MobileTrust: Secure Knowledge Integration in VANETs

GEORGE HATZIVASILIS, FORTH and Hellenic Mediterranean University
OTHONAS SOULTATOS, FORTH and City University of London
SOTIRIS IOANNIDIS, FORTH
GEORGE SPANOUDAKIS, City University of London
VASILIOS KATOS, Bournemouth University
GIORGOS DEMETRIOU, Ecole des Ponts Business School

Vehicular Ad hoc NETworks (VANET) are becoming popular due to the emergence of the Internet of Things and ambient intelligence applications. In such networks, secure resource sharing functionality is accomplished by incorporating trust schemes. Current solutions adopt peer-to-peer technologies that can cover the large operational area. However, these systems fail to capture some inherent properties of VANETs, such as fast and ephemeral interaction, making robust trust evaluation of crowdsourcing challenging. In this article, we propose MobileTrust—a hybrid trust-based system for secure resource sharing in VANETs. The proposal is a breakthrough in centralized trust computing that utilizes cloud and upcoming 5G technologies to provide robust trust establishment with global scalability. The ad hoc communication is energy-efficient and protects the system against threats that are not countered by the current settings. To evaluate its performance and effectiveness, MobileTrust is modelled in the SUMO simulator and tested on the traffic features of the small-size German city of Eichstatt. Similar schemes are implemented in the same platform to provide a fair comparison. Moreover, MobileTrust is deployed on a typical embedded system platform and applied on a real smart car installation for monitoring traffic and road-state parameters of an urban application. The proposed system is developed under the EU-founded THREAT-ARREST project, to provide security, privacy, and trust in an intelligent and energy-aware transportation scenario, bringing closer the vision of sustainable circular economy.

CCS Concepts: • **Security and privacy → Trust frameworks**; • **Networks → Network security**; **Network privacy and anonymity**; **Network mobility**; **Cyber-physical networks**; **Cloud computing**;

Additional Key Words and Phrases: VANET, IoT, CPS, trust, reputation, privacy, mobility, cloud, MEC, parallel computing, GPU, circular economy

This work has received funding from the European Union Horizon's 2020 research and innovation programme H2020-DSSC7-2017 under grant agreements No. 786890 (THREAT-ARREST) and No. 830927 (CONCORDIA), as well as the Marie Skodowska-Curie Actions (MSCA) Research and Innovation Staff Exchange (RISE), H2020-MSCA-RISE-2017, under grant agreements No. 777855 (CE-IoT) and No. 778228 (Ideal-Cities).

Author's addresses: G. Hatzivasilis, FORTH and Hellenic Mediterranean University; email: hatzivas@ics.forth.gr; O. Soultatos, FORTH and City University of London; email: sultatos@ics.forth.gr; S. Ioannidis, FORTH; email: sotiris@ics.forth.gr; G. Spanoudakis, City University of London; email: G.E.Spanoudakis@city.ac.uk; V. Katos, Bournemouth University; email: vkatos@bournemouth.ac.uk; G. Demetriou, Ecole des Ponts Business School; email: g.demetriou@pontsbschool.com.

2378-962X/2020/03-ART33 $15.00
https://doi.org/10.1145/3364181

ACM Reference format:
George Hatzivasilis, Othonas Soultatos, Sotiris Ioannidis, George Spanoudakis, Vasilios Katos, and Giorgos Demetriou. 2020. MobileTrust: Secure Knowledge Integration in VANETs. *ACM Trans. Cyber-Phys. Syst.* 4, 3, Article 33 (March 2020), 25 pages.
https://doi.org/10.1145/3364181

1 INTRODUCTION

In this era of the fourth Industrial revolution, Internet-of-Things (IoT) applications influence our daily activities, with the cyber-physical systems (CPS) market value reaching 8,120 million dollars in 2021 [1]. Vehicular Ad hoc NETworks (VANETs) constitute a special type of mobile networking in this domain [2–5]. They include vehicle-to-vehicle (V2V) and vehicle-to-roadside infrastructure (V2I) communication for providing traffic information, navigation, and safety, among other services [6, 7].

In recent years, several circular economy (CE) initiatives are emerging, penetrating also in the VANET domain. Quality-of-Service (QoS) and energy-aware solutions that serve numerous users and devices enhance the design perspectives towards CE, creating looping assets and maximizing the utilization of resources, like ridesharing [8]. Other interesting proposals for intelligent transportation include smart traffic control, ride service hailing, or even car-sharing.

The USA standard IEEE 1609 Wireless Access in Vehicular Environments (WAVE) is built on IEEE 802.11p [9, 10]. It operates in 5.9 GHz frequency and supports multi-channel communication, security, and lightweight application layer protocols. The relevant European standard ETSI ITS G5 [11] operates in the same band, implementing multi-radio multi-channel functionality, security, and a complex hierarchy of higher-layer protocols that integrate a broad range of main services.

This article focuses on the security aspects of VANET ecosystems. The aforementioned standards cover coarse and generic security services where cryptographic primitives are utilized to accomplish the properties of authentication, authorization, confidentiality, and integrity. However, the VANET problem domain constitutes a field of security challenges arising from the violation of fair use by selfish or malicious participants [12]. Selfish entities may attempt to avoid the additional computational or communication effort from contributing to the collective operations of the social network. Malicious entities exploit the system's vulnerabilities and rating mechanisms to launch attacks, such as Denial of Service (DoS), Byzantine failures, or interception of normal interaction by injecting fake information.

As such, a trust-based computing approach must be adopted [13, 14]. The underlying techniques aim to evaluate the participants' activity to detect security policy violation attempts and in which case countermeasures would be enforced that restrict or ban the involving entities. The goal is to encourage the legitimate nodes in keeping up with their positive contribution, discourage and penalize the selfish and malicious participants from misbehaving, and protect the whole infrastructure from attacks.

