, rain, etc.).

On the plus side, VLC has important advantages over radio communications such as: practically unlimited bandwidth (unlike the hyper-regulated radio spectrum), a relatively low power consumption and resilience against jamming and DoS attacks.

3. Visual Light Communication in VANETs

The main goal of this article is to determine whether VLC could be an effective way to transmit information in a VANET (either on its own or in collaboration with WAVE-DSRC). However, the fact that the technology is not yet fully developed has to be taken into consideration. In addition, current research is focused on indoor applications because of its lower complexity. As a result, our experiments will only focus on the transmission range and we will consider 5m to be the maximum VLC range, because beyond that distance the data rate decreases dramatically. Notice how 5m should be enough to allow a vehicle to at least communicate with its immediate neighbors.

In the simulated scenario every vehicle is equipped with a set of VLC emitters and receivers distributed as depicted in Fig. 1. Even though the emitter's transmission cone is yet to be defined by manufacturers, we do know that LEDs are relatively inexpensive, which allows us to

install several emitter-receiver sets in array to maximize the chance of a successful transmission regardless of the vehicles' position.

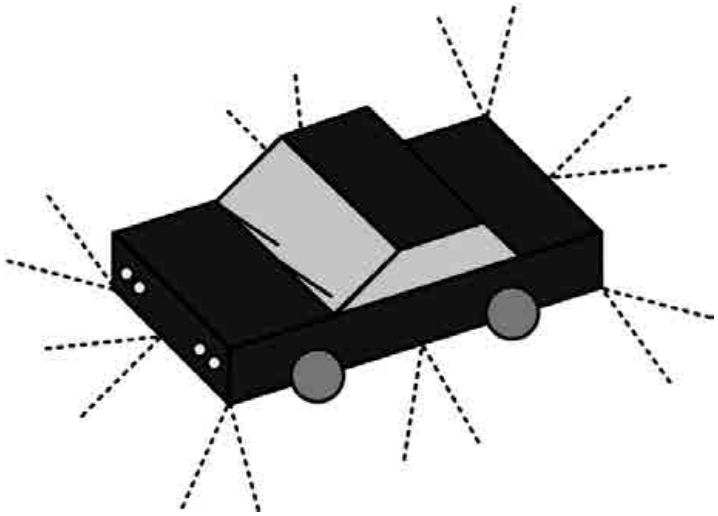


Figure 1: Emitter-receiver sets positioned in a vehicle and their transmission cone.

In order to determine how VLC would perform in a real VANET we need a realistic simulation tool. Simulation tools like Glomosim or ns-2 were discarded because in order to simulate hundreds of thousands of nodes they require a massive amount of memory. Thus, we were inclined to use a modified design of our own simulation tool [11, 12]. Like in [25], it was decided to analyze the realistic vehicular trace produced by the *Multi-Agent Traffic Simulator* (MMTS) developed by K.Nagel at ETH Zurich. The MMTS is capable of simulating public and private traffic over real regional road maps of Switzerland with a high level of realism. It models the behavior of people living in the area, reproducing their movement (using vehicles) within a period of 24 hours. The decision of each individual depends on the area it lives in. The individuals in the simulation are distributed over the cities and villages according to statistical data gathered by a census. Within the 24 hours of simulation, all individuals choose a time to travel and the mean of transportation according to their needs and environment, e.g., one individual might take a car and go to work in the early morning, another one wakes up later and goes shopping using public transportation, etc. All in all, with over 260.000 simulated nodes or vehicles in an area of around 250 km x 260 km, this mobility trace suited our simulation needs.

The mobility trace roughly consists in a x, y, z position update for every node every t seconds

(different periods t for every node). It has 3 different types of updates: node starts a trip, node updates its position and node finishes a trip. Every time the trace provides an update on a vehicle's position, the simulation tool computes a rectilinear trajectory between the previous x, y, z and the new x', y', z' coordinates for the updated node, as depicted in Fig. 2. Then, its trajectory is compared to the trajectory of every active node (every vehicle currently on the road) and it determines if their paths cross and should that be the case if the crossing point falls within the segment delimited by the x, y, z and x', y', z' coordinates. Finally, it also takes into consideration the speed of both vehicles and the transmission range of VLC to determine if the vehicles are in range of one another and if the transmission succeeded.

