On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks*

Maxim Raya†, Panos Papadimitratos†, Virgil D. Gligor‡, Jean-Pierre Hubaux†

† Laboratory for computer Communications and Applications (LCA)
School of Computer and Communication Sciences
EPFL, Switzerland
‡ Department of Electrical and Computer Engineering
University of Maryland at College Park, USA

Technical report LCA-REPORT-2007-003

ABSTRACT

We argue that the traditional notion of trust as a relation among entities, while useful, becomes insufficient in ephemeral ad hoc networks. In this paper, we address the challenge of extending the traditional notion of trust to *data-centric trust*, that is, trustworthiness attributed to node-reported data per se. We propose a framework for data-centric trust establishment: First, trust in each individual piece of data is computed; then multiple, related but possibly contradictory, data are combined; finally, their validity is inferred by a decision component based on the Dempster-Shafer Theory. We are especially interested in, yet not restricted to, *ephemeral* ad hoc networks, i.e., highly volatile systems with short-lived node encounters. We consider and evaluate an instantiation of our framework in vehicular networks as a case study. Our simulation results show that our scheme is highly resilient to attackers and converges fast to the correct decision.

1. INTRODUCTION

In all traditional notions of trust, data trust (e.g., trust in the identity or access/attribute certificates) was based exclusively on a priori trust relations established with the network entities producing these data (e.g., certification authorities, network nodes) [8, 16, 17]. This was also the case when trust was derived via fairly lengthy interactions among nodes, as in reputation systems [2, 19, 28]. Moreover, any new data trust relationships that needed to be established required a priori trust in the entity that produced those data. All trust establishment logics proposed to date have been based on entities (e.g., "principals" such as nodes) making statements on data [2, 7, 8, 13, 16, 17, 25, 26]. Furthermore, traditional trust relations were generally time-invariant: once established, they lasted a long time.

Based on the above observations on existing *entity-centric* notions, this paper approaches trust from a different point of view: we are concerned with *data-centric* trust establishment. The problem at hand is *how to evaluate the trustworthiness of the data reported by other entities* rather than of *the entities themselves*. This question is crucial for emerging data-centric networks, including sensor networks, vehicular networks, and environment aware pervasive computing applications. A number of technical challenges are present. Primarily, the distinction among data reporting nodes is blurred by the network operation itself (e.g., due to high mobility, privacy measures). Moreover, nodes can be unreliable, faulty, or not sufficiently equipped for accurate data reporting. To make things worse, the network operation can be *ephemeral*. A typical type of ephemeral networks are vehicular networks, featuring short encounters between nodes, high mobility, and large scale.

Under such conditions, the question remains: How can a network node trust node-reported data, especially when contradictory pieces of evidence are received? We propose a solution for exactly this problem: a data-centric trust establishment framework that can be applied in any ad hoc network and, most often, ephemeral networks. The logic we propose extends the traditional notions of trust and methods of trust establishment in several ways.

First, unlike traditional trust, a priori trust relationships in entities (nodes) represent only one of the default parameters for establishing data trust. For example, our logic, while using nodes' statements on data, does not rely exclusively on such statements. Instead, it takes into account dynamic factors, such as location and time, as well as the number and type of the statements on data, to derive data trust relations. Second, beyond the traditional time-invariant or slow-evolving trust notions, data-centric trust relations are by definition ephemeral and have to be established and re-established frequently, based on network and perceived environment changes. Just like the network itself, data trust relations are ephemeral. For example, an event report (e.g., accident report, weather report) that must be believed by recipient nodes in real-time cannot last

*This work is partially funded by the EU project SEVECOM (http://www.sevecom.org).

longer than the lifetime of the event or the network formation. Multiple rounds of node interactions are typically not possible in such networks. Third, trust does not stem from a single source of data (e.g., a certification authority) and generally it is not application-independent (e.g., when multiple applications use certificates for their access control and authentication policies). In contrast, we derive data-centric trust relations from multiple pieces of evidence, including environmental data, and very rarely if at all, from exclusively a single node report. Our logic weighs each individual piece of evidence according to well-established rules, and takes into account various trust metrics defined specifically in the context of an application. Then, data and their respective weights serve as inputs to a decision logic that outputs the level of trust in this collection of evidence, and more importantly that the event has taken place or not.

In the rest of this paper, we present our framework in Sec. 2. In Sec. 3 we mathematically develop our approach. Then, we instantiate our framework in the context of vehicular communication systems in Sec. 4. We evaluate the effectiveness of our scheme through simulations in Sec. 5, and conclude with a survey of related work in Sec. 6.

2. GENERAL FRAMEWORK

2.1 Preliminaries

We consider systems with an authority responsible for assigning identities and credentials to all system entities that we denote as *nodes*. All legitimate nodes are equipped with credentials (e.g., certified public keys) that the authority can revoke. Specific to the system and applications, we define $\Omega = \{\alpha_1, \alpha_2, \ldots, \alpha_I\}$, a set of mutually exclusive *events* where Ω is by no means the set of all possible events in the system. α_i is a perceivable environment, network, or application event. There may be multiple applications, each having its own set of relevant events Ω_j. These sets are overlapping, as their events can belong to a basic pool of events, e.g., regarding location or time.

We consider *reporters of events*, that is, nodes $v_k \in V$, classified according to a system-specific set of node types, $\Theta = \{a, b, \ldots, z\}$. We define a function $\tau : V \rightarrow \Theta$ returning the type of node v_k. *Reports* are statements on events, including related time and geographic coordinates where applicable. For simplicity, we consider reports on single events, as reports on composite events are straightforward. We do not dwell on the exact method for report generation, as this is specific to the application.

2.2 Default Trustworthiness

We define the *default trustworthiness* of a node v_k of type a as a real value $0 < t_a < 1$, which depends on the attributes related to the designated type of node v_k. For all node types, there exists a trustworthiness ranking $t_a < t_b < \ldots < t_y < t_z$. For example, some nodes are better protected from attacks, more closely monitored and frequently re-enforced, and, overall, more adequately equipped, e.g., with reliable components. As they are less likely to exhibit faulty behavior, they are considered more trustworthy.