This article presents a trust-based resource sharing system for VANETS called MobileTrust. The system is built upon the aforementioned main security functionality (i.e., IEEE 802.11p [10], ETSI ITS G5 [11]) and its main contribution is to evaluate the entities' behavior once the cryptographically secure communication is achieved. As a case study, we considered the scenario where vehicles and smart city infrastructure components exchange information regarding traffic congestion and other road events. A simulation study is conducted in the Simulation of Urban MObility (SUMO)[1] for the evaluation of the main protection and performance properties of our proposal and the

[1] SUMO: http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/.

comparison with relevant systems under the same setting. A real implementation is also deployed on embedded devices and Android smart phones that emulate the sensors and the communication equipment of a smart vehicle.

The remainder of the article is organized as follows: Section 2 discusses the related work in the domain. Section 3 presents MobileTrust. Section 4 describes the theoretical analysis. Section 5 shows the simulation analysis and the comparison of our proposal with the current solutions. Section 6 details the real implementation of MobileTrust on embedded devices. Finally, Section 7 concludes and refers future work.

2 RELATED WORK

In the context of VANETs, trust-based interactions occur at high speeds and short time periods [13]. The knowledge integration algorithm should exhibit low complexity constraints, such as effective throughput, fast memory access, and cost-efficient processing. Scalability is a main concern, as incidents or road events could be reported either by single mobile nodes or by multiple vehicles.

General attacks on mobile ad hoc networks are surveyed in References [15] and [16]. Furthermore, VANETs are vulnerable to false positioning, message modification, false message sending, and DoS [14]. Trust-based schemes are deployed as a main countermeasure against most of these threats. Inevitably, the formal trust evaluation processes are targeted by more sophisticated attackers [17]. The involved challenges of trust management in VANETs are reviewed in References [18] and [19].

2.1 Trust-based Resource Management

Several trust-based schemes have been proposed in the literature attempting to tackle resource management. For this study, we focus on these systems that collect and process information in real-time for a VANET setting.

The Vehicle Ad hoc network Reputation System (VARS) [20] is one of the first attempts to incorporate trust-based computing in the field of VANETs. It forms a modular and peer-to-peer (P2P) reputation system that separates direct and indirect information when integrating knowledge. The system rests its confidence on the distributed content itself rather on the node behavior. The forwarding nodes attach in the transmitted message their opinion about the content (e.g., traffic congestion on a road segment or blocked route). The receiver utilizes the contextualized data to evaluate the trustworthiness of the message. A node's opinion about an event can be formed by direct knowledge, indirect knowledge sent by a known sender, partially by the attached opinions on the message by the forwarding nodes, or by a combination of all these interactions. A main performance drawback of VARS is the fact that the communication bandwidth is substantially affected due to the overheads from the information that is cumulatively appended along the communication path when a high number of forwarding nodes is involved. Furthermore, the proposed approach does not enforce authentication, thus making the system vulnerable to attacks from simple modification to complete deletion of messages.

The Event-based Reputation System (ERS) [21] attempts to detect inaccurate traffic events and deter attackers from spreading false messages. Traffic information is collected from the vehicles and the road infrastructure. The system utilizes distributed observations and tries to figure out if a traffic event really exits and how long it lasts. The information of an event is maintained in a table by every vehicle that becomes aware of the incident, either by direct interaction or indirect notifications. The event table stores data regarding the event's identity, type, occurrence, location, message transmission range, reputation, and indirect confidence list. One main scalability problem that arises is the difficulty in managing the confidence lists for all events that are reported in the network. Also, the simple evaluation process of indirect notifications makes the system vulnerable to attackers that perform badmouthing (advertise bad recommendations for legitimate nodes) and ballot-stuffing (gain good reports from colluding malicious entities) [17]. Moreover, as ERS requires a significant amount of contributions to designate that an event has occurred, it fails when it comes to the early and timely notification of drivers regarding the existence of an accident on the road ahead. However, the safety of drivers is an essential goal towards the real deployment of VANETs [18, 19].

The Trust Based Security Enhancement (TBSE) for VANETs [22] is designed upon the core cryptographic techniques and acts as a second defense mechanism against inside attackers. Both direct and indirect knowledge are utilized for the trust estimations. Direct interaction is evaluated based on the Bayesian rule. For the rating of the third-party recommendations, the Dempster-Shafer Theory (DST) [23] is employed, allowing the probabilistic assessment of the truthfulness of an event in the presence of uncertainty. Then, the calculated trust values are stored in a repository where upper-layer applications can gain access and utilize this information to achieve their security goals, such as secure resource sharing or routing. However, the attacker model is restricted to a low number of malicious entities that do not collude with each other. Passive attacks are also not studied. Moreover, TBSE is vulnerable to information cascading and oversampling [24]. In cascading, a decision is made by the nodes deciding sequentially after observing the behavior of the other participants. Therefore, a node's choice is highly influenced by the previous decisions that can overrule its own observation. Oversampling occurs when a node receives the opinion of nodes *i* and *j*, but *j*'s opinion has been initially influenced by the decision of *i*. Thus, the contributing information by *j* is oversampled.

A Social Network approach to Trust Management (SNTM) in VANETs is presented in Reference [24]. The main contribution of this study is the limitation of the bad effects from information cascading and oversampling in P2P communication. SNTM tackles cascading directly by assigning higher voting weight to the vehicles that are closer to the event. As the influence from cascading is eliminated and the result becomes more accurate, the potential oversampling is also indirectly limited. The nodes also attach their opinion regarding an event in the message. The receivers contrast this knowledge with their own opinion. Then, they rate the other contributors and estimate the message's quality.