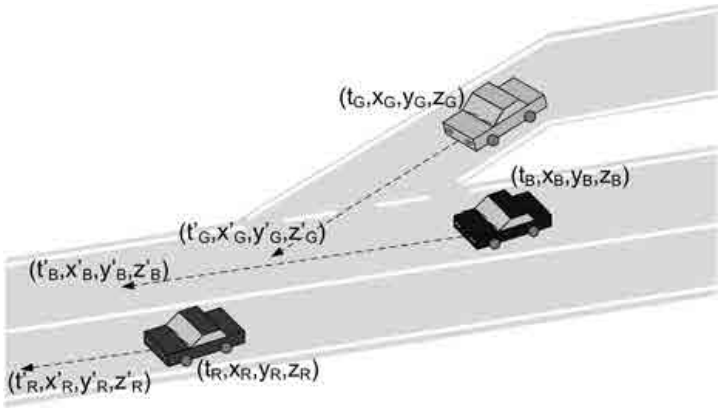


Figure 2: In range detection based on vehicles R, G, B trajectories.

In the next sections we present the results of our simulations. VLC can transmit at 115Kbps at approximately 5m [18], although in order to account for future improvements on the technology we will also consider ranges of 10m and 15m and compare those results to the results yielded by the range of WAVE-DSRC (120m). It should be noted that the vehicles or nodes being simulated spend an average of 3,134.17s on the road (slightly less than an hour) and make 1.99 trips. Our simulations were designed with the following goals in mind:

- compute the mean of the number of packets received by each node and its distribution.
- study the transmission of information over an area with a gossip protocol.
- identify the limitations of WAVE-DSRC on the usage of the physical medium.

3.1. Average number of received packets

As depicted in Fig.3, the average number of packets received by every node is computed. For ranges 5m, 10m and 15m it can be seen that a similar number of packets was received (443.38 packets, 458.55 packets, 473.88 packets). However, when compared with the 120m range of WAVE (1,491.60 packets) the difference in performance is quite evident. If we look at the distribution of the mean, it can be observed that the VLC ranges share similar results: over 150,000 nodes receive between 0 and 499 packets, while over 300 receive 2,500 or more. With a range of 120m, over 70,000 nodes in the WAVE VANET receive between 0 and 499 packets, while over 50,000 nodes receive 2,500 or more.

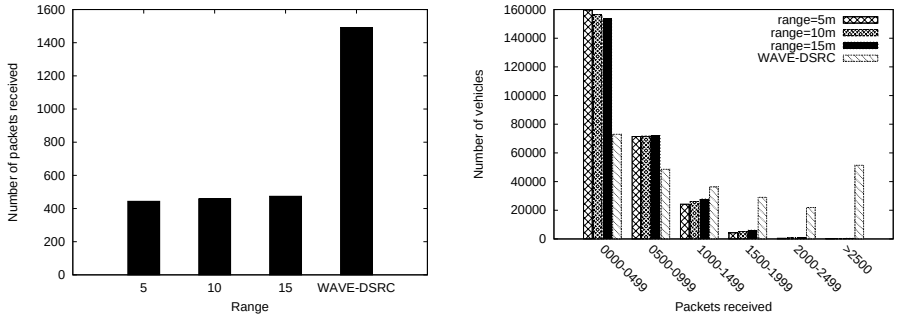


Figure 3: Mean and distribution of the number of packets received by each node.

Solely looking at these results it can be firmly stated that VLC cannot replace WAVE-DSRC without a decrease in the network's throughput. That being said, the results also show that even with a range of 5m 443 packets were received, which means that VLC may be able to work together with WAVE to protect VANETs from DoS attacks.

3.2. Received packets over an area

In order to find out how important the transmission range is to propagate a message over a certain area another experiment was designed. Considering the results from the previous section, a node which received an average number of packets was selected as a representative sample of the network population. In the new simulation, that node will broadcast a packet every time its position is updated, at the same time the rest of the network will remain silent until they receive that message. From that point onwards they too will broadcast the message to its neighbors and so on until the simulation finishes.