2.3 Event- or Task-Specific Trustworthiness

Nodes in general perform multiple tasks that are system-, node- and protocol-specific actions, with Λ being the set of all relevant system tasks. Then for some nodes v_1 and v_2 with default trustworthiness rankings $\tau(v_1) = a$ and $\tau(v_2) = b$ and $t_a < t_b$, it is possible that v_1 is more trustworthy that v_2 with respect to a task $\lambda \in \Lambda$.

Reporting data on events is clearly one of the node tasks. For the sake of simplicity, we talk here about event-specific trustworthiness implying that it is actually task-specific trustworthiness. Nevertheless, the two can be easily distinguished, when necessary; e.g., when tasks include any other protocol-specific action such as communication.

With the above considerations in mind, we define the event-specific *trustworthiness* function $f : \Theta \times \Lambda \rightarrow [0, 1]$. f has two arguments: the type $\tau(v_k)$ of the reporting node v_k and the task λ_j. f does differentiate among any two or more nodes of the same type, and if $\lambda_j = \emptyset$ (no specific event or task), f is the default trustworthiness $f = t_{\tau(v_k)}$.

2.4 Dynamic Trustworthiness Factors

The ability to dynamically update trustworthiness can be valuable, especially for capturing the intricacies of a mobile ad hoc networking environment. For example, nodes can become faulty or compromised by attackers and hence need to be revoked. In addition, the location and time of report generation change fast and are important in assigning trustworthiness values to events.

To capture this, we define a *security status* function $s : V \rightarrow [0, 1]$. $s(v_k) = 0$ implies node v_k is revoked, and $s(v_k) = 1$ implies that the node is legitimate. Intermediate values can be used by the system designer to denote different trustworthiness levels, if applicable.

Second, we define a set of *dynamic trust metric* functions $M = \{\mu_l : V \times \Lambda \rightarrow [0, 1]\}$ indexed by a selector l indicating different node attributes (e.g., location) that dynamically change. That is, for each attribute a different metric μ_l is defined. μ_l takes node $v_k \in V$ and task $\lambda_j \in \Lambda$ as inputs and returns a real value in $[0, 1]$. Metrics are calculated by a node v_k for each of the nodes $v_i, i \neq k$, that have interacted with (and possibly observed) v_k within a time window.

2.5 Location and Time

Among the possible l for metric μ_l, *proximity* either in *time* or *geographic location* is an attribute of particular importance. Proximity can increase the trustworthiness of a report: The closer the reporter is to the location of an event, the more likely it is to have accurate information on the event. Similarly, the more recent and the closer to the event occurrence time a report is generated, the more likely it is to reflect the system state.

Cryptographic primitives, such as digital signatures, can ensure that location and time information cannot

be modified if included in a report. However, the accuracy of such information can vary, due to nodes' differing capabilities or (malicious or benign) faults. This is especially true for reports that depend on fine-grained time and location data. Hence, different types of nodes are more or less trustworthy when reporting such data. In some cases, time- or geo-stamping a report can be a distinct task.

2.6 Scheme Overview

We compute the trustworthiness of a report by using both (i) static or slow-evolving information on trustworthiness, captured by the default values and the event-specific trust f, and (ii) dynamically changing information captured by security status s and more so by metric μ_l. We combine these as arguments to a function

$$F = F(s(v_k), f(\tau(v_k), \lambda_j), \mu_l(v_k, \lambda_j))$$

that returns values in the $[0, 1]$ interval. These values are calculated locally for each report received from another node and are called the *weights* (or *trust levels*) of the reports. Fig. 1 illustrates our scheme.

Nonetheless, such a per message assessment may often be insufficient. It can be hard to decide whether the reported event took place based on a single message, and it is vulnerable to faults (e.g., equipment failures or compromised nodes). Instead, we propose the collection of multiple reports related to the same event and of their weights, i.e., the accompanying F value, and their combination into a robust decision scheme. Then, the reports along with their weights are passed to a *Decision Logic* module that outputs an assessment on the event in question.[1] More important, we are interested in a system that not only decides on an event but also identifies the residual uncertainty regarding the event (alternative algorithms for the decision logic are discussed in detail in Sec. 3.2). The way to use such decisions and inferences is beyond the scope of this paper, as it is specific to particular systems.

3. EVIDENCE EVALUATION

The literature on trust in ad hoc networks proposes several approaches for trust establishment, which we survey in Sec. 6. In this work, we look at evidence evaluation for trust establishment as a decision-making process that emulates human reasoning. More specifically, the lack of knowledge about an event is not necessarily a refutal of the event. In addition, if there are two conflicting events, uncertainty about one of them can be considered as supporting evidence for the other (if not event α_1 then maybe event α_2). The Dempster-Shafer Theory (DST) [21] addresses the above two issues (lack of knowledge and conflicts) and hence seems appropriate for the type of decision problems we study in this paper.

[1]It is possible that a decision regards not a single event α_i but also a composite event, consisting of union(s) and intersection(s) of multiple α_i.

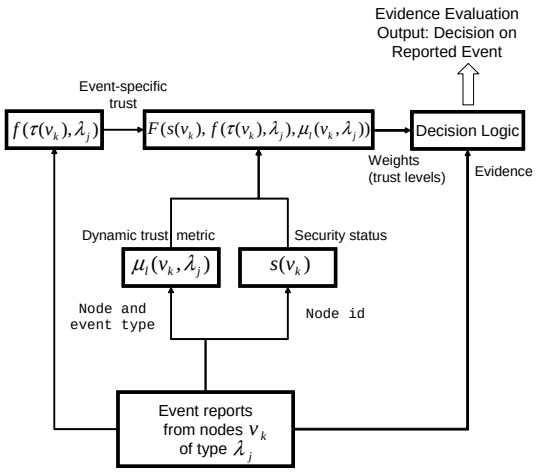


Figure 1: Data-centric trust establishment framework.