2.2 Discussion

Centralized trust evaluation was not considered an appropriate architecture for VANETs by the research community due to scalability issues attributed to the high computational and communicational burden on a central entity as well as dependability concerns, as this central entity would be a single point of failure [24]. The design of P2P architectures seemed as a one-way approach towards the establishment of mobile trust. All the aforementioned related works promote the P2P paradigm. However, to build effective and trustworthy P2P relationships, one would need a significant and frequent amount of interactions between the peers. Thus, these studies fail to capture some inherent properties of VANETs. The ephemeral nature of such networks does not veritably facilitate the P2P trust. Moreover, trust is not necessarily transitive, and the gathering of information from past interaction, by each peer for every other participant, is computationally expensive and even impossible under realistic assumptions. As the vehicles drive at high speeds, when the selection and processing of the proper knowledge for an event are completed, the outcome is no longer required, as the vehicle has either faced, dealt with, or bypassed the reported problem. P2P trust will inevitably become vulnerable to information cascading and oversampling in an actual setting.

Authentication and cryptographic security is another important issue. A number of studies (e.g., References [20, 21]) ignored or overlooked the implications of cryptographic communication and the overheads from supporting this essential functionality in P2P networks. However, when peer

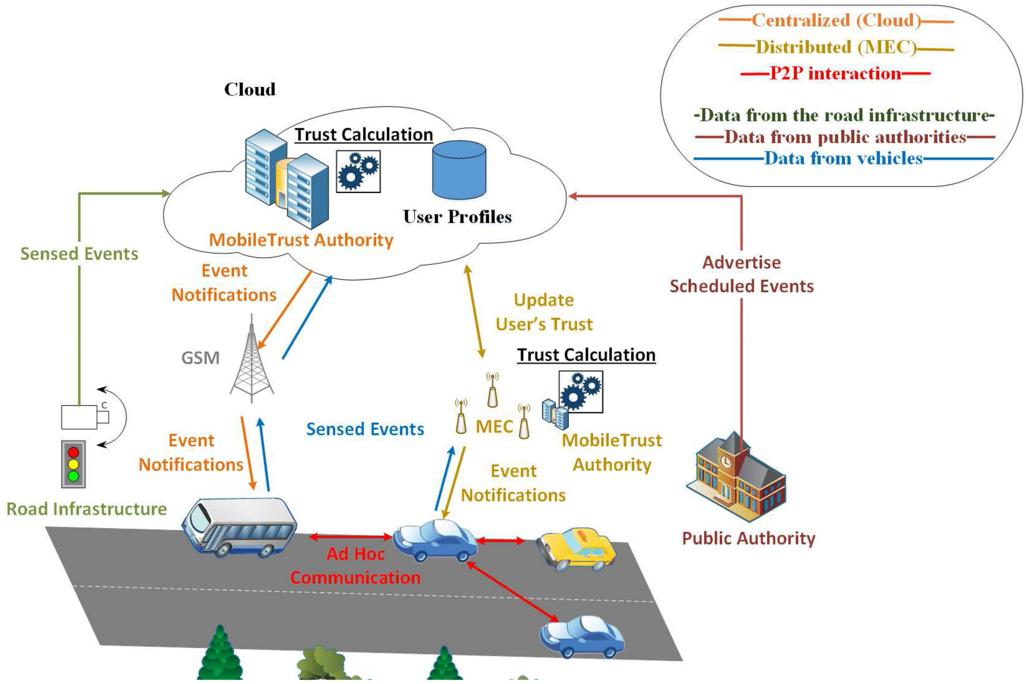

Fig. 1. The MobileTrust design approach.

authentication is implemented, privacy issues arise, as a vehicle publishes data regarding its driving route or other user sensitive information [25].

The evolution of cloud computing and upcoming 5G technologies, like Multi-access Edge Computing (MEC) [26, 27], can help towards overcoming the restrictions of both centralized and P2P trust in the VANETs domain. This study proposes a hybrid trust algorithm where both types of interaction are utilized. Theoretical results regarding the scalability of hybrid P2P systems for data distribution are reported in Reference [28]. The core trust calculation and maintenance is performed in a centralized manner, focusing equally on the data correctness and the evaluation of the nodes' behavior. Nevertheless, the obstacles of centralized knowledge integration are overcome by distributing and parallelizing the related computations in the cloud and MEC. The internal computations are further parallelized in Graphics Processing Units (GPUs), enhancing the overall performance.

Authenticated users contribute with first-hand observations. Only the central authority integrates this knowledge and reports the existing road events, eliminating the bad effects of cascading and oversampling. To decrease the centralized communication, the peers can broadcast this timely and spatially sensitive data that are digitally signed by the authority without requiring to authenticate each other and, thus, preserve their privacy between the vehicle nodes. Figure 1 illustrates the proposed setting. In short, the proposed MobileTrust system aims to offer:

- Efficient centralized knowledge integration, exploiting the capabilities of cloud computing, MEC, and GPUs;
- Effective trust computation of authenticated users in a centralized manner, capturing the inheriting properties of VANETs (e.g., ephemeral environment, time-sensitive information, different sensing capabilities, and safety response);

- High accuracy in estimating the presence of a road event, even in the presence of malicious entities;
- Preservation of P2P privacy;
- Secure cryptographic communication in all transactions;
- DoS resilience.

The overall setting provides efficient, effective, and timely accurate information, countering a high variety of attacks and offering secure and privacy-preserving communication.

3 MOBILETRUST

This section details the MobileTrust design. The system integrates first-hand information from the involving participants. The trustworthiness of each contributing node is taken into consideration during this process. The contributors are authenticated while privacy is preserved in the P2P interaction. The overall setting presupposes that there exists a safe procedure to register/identify the participating vehicles to prevent the attackers from introducing virtual vehicles that transmit bogus data. The potential towards global scalability is also presented. The assignment of the constant coefficients for the reputation and trust computations (e.g., the reputation rating values or the trust integrating weights) is based on relevant studies [20–22, 24] and in a previous analysis that the authors have conducted under a similar setting [37].