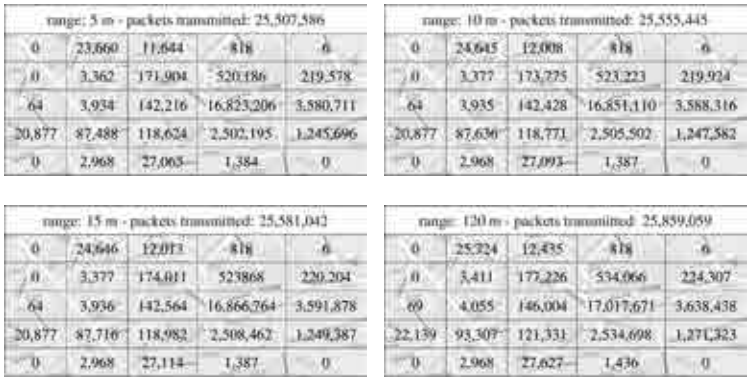


Figure 4: Distribution of packets transmitted in the traveled area.

In Fig. 4 we can see the result of the described scenario in the number of packets that were transmitted. The three different ranges for VLC (5m, 10m and 15m) obtained very similar results both in number of packets and their distribution. As far as WAVE is concerned, even though it produced approximately 300,000 transmissions more than VLC it did so with a very similar distribution. These results show that shorter transmission ranges can be compensated by the use of gossip broadcast protocols.

3.3. Analysis of WAVE scalability

In order to analyze the scalability of WAVE-DSRC a simulation in ns-3 [26] was implemented defining a vehicular scenario with 400 nodes arranged in 4 lanes as depicted in Fig.5, connected through a WAVE-DSRC 27Mbps link with a 120 meters range. This scenario represents a traffic jam, which is the worse possible situation for radio communication due to the high density of vehicles. It should be noted that our simulation uses ns-3 *YansWifiPhyHelper* and *YansWifiChannelHelper* classes, as defined in [27].

In a nutshell, the simulation schedules the broadcast of *numPackets* 1000 bytes packets at a randomly chosen time between the start of the simulation and its ending point, defined as *period*. For every scenario (*numPackets*/*period* combination) the number of broadcasts received by each of the 400 simulated nodes is computed ($results_{numPackets,period}$) and compared with how many broadcasts each of those nodes would have received without packets loss ($reference_{numPackets}$), considering the mean as the scenario's result:

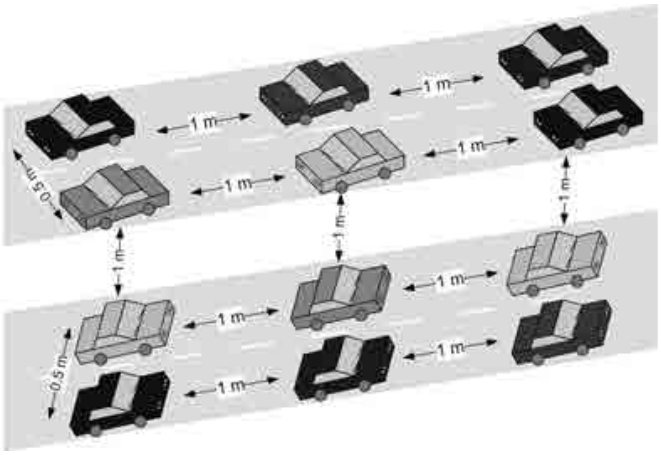


Figure 5: Vehicle layout for the 400 nodes simulated in ns-3.

$$Received\ broadcasts\ \% = \sum_{node=1}^{400} \left(\frac{results^{node}_{numPackets,period}}{reference^{node}_{numPackets}} \right) \quad (1)$$

Looking at the results in table 1 it can be seen that for 400 packets every 30s the percentage of received broadcasts drops to 60.52%; the general tendency is that for a high number of packets transmitted over short periods the network throughput decreases. It should be noted that in this simulation we considered a scenario where every node broadcasts a message and there are no acknowledgements or retries. Had we considered bidirectional communication between vehicles and a road side unit the network throughput would have been even lower due to the number of retries. We strongly believe that VLC could help improve the delivery rate because in VLC users do not have to compete for the physical medium.

4. Conclusions

In this article we have explored the future possibilities of VLC replacing or complementing the current standard for communication in VANETs (WAVE-DSRC). Several experiments were prepared, each with a different objective in mind: (i) determine how many packets are received by each node (on average), (ii) how the transmitted information is distributed when VLC and WAVE are compared and (iii) analyze the success data rate of a worst case scenario (traffic jam) with WAVE.