To illustrate our approach, we use a practical example from vehicular networks. Assume a vehicle A has to choose between two routes depending on whether they are congested or not[2]. To complicate the decision process, reports from respective routes contain conflicting evidence that can make a difference in the route A would choose. We denote the hypothesis that route 1 is congested (and hence A chooses route 2) by H and its opposite (route 1 is not congested and hence A chooses it) by $\bar{H}$. There are two types of evidence that each vehicle can report: e_k^1 means that the report of vehicle k confirms hypothesis H (congestion on route 1) and e_k^0 means that the report of vehicle k confirms hypothesis $\bar{H}$ (or simply does not support hypothesis H). This last distinction between confirming $\bar{H}$ and not supporting H is a distinguishing property of DST as shown later. In the following, we mathematically develop the decision logics based on Bayesian inference [23] and DST.

3.1 Bayesian inference

In Bayesian inference, the posterior probability of a hypothesis H given new evidence e is expressed in terms of the prior probability $P[H]$ using the Bayes' theorem:

$$P[H|e] = \frac{P[H]P[e|H]}{P[e]} \quad (1)$$

Given K independent pieces of evidence e_k (reports from K distinct vehicles), the posterior probability can be computed iteratively as:

$$P[H|e] = \frac{P[H]\prod_k P[e_k^i|H]}{P[H]\prod_k P[e_k^i|H] + P[\bar{H}]\prod_k P[e_k^i|\bar{H}]} \quad (2)$$

where $i \in \{0, 1\}$ and $P[e_k^1|H]$ is the probability that vehicle k confirms hypothesis H, given that H is true. Using trust levels as weights of binary reports (1 or 0

[2]Multi-valued cases can be derived from the following analysis in a straightforward way by using several hypotheses.

equivalent to H or $\bar{H}$, respectively), this probability is equal to the *trust level*.

$P[e_k^0|H]$ is the probability that vehicle k does not confirm H (hence, it confirms $\bar{H}$), given that H is true. This is equivalent to a malfunctioning or cheating vehicle (ideally, a vehicle would report a real event). $P[e_k^0|H] = 1 - P[e_k^1|H]$.

$P[e_k^0|\bar{H}]$ is the probability that vehicle k does not confirm $\bar{H}$, given that H is false. In other words, this is the probability that vehicle k confirms $\bar{H}$, given that $\bar{H}$ is true. As above, this is equal to the *trust level*.

$P[e_k^1|\bar{H}]$ is the probability that vehicle k confirms H, given that $\bar{H}$ is true. As before, this is the probability of malfunctioning or cheating. $P[e_k^1|\bar{H}] = 1 - P[e_k^0|\bar{H}]$.

3.2 Dempster-Shafer Theory

The major difference between Bayesian inference and DST is that the latter is more suitable for cases with uncertain or no information. More precisely, in Bayesian inference a node either confirms or refutes an event, whereas in DST a node does not necessarily refute the event. For example, if a node A confirms the presence of an event with probability p, in Bayesian inference it refutes the existence of the event with probability $1-p$. In DST, probability is replaced by $belief$. Hence, in this example, node A has p degree of belief in the event and 0 degree of belief in its absence. 0 in this case is called *plausibility* and means that A provides no support for the absence of the event (but it does not necessarily refute this absence as in Bayesian inference).

Belief and plausibility are the upper and lower bounds, respectively, of the probability of an event. The frame of discernment Ω contains all mutually exclusive possibilities related to an observation[3]. Thus, in the case of a hypothesis H with a binary state, $\Omega = \{H, \bar{H}\}$. The belief value corresponding to hypothesis H and provided by vehicle v_k is computed as:

$$bel_k(H) = \sum_{q:e_q \subset H} m_k(e_q) \tag{3}$$

which means it is the sum of all basic belief assignments $m_k(e_q)$, e_q being all pieces of evidence supporting hypothesis H. As the hypothesis H is binary in our example and hence the only piece of evidence provided by v_k is the affirmative report, $bel_k(H) = m_k^1(H)$.

The plausibility value corresponding to hypothesis H represents the sum of all evidence that does not refute H and is computed as:

$$pls_k(H) = \sum_{r:e_r \cap H \neq \emptyset} m_k(e_r) \tag{4}$$

Belief and plausibility are related by $pls(H) = 1 - bel(\bar{H})$.

Independent pieces of evidence (provided by independent observing vehicles) can be combined using Dempster's rule for combination:

$$m_1(H) \bigoplus m_2(H) = \frac{\sum_{q,r:e_q \cap e_r = H} m_1(e_q) m_2(e_r)}{\sum_{q,r:e_q \cap e_r = \emptyset} m_1(e_q) m_2(e_r)} \tag{5}$$

Belief can be computed similarly by iterative combination of independent basic beliefs assignments $m_k^i(H)$ (where $i \in \{0,1\}$) that either confirm H (i.e., $m_k^1(H)$) or do not refute H (i.e., $m_k^\Omega(H)$):

$$bel(H) = \bigoplus_k bel_k(H) = \bigoplus_k m_k^i(H) \tag{6}$$

$m_k^1(H)$ is the basic belief assignment, reported by vehicle k, that confirms H. As before, using trust levels as weights of binary reports, this value is equal to the *trust level*.

$m_k^0(H) = 0$ is the basic belief assignment, reported by vehicle k, that refutes H. As explained before, in DST, non-supporting evidence is not refuting evidence.

$m_k^\Omega(H)$ is the basic belief assignment corresponding to the universal hypothesis Ω. It represents the uncertainty and can support either H or $\bar{H}$. Hence $m_k^\Omega(H) = 1 - m_k^1(H)$.

Similarly, $m_k^0(\bar{H})$ is equal to the *trust level*, $m_k^1(\bar{H}) = 0$, and $m_k^\Omega(\bar{H}) = 1 - m_k^0(\bar{H})$.

The expressions of the other values $bel(\bar{H})$, $pls(H)$ and $pls(\bar{H})$ can be derived similarly.

4. CASE STUDY

4.1 Secure Vehicular Communications System

Vehicular Ad hoc NETworks (VANET) and *Vehicular Communication (VC)* systems [18] seek to enhance the safety and efficiency of transportation systems, providing, for example, warnings on environmental hazards (e.g., ice on the pavement) and traffic and road conditions (e.g., emergency braking, congestion, or construction sites). From a networking point of view, the nodes are vehicles and road-side infrastructure units (RSUs), all equipped with on-board processing and wireless modules, enabling multi-hop communication in general.