3.1 Knowledge Integration

Three reputation resources are considered by direct interaction, notices sent by trustworthy public authorities, and notifications made by a centralized entity. The following event types (E) are modelled: (i) Road accidents, (ii) Traffic congestion, (iii) Bad road conditions, (iv) Working activities, (v) Protest, ceremony, military, religious, athletic, or other march, (vi) Route diversion by police, (vii) Blocked road (i.e., by physical obstacle), (viii) General alarm to draw the drivers' attention and be cautious.

When a major event is reported, the nodes that are located closer and reckoned sooner can contribute with data having a higher level of accuracy. These advantageous geo location reporters gain higher reputation values if the transmitted data are confirmed.

Based on the evaluation result, a message can be rejected, accepted but not integrated, or accepted and integrated. Messages originating from untrusted nodes or those who report events that are not close to the nodes' location are rejected and the involving nodes are rated negatively.

For the traffic report application, the vehicles must collect information regarding their speed. The road routes are segmented, for example, every 500 m. A vehicle calculates its average speed in each crossed segment. If the speed falls under a threshold, the vehicle indicates the segment as congested. When a decent amount of similar indications are received by the backend component (see the text below), the application reports the traffic congestion. If there are no incoming indications, the congestion report for a segment fades as time elapses until it is assigned as normal again. MobileTrust bounds the misbehavior of a malicious vehicle from arbitrarily reporting segments as congested. The attacker will be rated negatively as long as his indications will not be supported by other participants.

The events of working activities, protest or other marches, route diversion by the police, blocked road segment, or general alarm are manually designated by the driver. Moreover, the backend application can be informed in advance for such events by the involving authorities, which are considered trustworthy. For example, the municipality can report the scheduled working activities or athletic events, which are then imported into the system.

The centralized trust management component performs the following steps:

1. The first notification of an event $e \in E$ on a road section *rs* creates a new event entry with neutral confidence (the value depends on the node's overall trustworthiness and its reputation in observing the specific event type).
2. As other nodes located in the same road section advocate that the event is ongoing (by sending the same notification type as in step 1), the confidence that the event is actually happening increases.
3. The confidence $C_{rs,e}$ is calculated as the ratio of the contributing nodes ($contr_{rs,e}$) to the nodes that enter the road section ($passing_{rs}$). It is defined in Equation (1) as:

$$C_{rs,e} = \frac{contr_{rs,e}}{passing_{rs}} \Rightarrow \left\{ \begin{array}{cc} [0\% - 50\%], & low \\ (50\% - 70\%], & moderate \\ (70\% - 100\%], & high \end{array} \right\} \tag{1}$$

4. Incoming vehicles are notified about the active events in advance, before entering the road section. The reported event's confidence fades as time elapses and no new notifications are made, to designate that the event is smoothing out (i.e., *10%* decrement for every minute that passes while the incoming vehicles do not notice the event).
5. The event entry is terminated after a fixed-time period (i.e., *5* minutes) to prevent the case where an attacker sends the same notification over-and-over to keep the entry active.
6. Then, the nodes that cross the road section are evaluated. The nodes gain a reputation rate according to the events maximum level of confidence that was achieved during the active period.
   a. The *contributing nodes* gain negative (*−4*), neutral (*0*), and positive (*+1*) reputation values *r* for low, moderate, and high level of confidence, respectively. In the last case, the first responders (*2%* of the total contributors) are awarded with an extra bonus score (*+1*).
   b. Similarly, the *non-contributing nodes* are rated negatively (*−4*) if the event gained moderate or high confidence (based on Equation (1)) to detect selfish nodes; otherwise, they are ranked neutrally (*0*).
   c. Also, the system ranks negatively (−4) the *nodes that remain constantly in a road segment* and report events (i.e., traffic jam) while the majority of the entering nodes continue their route, in an attempt to force the malicious nodes to be in continuous movement and be unable to target distinct road segments.
7. If notifications are sent after the active period, a new event entry is created (step 1).

3.2 The Trust Framework

For every user, MobileTrust evaluates his/her trustworthiness based on the past interaction. All new users start with neutral trust and reputation values (*0*), which are continuously updated as new pieces of knowledge are integrated.

MobileTrust maintains a small history of the latest evaluation results for each event type. The history size (h_e) is determined by the event's frequency, ranging from *500* ranks for traffic monitoring to *30* ranks for the user-defined reports (e.g., blocked road segments or protest march). Reputation fading is applied for each event type, based on a beta distribution [29], by assigning higher weights ($fading\_w_i$) to the most recent interactions. The reputation value of the user for a specific event type, $R_{(User,\ e)}$, is the average weighted summation of the relevant historical values, as defined in Equation (2):

$$R_{(User,e)} = \sum_{i=0}^{h_e} r_i \times fading\_w_i / h_e. \tag{2}$$

This information determines the effectiveness of a user in reporting a specific type of event. Thus, the system can discriminate the different sensing capabilities of the various nodes and exploit this knowledge (if the reputation value is, for example, >0.5), especially when an event has just been reported or in road segments with few contributors. Moreover, the system can detect a malfunction in the sensing equipment or a malicious activity that tries to manipulate the rating mechanism for a specific event type (i.e., when the reputation value decreases beyond a threshold of *0.4*) and not integrate the relevant contributions.

Trust estimates the overall cooperativeness of a user. Each event reputation gains a weight (w_e) based on the event's frequency and severity. Traffic monitoring and road accident reports are assigned with a weight of *0.30* and the rest types are assigned with a weight of *0.1* (the weights summation is equal to *1.0*). The trustworthiness for a user, T_{User}, is then calculated by aggregating the underlying reputation values, as defined in Equation (3):

$$T_{User} = \sum_{e=0}^{Totaleventtypes=8} R_{(User,e)} \times w_e. \tag{3}$$

A user with a trust level that exceeds *0.8* or *0.6* is considered trusted and legitimate, respectively. When the trust falls beyond a threshold (i.e., ≤ 0.5), the user is notified for his/her misbehavior and is categorized as suspicious. If the cooperation level keeps degrading (i.e., ≤ 0.4), the node is finally reported as malicious and is expelled from the network.