Percentage of received broadcasts						
Number of packets / Period (s)	10	20	30	40	50	60
100	71.82	87.08	91.48	93.66	95.03	95.93
200	36.23	71.79	82.08	87.04	89.71	91.46
300	15.77	54.50	71.75	79.71	84.06	87.05
400	9.45	36.71	60.52	71.88	78.14	82.21
500	6.83	23.43	48.62	63.59	72.13	77.26
600	5.28	15.89	36.77	54.55	65.24	71.99
700	4.28	11.88	27.18	45.53	58.29	66.44
800	3.64	9.51	20.41	36.85	51.05	60.65

Table 1: Percentage of received broadcasts for every simulated scenario.

The first experiment shows that every node receives at least three times as many packets with WAVE as they receive with VLC in any of its different transmission ranges. For the second simulation we choose a node which receives an average number of packets and make him transmit in an epidemic way (at the beginning of the simulation he is the only one transmitting, but once a node receives that packet he starts transmitting as well). The results show that even though WAVE-DSRC obtained a higher number of transmitted packets, i.e., infected more nodes, the distribution in the x, y, z space was very similar. Which leads us to the conclusion that the short range of VLC can be made up for with the use of epidemic or gossip protocols. Finally, the third simulation shows at which point WAVE-DSRC stops getting information through due to the high competition for the medium and the resulting packet collisions. At that point, the network throughput could be improved by using VLC to transmit as well, since in VLC nodes do not need to compete for the physical medium due to the nature of light communication.

In addition, we also need to consider the fact that while WAVE, like all radio communication, is subject to jamming VLC is not. With WAVE, an attacker with a powerful enough radio device could easily cause a blind spot in the network (which would lead to a DoS) with dimensions depending on how good is his equipment. However, in order to jam the transmission of information in VLC the attacker would have to physically block the beam of light from the emitter to the receiver.

All in all, we believe that once VLC is ready to be deployed in the open air it will be an important addition to VANET communication. Working together with WAVE-DSRC, it will

provide an extra link which can be used by public safety applications and whenever the WAVE-DSRC performance is below a certain threshold either due to the medium congestion or to an attack.

5. Acknowledgement

This work was partially supported by the EuroNF NoE and by Spanish grants TIN2010-21378-C02-01 and 2009-SGR-1167.

References

[1] M. Li, Z. Yang, W. Lou, Codeon: Cooperative popular content distribution for vehicular networks using symbol level network coding, Selected Areas in Communications, IEEE Journal on 29 (1) (2011) 223 –235. doi:10.1109/JSAC.2011.110121.

[2] Y. Zhang, J. Zhao, G. Cao, Roadcast: a popularity aware content sharing scheme in vanets, SIGMOBILE Mob. Comput. Commun. Rev. 13 (4) (2009) 1–14.

[3] S.-B. Lee, G. Pan, J.-S. Park, M. Gerla, S. Lu, Secure incentives for commercial ad dissemination in vehicular networks, in: MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing, ACM, New York, NY, USA, 2007, pp. 150–159.

[4] T. Delot, N. Cenerario, S. Ilarri, S. Lecomte, A cooperative reservation protocol for parking spaces in vehicular ad hoc networks, in: Mobility '09: Proceedings of the 6th International Conference on Mobile Technology, Application & Systems, ACM, New York, NY, USA, 2009, pp. 1–8.

[5] Y.-T. Tseng, R.-H. Jan, C. Chen, C.-F. Wang, H.-H. Li, A vehicle-density-based forwarding scheme for emergency message broadcasts in vanets, in: Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on, 2010, pp. 703 –708. doi:10.1109/MASS.2010.5663803.

[6] R. A. Uzcategui, G. Acosta-Marum, Wave: a tutorial, Communications Magazine, IEEE 47 (5) (2009) 126–133.

[7] H. L. Minh, D. O'Brien, G. Faulkner, L. Zeng, K. Lee, D. Jung, Y. Oh, E. T. Won, 100-mb/s nrz visible light communications using a postequalized white led, Photonics Technology Letters, IEEE 21 (15) (2009) 1063 –1065. doi:10.1109/LPT.2009.2022413.

[8] H. L. Minh, D. O'Brien, G. Faulkner, L. Zeng, K. Lee, D. Jung, Y. Oh, 80 mbit/s visible light communications using pre-equalized white led, in: Optical Communication, 2008. ECOC 2008. 34th European Conference on, 2008, pp. 1 –2. doi:10.1109/ECOC.2008.4729532.