Authorities are public agencies or corporations with administrative powers; e.g., city or state transportation authorities entrusted with the management of node *identities* and *credentials*. A subset of the infrastructure nodes serves as a gateway to and from the authorities.

We assume that each node v_k is equipped with a pair of private/public cryptographic keys Pr_k/Pu_k, and a certificate issued by an authority X as $Cert_X\{Pu_k\}$. Nodes are equipped with a clock and a positioning system (such as GPS or Galileo). This allows them to include their time and location information in any outgoing reports. All messages are digitally signed.

Unicast and multicast communication is possible; however, controlled flooding and geo-casting (flooding controlled by the coordinates of the targeted area and receivers) are predominantly in use. A large portion of the vehicular network traffic is broadcasted at the network or application layers. Vehicle-specific information (e.g.,

[3]We use the same notation Ω as in Sec. 2.1 as both sets correspond to each other in this case.

velocity, coordinates) is transmitted frequently and periodically in the form of *safety messages*. In addition, reports are *triggered* by in-vehicle or network events. The combined safety and other messages, generated by vehicles and RSUs, can result in an abundant influx of information about events. It is important to note here that our approach based exclusively on local processing does not add any communication overhead and very little computation overhead in the nodes.

A general overview of VC security and its issues can be found in [20] and references within.

4.1.1 Adversary Model

Nodes either comply with the implemented protocols (i.e., they are correct) or they deviate from the protocol definition and become adversaries. The behavior of an adversary can vary widely according to its capabilities. Any wireless-enabled device that runs a rogue version of the vehicular network protocol stack poses a threat. The types of attacks that can be mounted by either internal (equipped with credentials and cryptographic keys) or external adversaries vary greatly. In brief, adversaries can replay any message, jam communications, and modify (yet in a detectable manner due to the digital signatures) messages. More importantly, they can inject faulty data and reports, or control the inputs to otherwise benign nodes and induce them to generate faulty reports.

We assume that at most a small fraction of the nodes is faulty, and accordingly the fraction of the network area affected by adversaries is bounded. This bound on the presence of adversaries could be further refined by distinct values for different node types. However, this assumption does not preclude that a few adversarial or faulty nodes surround a correct node at some point in time.

4.2 Framework Instantiation

We focus on the use of our scheme on-board a vehicle. Clearly, it could be run on RSUs, nonetheless, the challenge is to design a scheme practical for nodes that are not part of the system infrastructure.

The forms of the f (event-specific trust), s (security status), μ_l (dynamic trust metric), and F (trust level) functions are determined by the secure VC system: they are either preloaded at the time the node is bootstrapped, or updated after the node joined the system. Their values are either provided by the authorities or distributed by the infrastructure. The desired degree of autonomy for individual nodes is integrated into and thus prescribed by the same system-wide methods. Based on area-wide measurements, an authority can be cognizant of the likelihood that faults occur in public vehicles and compromised road-side units. Accordingly, manufacturers provide the authority with results of field tests regarding the reliability of on-board equipment. These periodically estimated values are downloaded by the vehicles, at technical checks, when crossing the border of an area, or when significant changes of those values occur. For example, consider that a fault is discovered for a particular component and a vehicle recall is currently in effect. In this case, the infrastructure must distribute such up-to-date information.

Table 1 illustrates an example of default values for each node type. We assume that private vehicles are classified in different categories, especially due to the expected gradual deployment and diversity in VC systems. For example, different levels may be assigned to vehicles of foreign authorities, for internal administrative or compatibility reasons. Or, vehicles may be equipped with hardware and software of differing levels of sophistication. Also, vehicle models may comply with one among multiple certified levels of equipment. Similarly, public vehicles', as well as RSUs', trustworthiness varies according to the level of protection, physical or other, as well as their type of equipment.

Vehicle Type	**Trustworthiness** t
Private Vehicle - Level 1	0.50
Public Transport - Bus	0.80
Road Maintenance Vehicle	0.85
Police Car	0.90
Road Side Unit - Level 1	0.50

Table 1: Example of default trustworthiness values for nodes.

Table 2 illustrates example values of the event-specific trustworthiness f. Police cars are the first responders to accidents and thus their reports are the most trustworthy, whereas RSUs announce highly trusted junction warnings. A node may also simply be unauthorized to perform a particular task. For example, no node is authorized to issue a revocation list (RL), but nodes can be responsible for relaying the latest RL signed by the authority; a police car is more trustworthy in that task than a private vehicle or mid-level RSU. Occasionally, one could consider an unorthodox assignment of roles, in case of emergencies. For example, a junction warning is most likely to be issued by a road maintenance vehicle if a junction RSU is failing.

Trustworthiness is also adjusted by metric μ_l according to the reporter's proximity to the event. However, this is done in different ways, i.e., different functions, according to the reporter's type. For example, an RSU's trustworthiness decays slowly with the decreasing time proximity to the event, as the infrastructure disseminates relevant information for relatively long but necessary periods. For example, accident information is distributed as long as the traffic is affected or the attention of the drivers is needed.

Considering geographic proximity, trustworthiness of private vehicles decays with their distance from the event location. We express this in terms of the number of hops, $h(v_k) = \lceil d(v_k)/R \rceil$, where d is the distance of the reporting node from the accident and R is a nominal communication range. However, being within radio communication range does not ensure at all times first-hand contact with the reported event. If this distinction is mandated by the application, then the above definition is meaningful for nodes beyond those in immediate contact with the event. We use here $\mu_l(0, \lambda) = 1$, $\mu_l(1, \lambda) = 0.9$, $\mu_l(h, \lambda) = -0.25h + 1$ if $1 < h \leq 4$, and

Node Type	Task / Event Report				
	Traffic Jam	*Accident Alert*	*Junction Warning*	*Revocation List Dist.*	*Location Receipt*
Private Vehicle - L1	0.5	0.5	0.6	0.5	0.3
Road Maintenance Vehicle	0.7	0.6	0.9	0.7	0.5
Police Car	0.85	0.95	0.7	0.8	0.9
Public Transport - Bus	0.6	0.8	0.7	0.5	0.85
Road Side Unit - L1	0.5	0.8	0.95	0.8	0.8

Table 2: Example of trustworthiness values as a function of the node and event/task type.