3.3 Authentication and Privacy

To maintain the reputation records and operate in a trustworthy manner, the users that contribute information to MobileTrust must be authenticated. Each driver creates an account with a username/password pair (or other means of authentication, such as near-field communication (NFC), Bluetooth pairing, etc.) and logs on the system through the vehicle's infotainment system (SSL/TLS communication). Only authenticated users can send information to the application through a secure channel. The data are encrypted with a symmetric session key (such as AES) and through an appropriate protocol such as SSL/TLS.

Furthermore, passive anonymous use is also supported. The notifications that are reported by MobileTrust are signed with the authority's private asymmetric key (RSA). To reserve resources, all active notifications for a road segment are included in the signed message. Thus, every node can verify the central authority, as its public asymmetric key is known to all participants. However, it should be stated that a high volume of anonymous users who do not contribute any information could degrade the effectiveness of the system in the presence of attackers (as detailed in the theoretical analysis subsection 4.2).

A main contribution of the overall setting is the privacy protection of the V2V interaction. To preserve privacy, the nodes exchange only the publicly signed notifications. Thus, if a vehicle requests information regarding some road segments, the other vehicles or the road infrastructure will respond with the relevant messages that have been previously retrieved by the central authority without requiring to authenticate each other. Both the request and the response are broadcasted messages in a P2P communication of the nearby nodes.

The notification format, as described in Equation (4), consists of the road segment's ID, the event type, the date (timestamp) where the event occurred, and a sequence number (bounded to a specific segment ID and event type and initialized every day when the date is changed). This information is both included in the encrypted data and also sent in plaintext along with the rest of the message. To preserve resources, the nodes maintain only the latest notifications of each event type for a road segment. Moreover, a node deletes the older messages of passed segments as time elapses and the events' active periods terminate (i.e., after *5* minutes) to keep the memory usage profile low.

$$\begin{gathered}\text{Notification format:}\\ [\text{Road segment ID}|\ \text{Event type}\ |\text{Date}|\text{Sequence no.}]\end{gathered} \tag{4}$$

The plaintext notification identifier is utilized as an automatic access control tag to degrade the effectiveness of DoS attacks at the responder and requester ends. A request message is unencrypted and contains road segment IDs. The responder examines the included road segment IDs and broadcasts the latest related notifications. If irrelevant segments are included, the node discards the message and does not proceed in any further actions. The node also adheres to a threshold for a maximum number of request responds in the current time window (i.e., at most 10 responds per minute) as an additional measure against DoS by bogus request messages.

A response message replays the public notifications as they were retrieved by the central authority. For each incoming response, the requester examines the plaintext notification identifier. The node automatically recognizes the relevant notifications by examining the road segment ID and the timestamp. Then, it checks the sequence number and evaluates the newest notification version. The node decrypts the message with the authority's public key. If the decryption and the integrity check (SHA512) are successful, the node integrates the notification; otherwise, the node proceeds to the older received versions of the notification. However, an attacker can still send erroneous messages with a valid notification identifier and a high sequence number to enforce the requester to perform the decryption operation. Thus, if several decryption attempts fail in a time interval (e.g., five fails in the last minute), the node turns in back-off mode for a while, until the vehicle drives away from the region. The user can later log in to the system and report the road segments where the malicious activity was observed.

3.4 Vehicle Registration

Except from authenticating the active users through the proposed application, we must also enforce a procedure to examine that he/she is driving a real vehicle; otherwise, a malicious entity could create several user accounts for virtual vehicles, which would later launch coordinated sophisticated attacks online by sending fake positioning information along with erroneous road events. This is a general problem that affects all VANET settings (e.g., References [20–22, 24]). An indicative case is the money-laundering scam in Uber.[2] Collaborating drivers log in to the system through a virtual machine (VM) that emulates a fake-GPS service. Users/payers find these taxicabs via specific marketplaces in the dark web and hire services that they do not actually use, laundering money in the process.

Identification and authentication of a vehicle is out of the scope of this study, and MobileTrust presupposes that the smart vehicle industry will deploy such an identification mechanism. Nonetheless, a basic protection approach is sketched. Each manufacturer nowadays assigns a unique identifier for each produced vehicle, known as the Vehicle Identification Number (VIN).[3] The VIN is composed of 17 digits and acts as the vehicle's fingerprint. For the modern smart functionality, the manufacturer (or another trusted party) can sign and provide the vehicle owner with a digital certificate. The certificate will contain the VIN and other vehicle-related information. Then, MobileTrust or other smart applications can obtain this data through the user's equipment and check the existence of the mobile node. The application can also interact with the signing entity to evaluate the certification's validity (e.g., if it has been revoked). Except from VIN, if the

[2] Uber money-laundering scam: https://bdtechtalks.com/2018/03/21/fraudsters-uber-money-laundering/.
[3] Vehicle Identification Number (VIN): https://en.wikipedia.org/wiki/Vehicle_identification_number.

vehicle is equipped with a modern Electronic License Plate (ELP),[4] the related data could also be utilized for the same purpose.

The overall process can be further enhanced with the analysis that is performed by popular Internet services and social media (i.e., Gmail and Facebook). For security purposes, they can record the set of devices with which the user usually logs onto the system and try to keep out intruders that have disclosed the user's credentials.

3.5 Towards Global Scalability

Centralized architectures were considered, in general, disadvantageous in comparison to P2P settings, as they constitute a single point of processing and a bottleneck [13, 24]. However, cloud and fog computing is now providing several solutions regarding efficiency and dependability [30].