[9] IEEE 802.15 Task Group 7 (TG7) Visible Light Communication (Nov. 2012).
URL `http://www.ieee802.org/15/pub/TG7.html`

[10] Visible Light Communications Consortium (Nov. 2012).
URL `http://www.vlcc.net`

[11] D. A. Rivas, M. Guerrero-Zapata, Chains of trust in vehicular networks: A secure points of interest dissemination strategy, Ad Hoc Networks 10 (6) (2012) 1115 – 1133. doi:10.1016/j.adhoc.2012.02.011.

[12] D. A. Rivas, M. Guerrero-Zapata, Simulation of points of interest distribution in vehicular networks, SIMULATION 88 (11) (2012) 1390–1404. doi:10.1177/0037549712456440.

[13] Realistic mobility vehicular trace by K. Nagel at ETH Zurich (Nov. 2012).
URL `http://www.lst.inf.ethz.ch/research/ad-hoc/car-traces/`

[14] M. Raya, J.-P. Hubaux, The security of vehicular ad hoc networks, in: SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks, ACM, New York, NY, USA, 2005, pp. 11–21. doi:http://doi.acm.org/10.1145/1102219.1102223.

[15] D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, W. Schulz, Cartalk 2000: safe and comfortable driving based upon inter-vehicle-communication, in: Intelligent Vehicle Symposium, 2002. IEEE, Vol. 2, 2002, pp. 545–550 vol.2.

[16] N.-W. Lo, H.-C. Tsai, A reputation system for traffic safety event on vehicular ad hoc networks, EURASIP J. Wirel. Commun. Netw. 2009 (2009) 9:1–9:2. doi:http://dx.doi.org/10.1155/2009/125348.

[17] A. G. Bell, On the production and reproduction of sound by light, American Journal of Science, Third Series XX (11) (1880) 305–324.

[18] IEEE 802.15 Visible Light Communication Overview (Nov. 2012).
URL http://ieee802.org/802_tutorials/2008-03/15-08-0114-02-0000-VLC_Tutorial_MCO_Samsung-VLCC-Oxford_2008-03-17.pdf

[19] C. Davis, I. Smolyaninov, S. Milner, Flexible optical wireless links and networks, Communications Magazine, IEEE 41 (3) (2003) 51 – 57. doi:10.1109/MCOM.2003.1186545.

[20] P. Chowdhury, M. Tornatore, S. Sarkar, B. Mukherjee, Building a green wireless-optical broadband access network (woban), Lightwave Technology, Journal of 28 (16) (2010) 2219 –2229. doi:10.1109/JLT.2010.2044369.

[21] S. Sarkar, S. Dixit, B. Mukherjee, Hybrid wireless-optical broadband-access network (woban): A review of relevant challenges, Lightwave Technology, Journal of 25 (11) (2007) 3329 –3340. doi:10.1109/JLT.2007.906804.

[22] Q. Liu, C. Qiao, G. Mitchell, S. Stanton, Optical wireless communication networks for first- and last-mile broadband access], J. Opt. Netw. 4 (12) (2005) 807–828. doi:10.1364/JON.4.000807.

[23] D. O'Brien, L. Zeng, H. Le-Minh, G. Faulkner, J. Walewski, S. Randel, Visible light communications: Challenges and possibilities, in: Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on, 2008, pp. 1 –5. doi:10.1109/PIMRC.2008.4699964.

[24] S. Iwasaki, M. Wada, T. Endo, T. Fujii, M. Tanimoto, Basic experiments on paralle wireless optical communication for its, in: Intelligent Vehicles Symposium, 2007 IEEE, 2007, pp. 321 –326. doi:10.1109/IVS.2007.4290134.

[25] V. Naumov, R. Baumann, T. Gross, An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces, in: MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '06, ACM, New York, NY, USA, 2006, pp. 108–119. doi:http://doi.acm.org/10.1145/1132905.1132918.

[26] T. R. Henderson, S. Roy, S. Floyd, G. F. Riley, ns-3 project goals, in: Proceeding from the 2006 workshop on ns-2: the IP network simulator, WNS2 '06, ACM, New York, NY, USA, 2006. doi:http://doi.acm.org/10.1145/1190455.1190468.

[27] M. Lacage, T. R. Henderson, Yet another network simulator, in: Proceeding from the 2006 workshop on ns-2: the IP network simulator, WNS2 '06, ACM, New York, NY, USA, 2006. doi:http://doi.acm.org/10.1145/1190455.1190467.