$\mu_l(h, \lambda) = 0$ if $h > 4$. Of course, this is just an example, and any other function form can be devised.

The weight for each report is calculated by the following rule/expression:

$$F = s(v_k) \times f(\tau(v_k), \lambda_j) \times \mu_l(v_k, \lambda_j)$$

To illustrate our instantiation, we consider an example scenario: a collision at a junction between two vehicles A and B. After their airbag opening, they start broadcasting an accident report; A for example disseminates rpt_A = "Own accident; location L_A; time T_A." The system considers the locations L_A, L_B and the times T_A, T_B, which are very close to each other, to be the same. A and B are at full proximity ($h = 0$) to the event.

We consider a binary security status s of a vehicle: "revoked" or "valid", λ_j ="Accident Alert", and geographic proximity in number of hops as input to metric μ_l. rpt_A and rpt_B are received by nodes that either relay them or generate their own reports on the same accident. For example, a bus C reaches the accident site, comes to a stop, and generates its own report. A private vehicle D relays rpt_A, rpt_B and rpt_C. A police vehicle P broadcasts an own manually initiated on-site report. An RSU $h = 3$ hops from the accident site collects multiple reports, evaluates the evidence as described in Sec. 2.6, and generates its own report. For the police car P ($h = 0$), it is calculated $F = 0.95$, for Level-1 private vehicles at $h = 1$, $F = 0.45$, and for the RSU at the junction, assuming a Level-1 unit at $h = 2$ from the accident, $F = 0.4$. This probability can be used by the application to decide the consequent action; the specific use of these values by the application is beyond the scope of this paper. For example, a collision avoidance application can instruct the driver to start braking even when the collision probability is close to 0.5; a traffic jam avoidance application may only react if the probability is higher than 0.8.

5. PERFORMANCE EVALUATION

In this section, we examine the performance of the data trust establishment system described in the previous sections. We compare three decision logics: Bayesian inference, DST, and *weighted voting* that computes the difference between the sum of all supporting evidence (i.e., weights of reports affirming the event) and the sum of all refuting evidence; if this difference is positive, it outputs 1, otherwise it outputs 0. The decision logics based on Bayesian inference and DST are simulated using the mathematical frameworks developed in Sec. 3. The results show that DST performs overall better than the other two methods: First, both DST and weighted voting behave similarly in terms of decisions on events and resilience to attackers whereas Bayesian inference performs poorly in some cases; second, DST provides finer decision granularity than weighted voting.

We assume that a vehicle receives several reports concerning an event. A trust level is computed for each report as illustrated in Fig. 1. The vehicle then locally applies a decision logic that outputs the probability of the event.

To study the performance of each method, we varied several parameters, namely the *average trust level*, the *percentage of affirmative reports*, and *time*. The average trust level is the mean of the trust levels assigned to the reports received by the observing vehicle. The percentage of affirmative reports indicates how many reports affirm the event. We use time to study the speed of data trust establishment as the reports arrive from vehicles. We also study the effect of the *percentage of attackers* on the behavior of each decision method and hence the corresponding resilience, which is very important in a security context.

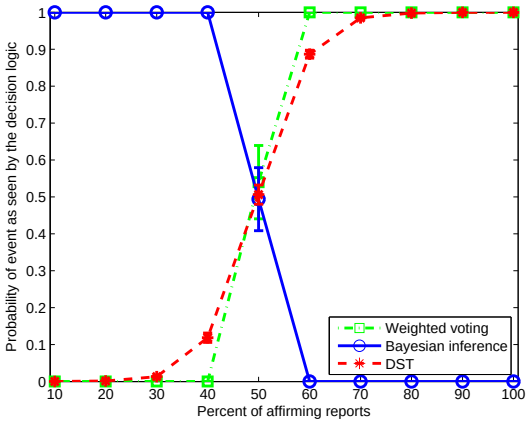
For Bayesian inference we use the neutral prior probability value of 0.5 (i.e., no prior knowledge). Simulations were performed in MATLAB and ns-2 (Sec. 5.3), results were averaged over 100 simulations and plotted with 95% confidence intervals.

5.1 Effect of the Average Trust Level

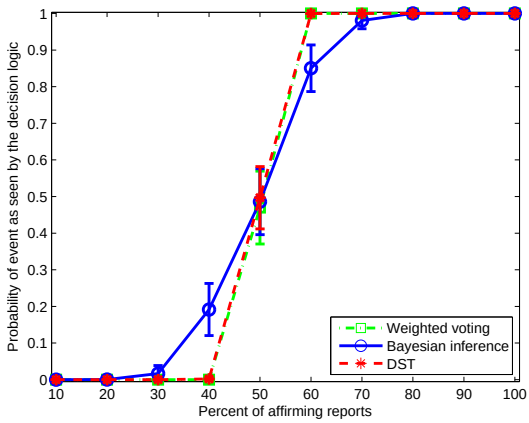
We use a Beta distribution, with its mean equal to the average trust level, to distribute the trust levels among the reports received by the observing vehicle. We chose the Beta distribution because it approximates the Normal distribution, a common choice in statistics, but with bounds (0 and 1). We simulate scenarios with two basic common cases: relatively numerous nodes/reporters and only a few ones. In each case, we also vary the average trust level between low and high.