The component of MobileTrust that performs all the computationally intensive operations is the central authority. Nevertheless, VANETs and the proposed trust algorithm exhibit some inherited design properties that enable distribution, parallelization, and global scalability. Firs, each authenticated user/driver is located in a specific region at each time point, and, usually, drives around a particular geographic location most of the time. Second, the processed data are region dependent. Thus, the knowledge integration of each road segment can be parallelized in a fine-grained manner.

The IT industry that provides cloud services installs several datacenters in each continent around the globe. The application traffic is routed to the nearest datacenter based on the IP address. Load balancing among the datacenters is delivered by the cloud provider [31]. Thus, the traffic from a country is transmitted to specific datacenters. The user data can also be maintained in an efficient and distributed manner in the cloud, similarly to an email service, a social network, or other global-scale applications [32].

The MobileTrust data-gathering and integration algorithm can be further parallelized based on the road segments. Each hired parallel processing unit in a datacenter can perform the trust computations of a single or a group of nearby road segments. The processing units can read data from a pool and serve the incoming traffic completely independent from each other. Each core processes data buckets, where each bucket contains the information for a distinct event entry. As mentioned earlier, the event entries in a road segment remain active for a fixed-time period; for example, *5* minutes. Then, the core updates the trust profiles of the contributing users. GPU-processing can decrease the overall processing time.

Additionally, MEC technologies [26] can further reduce the communication delay. The parallel processing can be installed closer to the smart road infrastructure and drastically reduce the round-trip time (Figure 1).

4 THEORETICAL ANALYSIS

This section details the theoretical analysis of the proposed approach and its effectiveness in countering the attacker models that are detailed in the simulation study (Section 5). The overall analysis is based on the ordinary assumption that the mainstream cryptographic primitives (i.e., TLS, RSA, AES, and SHA512) are implemented and configured properly, and thus, their usage is considered theoretically secure. At first, we examine the security aspects that are provided by the deployed communication protocols. Then, we give proofs regarding the capabilities of MobileTrust in supporting the legitimate functionality in presence of attackers.

[4] Electronic License Plate (ELP): https://en.wikipedia.org/wiki/Electronic_license_plate.

4.1 Protocol Analysis

The theoretical security analysis of the communication links between the involved entities and the MobileTrust component is modelled in the verification tool ProVerif [33] (the code is not included for brevity). It is a widely used automatic symbolic protocol verifier that proves the security properties of the examined protocol, such as authentication, secrecy, and adversary equivalence aspects. The examined protocol is modelled in a process calculus and is automatically translated in Horn clauses [33]. The tool resolves these clauses and determines if the security properties hold or not. In the case where all properties are validated, ProVerif returns “true.” Otherwise, it outputs the properties that could not be satisfied. Three communication protocols are examined:

1. The user/vehicle login and knowledge transfer towards the Central Authority (CA);
2. Road segment notification requests from a vehicle to the CA;
3. Road segment notification requests from a vehicle to another vehicle (P2P).

4.1.1 User Log-in and Contribution. To actively participate in the MobileTrust community, the driver has to authenticate him/herself and enter the system. This includes an ordinary log-in process where the user inputs a pair of username/password through a SSL/TLS connection via HTTPS [34]. The CA safely receives the data and verifies the user.

The communication is initiated by the user. The user application and the CA perform the *TLS handshake phase*. The application obtains the CA’s digital certificate (X.509) and validates it. The two entities agree upon a randomly generated symmetric session key.

From this point on, the transmitted data are encrypted with the common key as in the *TLS record phase*. The user is prompted to provide his/her data (e.g., Reference [35]). The log-in request contains the user’s credentials (e.g., user name, password’s digest). The CA decrypts it, checks the message’s integrity (HMAC-SHA256) and uniqueness, and contrasts the received information with the credentials that are maintained in an internal database. If the process is successful, the user is authenticated and can upload the evidence for ongoing road events. Then, the CA performs MobileTrust and evaluates the user’s cooperation. Figure 2 depicts the communication steps.

The two entities began with the establishment of the secure channel (*HTTPS/TLS handshake*). Then, they utilize the derived session key and encrypt the rest exchanged messages (*TLS record*). ProVerif evaluates the abovementioned protocol steps (TLS phases and the transmission of the sensed data from the user to the CA) and validates that the overall setting offers authentication, confidentiality, integrity, and immunity to replay attacks.

4.1.2 Notification Requests. Every user can ask the CA for the active event notification list of specific neighboring road segments. If the user is logged in, then the communication can be encrypted by the session key. Otherwise, the user can make unauthenticated requests and retain his/her privacy.

The request contains the required road segment IDs and is encrypted with the CA’s public key. The digest of the data is also computed (SHA512), and the message is sent to the CA. The CA verifies the message’s integrity. The response contains the relevant notification list and a random nonce. The data digest is computed and then signed with CA’s private key. The user re-computes the digest of the received data and decrypts the signature with the CA’s public key. Figure 3 illustrates the exchanged messages.

ProVerif validates that the communication is safe and provides CA authentication, integrity, and immunity to replay attacks. As described in the previous sections, full protection is not required here, as the road event notifications are meant to be publicly available.