Having experimented with several values, we chose the following as sample average trust levels: 0.1 for low[4] trust and 0.6 for high trust. The reason behind this

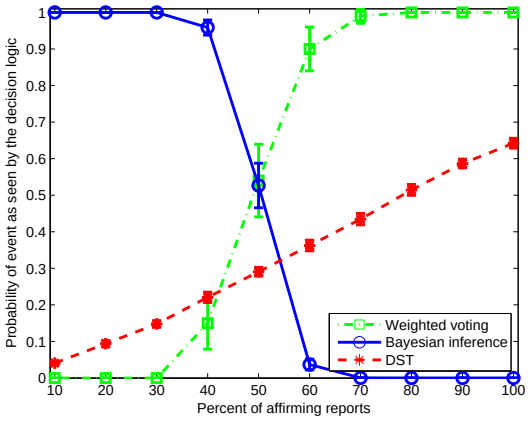
[4]Such values can result from low values of the security status s, e.g., due to the discovery of a virus in the network. If $s = 0.2$, $f = 0.5$, and $\mu = 1$ then $F = s \times f \times \mu = 0.1$ in the example in Sec. 4.2.



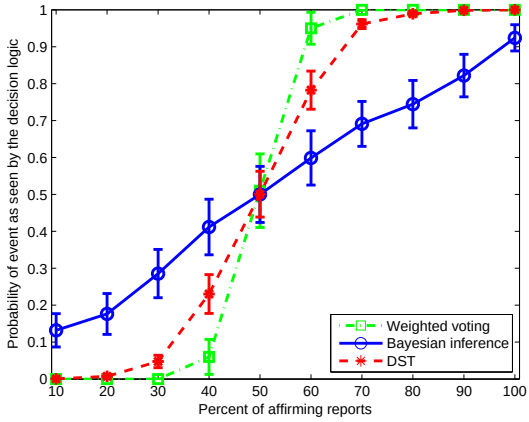
(a) Low average trust level (0.1), large number of reporters (100).



(b) High average trust level (0.6), large number of reporters (100).



(c) Low average trust level (0.1), small number of reporters (10).



(d) High average trust level (0.6), small number of reporters (10).

Figure 2: Performance of the three decision logics at different average trust levels and with respect to the percentage of affirmative reports. The upper two figures correspond to a large number of reporters whereas the figures below correspond to a small number of reporters.

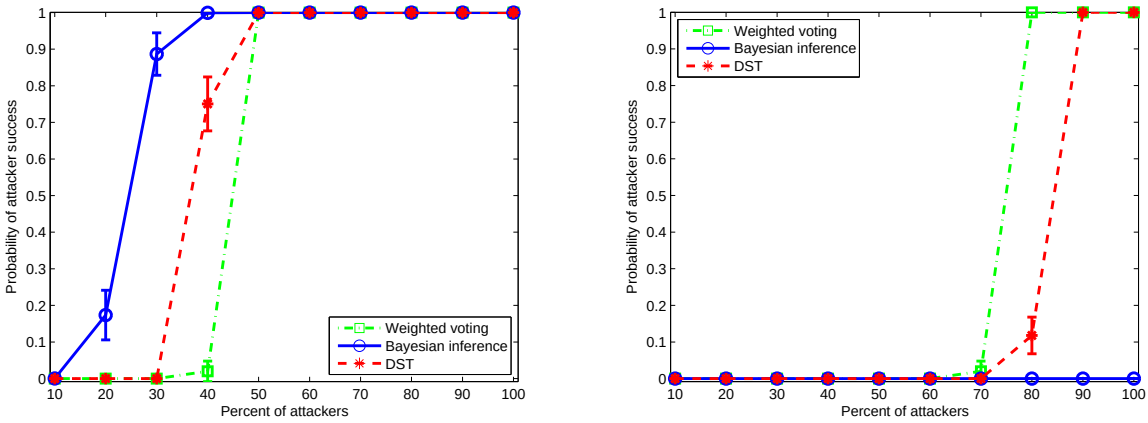
choice is that lower values show the behavior of the system at critical values, whereas higher values of trust in each range provide no additional information.

In Fig. 2, we observe that both weighted voting and DST behave similarly at both low and high trust levels. Given that an event happened, the probability of the event as seen by the observing vehicle increases as the percentage of affirmative reports increases. At low levels of trust, the evolution of DST is smoother than that of weighted voting because DST can output values other than 0 and 1. This means that the application logic has more available granularity with DST, which helps it make better informed decisions.

An interesting observation is related to the behavior of the Bayesian inference. At high trust levels (Fig. 2(b)), it exhibits behavior similar to that of the other two methods. But at low trust levels (Fig. 2(a)), it behaves opposite to the other two methods, because Bayesian inference deals with probabilities and if a report is assigned a 0.3 trust level (i.e., 0.3 probability of being correct), it is assumed to have a 0.7 mistrust level (i.e., 0.7 probability of being false). Thus, given a small percentage of affirmative reports with low trust levels, there is a high percentage of refuting reports with low trust levels also. In Bayesian logic, this high percentage transforms into a high percentage of affirmative reports with high trust levels (i.e., the opposite). Similar reasoning applies to high percentages of affirmative reports.

Another interesting parameter to observe is the number of reporters that the observing vehicle can hear. As we can see in Figs. 2(c) and 2(d), weighted voting does not differ much when the number of reporters is small (e.g., 10). But Bayesian inference and DST are more sensitive to this parameter. With a low number of reporters, DST yields higher output probabilities only at high trust values and hence represents better the typically cautious human response.

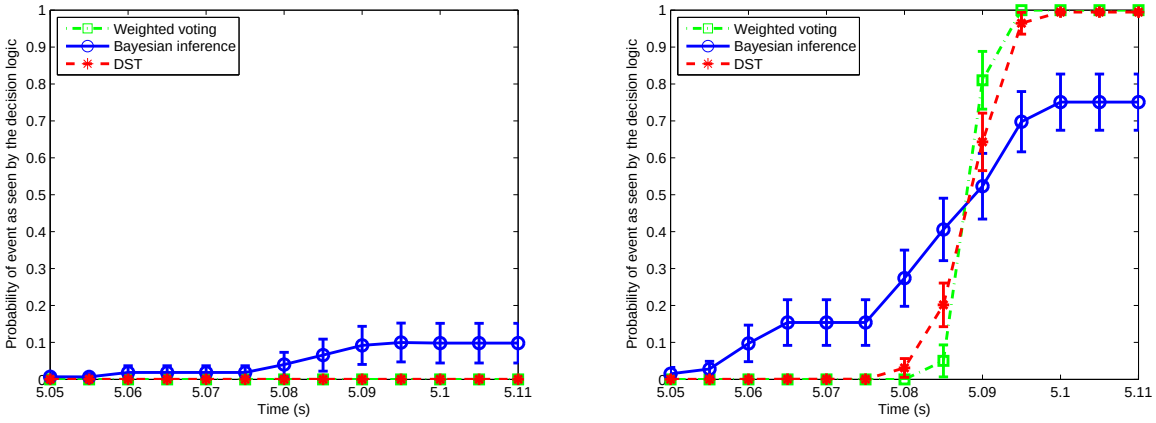
It is also worth noting that changing the value of the



(a) Average trust level of attackers (0.8) is higher than the average trust level of honest nodes (0.6).