4.1.3 P2P Interaction. Nevertheless, a vehicle can request information about the neighboring road segments by the nearby vehicles. The requester sends the segment ID list in plaintext along

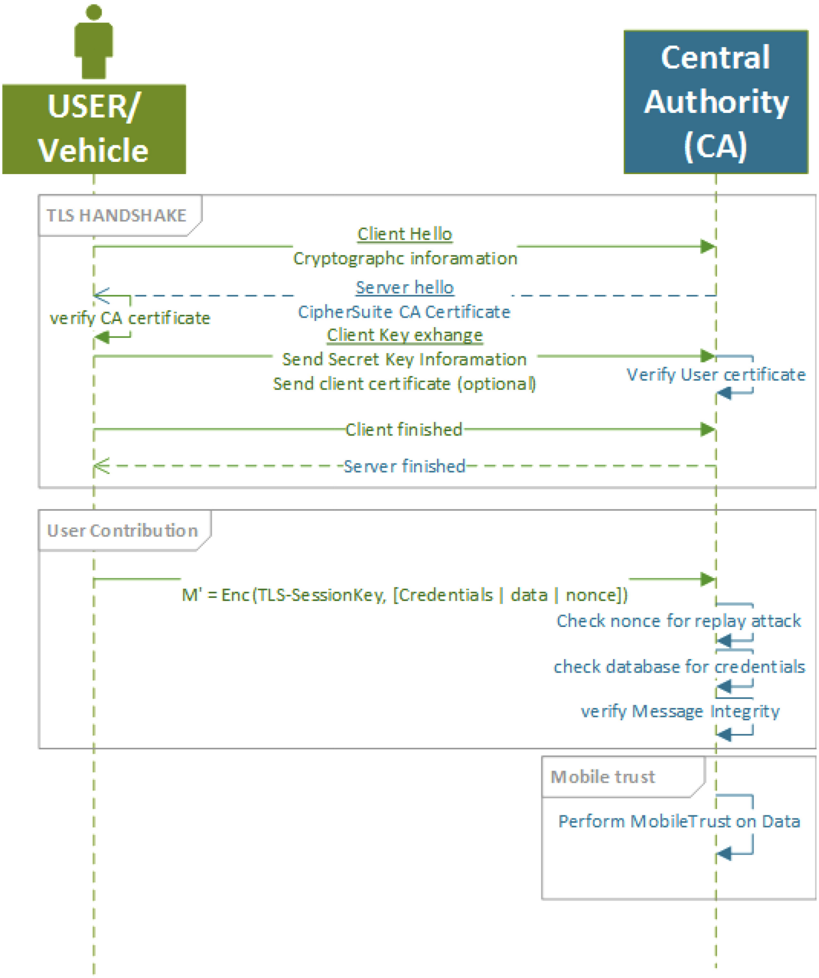
USER/ Vehicle
Central Authority (CA)
TLS HANDSHAKE
Client Hello
Cryptographc inforamation
Server hello
verify CA certificate
CipherSuite CA Certificate
Client Key exhange
Send Secret Key Inforamation
Send client certificate (optional)
Verify User certificate
Client finished
Server finished
User Contribution
M' = Enc(TLS-SessionKey, [Credentials | data | nonce])
Check nonce for replay attack
check database for credentials
verify Message Integrity
Mobile trust
Perform MobileTrust on Data

Fig. 2. User's log-in and knowledge contribution.

with the message's digest. The contributors check the message's integrity and respond with the related active notifications that they possess. The requester verifies the received notifications' legitimacy as in the previous case. The protocol's phases are similar with the relevant steps that are depicted in the second part of Figure 3 for the anonymous communication between the user and the CA, with the initial request being an unencrypted broadcast message.

As before, the tool verifies that the interaction accomplishes the CA authentication, integrity, and immunity to replay attacks. Moreover, the contributors' privacy is retained as no user-/vehicle-related data are disclosed.

4.2 Attack Mitigation

The theoretical security analysis of the trust computing mechanisms and its effectiveness in constraining the malicious activity is performed in two steps. First, we prove that the integration of malicious knowledge is bounded (*Theorem 4.1*). Then, we argue that MobileTrust's efficacy is strongly related with the active participation of the legitimate users (*Lemma 4.1* and *Theorem 4.2*).

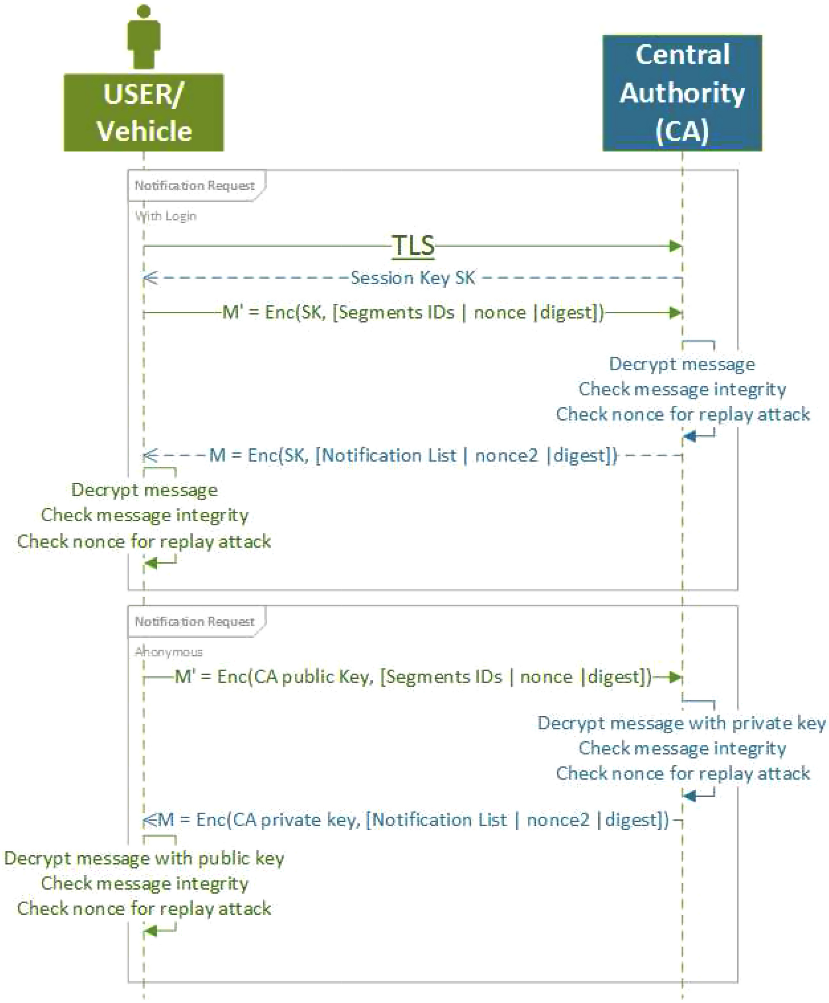
USER/ Vehicle
Central Authority (CA)
Notification Request
With Login
TLS
Session Key SK
M' = Enc(SK, [Segments IDs | nonce |digest])
Decrypt message
Check message integrity
Check nonce for replay attack
M = Enc(SK, [Notification List | nonce2 |digest])
Decrypt message
Check message integrity
Check nonce for replay attack
Notification Request
Anonymous
M' = Enc(CA public Key, [Segments IDs | nonce |digest])
Decrypt message with private key
Check message integrity
Check nonce for replay attack
M = Enc(CA private key, [Notification List | nonce2 |digest])
Decrypt message with public key
Check message integrity
Check nonce for replay attack