(b) Average trust level of attackers (0.3) is lower than the average trust level of honest nodes (0.8).

Figure 3: Performance of the three decision logics at different average trust levels and with respect to the percentage of attackers.



(a) High trust, small percentage of affirmative reports (0.3).

(b) High trust, large percentage of affirmative reports (0.7).

Figure 4: Evolution of decision logic outputs with time in a VANET scenario.

prior probability in Bayesian inference does not lead to significant changes in the results. Bayesian inference also exhibits higher variance than the other two methods and hence it is less reliable in threshold-based schemes.

5.2 Effect of the Percentage of Attackers

It is important to analyze the resilience of the different decision logics to attackers. The graphs in Fig. 3 and the observations detailed above can provide us with valuable insight into the effect of the percentage of attackers. As mentioned in the adversary model, we assume that the attacker tries to falsify event reports in order to disturb the perception of the observing vehicles. In this case, colluding attackers report information opposite to that reported by honest vehicles. Thus, there are two different pieces of information that are conflicting in their content.

Both the trust distribution of honest nodes and that of attackers follow Beta distributions with different means[5]. So we examine two scenarios: the average attacker trust is higher than that of honest nodes (Fig. 3(a)) and vice versa (Fig. 3(b)).

We can conclude as a general rule that the higher the average trust level of attackers, the smaller their percentage needed for success. In Fig. 3(a), Bayesian inference is the least resilient to attackers and weighted voting is the most resilient among the three methods. But in Fig. 3(b), when the average attackers' trust is low, Bayesian inference never favors their information and thus is more resilient to them. This observation can be explained as follows: When the percentage of

[5]The case of equal means is represented in Fig. 2.

attackers is small (Fig. 2(b)), honest nodes influence the output probability; when their percentage is high (Fig. 2(a)), Bayesian inference outputs the opposite of their reports.

5.3 Evolution in Time

In ephemeral networks, it is important to evaluate data trust rapidly in order to permit an application logic to use the resulting values. Hence, a decision logic should be able to output the final result as fast as possible, based on the freshly received reports. This property distinguishes the mechanisms explored in this work from other approaches that rely on a longer history of available reports (e.g., reputation systems [2, 19, 28]). The results show how fast an observing vehicle can make a decision once the reporters have detected an event. In this section, we are only interested in the networking delay of the event detection as inferred by the decision logic[6].

To simulate ephemeral networks, we used VANETs with highly mobile vehicles. Moreover, decisions in these VANETs should be made fast because they may incur life-critical consequences. We use a highway scenario with 3 lanes in each direction. Vehicles are moving at speeds between 90 km/h (≈ 56 miles/h) and 150 km/h (≈ 93 miles/h); the average distance between two vehicles on the same lane is 50 m (≈ 164 ft). To simulate the networking aspects of VANETs, we assume that vehicles periodically broadcast safety messages every 300 ms within a radius of 300 m, according to the DSRC specification [1]. In our simulations, vehicles begin message broadcasting at second 5 and receive reports from around 43 reporters on average.

We examine both cases of small (Fig. 4(a)) and large (Fig. 4(b)) percentages of affirmative reports. Fig. 4 shows how fast the three decision logics reach their final output values as event reports arrive. The obtained graphs show that this happens within 100 ms, which is fast enough to make a decision and consequently broadcast a safety message. The final output values are in concordance with the results of Fig. 2.

6. RELATED WORK

Work on trust has produced rich literature in conventional, P2P and ad hoc networks. In the latter, most works share assumptions that there is no infrastructure and no PKI; trust is a relation among entities; trust is based on observations, with a history of interactions needed to establish trust. To the best of our knowledge, the computation of trust values in the context of ad hoc networks has been considered in only two cases: certification [7, 11, 26] and routing [2, 28]. Otherwise, trust evaluation assumes the prior establishment of trust relations. In both certification and routing, trust values are established in very specific ways that cannot be generalized to other approaches.

[6]The total event detection delay by the observing vehicle depends also on how fast the reporters detect the event, which in turn depends on the particular detection sensors and hence we do not consider it in this work.

Eschenauer et. al. [7] introduce the general principles of trust establishment in mobile ad hoc networks and compare them to those in the Internet. They describe examples of generic evidence generation and distribution in a node-centric authentication process.

Several papers [2, 28] describe the use of modified Bayesian approaches to build reputations systems with secondhand information to establish trust in routing protocols.

The main approach advanced by Jiang and Baras [12, 13] is based on local voting that is a weighted sum of votes. Conflicting votes are mitigated by each other when summed. Voting cannot properly address conflicts between relative majorities in two distinct groups of voters (e.g., "Which group to trust: 9 out of 10 or 50 out of 100?"). These works also favor local interactions that we use as well.

The main idea behind the work by Sun et. al. [24, 25] is that trust represents uncertainty that in turn can be computed using entropy. They also introduce the notion of *confidence of belief* to differentiate between long-term and short-term trust. Trust can be established through direct observations or through a third party by recommendations.

Theodorakopoulos and Baras [26] assume the transitivity of trust to establish a relation between two entities without previous interactions. In this context, they model trust evaluation as a path problem on a directed graph. Given that nodes sign certificates for each other without any security infrastructure, this work extends PGP [27] by using secondhand evidence. Routing protocols are the main target of this approach.

Hubaux et al. [11] propose a distributed version of PGP for ad hoc networks. In their approach, nodes store partial local certificate repositories. When two nodes want to establish a certificate chain between them, they merge their repositories. In [3], Capkun et. al. exploit mobility and *friend* relations to establish security associations between nodes, but at the cost of additional delay.

The Internet and peer-to-peer (P2P) networks provide a rich pool of work on reputation systems. A comprehensive survey on these systems can be found in [15] and [19].

More closely related to VANETs and thus the case-study instantiation of our framework, Doetzer et al. [6] introduce a reputation system for VANETs. A vehicle makes, over time, opinions of other vehicles based on the consistency of their reports with its own observations. Moreover, vehicles propagate opinions by *piggybacking* them on messages. Another paper by Golle et. al. [9] proposes a framework for data validation in VANETs; it consists in comparing received data to a *model of the VANET* and accept their validity if both agree.

There is little work on applying the Dempster-Shafer Theory to ad hoc networks, the most relevant to our work is the paper by Chen and Venkataramanan [4] that describes how DST can be applied to distributed intrusion detection in ad hoc networks. Siaterlis and Maglaris [22] apply DST to DoS anomaly detection. The notion of belief, disbelief, and uncertainty appears

in the work of Jøsang [14]. The paper describes a certification algebra based on a framework for artificial reasoning called *Subjective Logic*.

The literature on sensor fusion is richer with examples of DST application. Several works compare DST to Bayesian inference [5, 10] but they do not consider them in a trust-related context.

7. CONCLUSION

In this work, we developed the notion of data trust. We also addressed ephemeral networks that are very demanding in terms of processing speed. We instantiated our general framework by applying it to vehicular networks that are both highly data-centric and ephemeral. Our approach consists in using the Dempster-Shafer Theory to evaluate data reports with corresponding trust levels. We compare this approach to weighted voting and Bayesian inference. The simulation results show that the local processing approach, based on DST, best suits the decision logic requirements and converges fast enough in a time-critical vehicular network.

8. REFERENCES

[1] http://grouper.ieee.org/groups/scc32/dsrc/.

[2] S. Buchegger and J.-Y. Le Boudec. A robust reputation system for P2P and mobile ad-hoc networks. In *Proceedings of P2PEcon'04*, 2004.

[3] S. Capkun, J.-P. Hubaux, and L. Buttyan. Mobility helps security in ad hoc networks. In *Proceedings of MobiHoc'03*, 2003.

[4] T. M. Chen and V. Venkataramanan. Dempster-Shafer Theory for intrusion detection in ad hoc networks. *IEEE Internet Computing*, 9(6):35–41, Nov.–Dec. 2005.

[5] B. Cobb and P. Shenoy. A comparison of Bayesian and belief function reasoning. *Information Systems Frontiers*, 5(4):345–358, December 2003.

[6] F. Doetzer, L. Fischer, and P. Magiera. VARS: A vehicle ad hoc network reputation system. In *Proceedings of WoWMoM'05*, 2005.

[7] L. Eschenauer, V. D. Gligor, and J. Baras. On trust establishment in mobile ad hoc networks. In *Proceedings of the 10th Int. Security Protocols Workshop*, 2002.

[8] V. Gligor, S. Luan, and J. Pato. On inter-realm authentication in large distributed systems. In *Proceedings of the IEEE Symposium on Security and Privacy'92*, 1992.

[9] P. Golle, D. Greene, and J. Staddon. Detecting and correcting malicious data in VANETs. In *Proceedings of VANET'04*, 2004.

[10] J. Hoffman and R. Murphy. Comparison of Bayesian and Dempster-Shafer Theory for sensing: a practitioner's approach. In *Proceedings of SPIE, Neural and Stochastic Methods in Image and Signal Processing II, Su-Shing Chen; Ed.*, volume 2032, pages 266–279, October 1993.

[11] J.-P. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *Proceedings of MobiHoc'01*, 2001.

[12] T. Jiang and J. S. Baras. Autonomous trust establishment. In *Proceedings of the 2nd International Network Optimization Conference*, 2005.

[13] T. Jiang and J. S. Baras. Trust evaluation in anarchy: A case study on autonomous networks. In *Proceedings of IEEE Infocom'06*, 2006.

[14] A. Jøsang. An algebra for assessing trust in certification chains. In *Proceedings of NDSS'99*, 1999.

[15] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *To appear in Decision Support Systems*, 2007.

[16] R. Kohlas and U. Maurer. Confidence valuation in a public-key infrastructure based on uncertain evidence. In *Proceedings of PKC'00*, volume 1751 of *Lecture Notes in Computer Science*, pages 93–112. Springer-Verlag, 2000.

[17] B. Lampson, M. Abadi, M. Burrows, and E. Wobber. Authentication in distributed systems: theory and practice. *SIGOPS Oper. Syst. Rev.*, 25(5), 1991.

[18] J. Luo and J.-P. Hubaux. A survey of inter-vehicle communication. Technical Report IC/2004/24, EPFL, 2004.

[19] J. Mundinger and J.-Y. Le Boudec. Reputation in self-organized communication systems and beyond. In *Proceedings of Inter-Perf'06 (Invited Paper)*, 2006.

[20] M. Raya, P. Papadimitratos, and J.-P. Hubaux. Securing vehicular communications. *IEEE Wireless Comm. Magazine, Special Issue on Inter-Vehicular Comm.*, 13(5):8–15, 2006.

[21] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[22] C. Siaterlis and B. Maglaris. Towards multisensor data fusion for DoS detection. In *Proceedings of SAC'04*, 2004.

[23] D. Sivia and J. Skilling. *Data Analysis: A Bayesian Tutorial*. Oxford University Press, 2nd edition, 2006.

[24] Y. Sun, Z. Han, W. Yu, and K.J. Ray Liu. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In *Proceedings of IEEE Infocom'06*, 2006.

[25] Y. Sun, W. Yu, Z. Han, and K.J. Ray Liu. Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):305–317, 2006.

[26] G. Theodorakopoulos and J.S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, Feb. 2006.

[27] P. Zimmermann. *The official PGP user's guide*. MIT Press, 1995.

[28] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas. Robust cooperative trust establishment for MANETs. In *Proceedings of SASN '06*, 2006.