Fig. 3. Notification requests and anonymous use.

The proposed defense mechanism increases significantly the effort that must be devoted by the attacker to exploit the system (*Lemma* 4.2). Finally, we evince that the system can provide protection even without the full participation of all legitimate users that may want to retain their privacy (*Lemma* 4.3).

For the *system model*, we form a VANET with N nodes, such as $V = \{1, 2, \ldots, N\}$. It is assumed that all links are bidirectional. If node i can receive packets that are directly transmitted by node j, then node j can receive packets that are directly transmitted by node i. The nodes are either vehicles or smart road infrastructure. Since the majority of these nodes are vehicles, the network dynamics are characterized by confined mobility, high speeds, and thus, short connection times between the nodes. The operational area can be very large, but the V2V communications are mainly local, and the currently processed information for each node concerns specific road segments and not the whole network. Information regarding the examined road events is collected and exchanged by each node, either with other nodes or with the CA. Vehicles can travel freely, but in most cases, they have a relatively fixed route, such as buses.

Due to the complexity of a VANET system, several types of attacks can be performed. For the *adversary model*, we consider that attackers are internal nodes (malicious or compromised) that send out fraudulent event messages: They notify unreal events (i.e., traffic jam while the road segment is not congested) and if an event exists, they do not send any message to the authority denoting the incident (mislead the trust mechanism into producing false negatives in the case where a smaller number of legitimate nodes have reported the events). They are global attackers (have an overall coverage of the network and can eavesdrop every message diffused by any vehicle), who can be either active or passive. Legitimate nodes can also malfunction with a low probability and perform the same actions for a while (i.e., Byzantine faults [36]).

We start by providing some definitions and then the theorems and the proofs.

Definition 3.1. Let $contr^{+}{}_{rs,e}$ be the total number of correct event notifications that are reported to the CA for a specific event *e* in a road segment *rs*.

Definition 3.2. Let $contr^{-}{}_{rs,e}$ be the total number of erroneous event notifications that are reported to the CA for a specific event in a road segment.

Definition 3.3. Let $Tcontr_{rs,e}$ be the total number of received event notifications for a specific event in a road segment, determined as the sum of $contr^{+}{}_{rs,e}$ and $contr^{-}{}_{rs,e}$.

Definition 3.4. Let *p* be the success rate of the total received event notifications $Tcontr_{rs,e}$.

4.2.1 The Bounded Malicious Effect.

Theorem 4.1. *As the contributing users are authenticated, they can send information for a specific region at each time point. The ideal network exhibits* $\sum_{rs\in RS,e\in E} contr^{-}{}_{rs,e} - p \cdot \sum_{rs\in RS,e\in E} contr^{+}{}_{rs,e} \leq 0$. *For up to an additive constant, ignoring a bounded number φ of erroneous notifications, it holds that the number of erroneous notifications is a* p*-fraction of the number of received notifications. Specifically, there exists an upper bound φ, as described in Equation (5):*

$$\sum_{rs\in RS,e\in E} contr^{-}{}_{rs,e} - p \cdot \sum_{rs\in RS,e\in E} contr^{+}{}_{rs,e} \leq \varphi. \tag{5}$$

Proof. Suppose that there exist *N* nodes in the network *V*, *M* of which are malicious with $M < N$. Let *MRS* denote the set of road segments *RS* that contain malicious nodes. The maximum value for *MRS* is *RS* (if there is at least one *M* in every $rs \in RS$).

Let β denote the number of erroneous notifications that exposes a node as malicious. The number of convictions $conv_{rs}$ is at least $\sum_{rs\in MRS,e\in E} contr^{-}{}_{rs,e}/\beta$. Thus,

$$\frac{\sum_{rs\in MRS,e\in E} contr^{-}{}_{rs,e}}{\beta} - \sum_{rs\in MRS} conv_{rs} < 0. \tag{6}$$

Similarly, the number of rewards rew_{rs} is at most $\frac{\sum_{rs\in MRS,e\in E} contr^{+}{}_{rs,e}}{\beta/p}$. Thus,

$$\sum_{rs\in MRS} rew_{rs} - \frac{\sum_{rs\in MRS,e\in E} contr^{+}{}_{rs,e}}{\beta/p} < 0. \tag{7}$$

Therefore,

$$\frac{\sum_{rs\in MRS,e\in E} contr^{-}{}_{rs,e}}{\beta} - \frac{\sum_{rs\in MRS,e\in E} contr^{+}{}_{rs,e}}{\beta/p} \leq \sum_{rs\in MRS} (conv_{rs} - rew_{rs}). \tag{8}$$

By combining Equations (5) and (8), we derive:

$$\sum_{rs \in MRS, e \in E} contr^{-}{}_{rs,e} - p \cdot \sum_{rs \in MRS, e \in E} contr^{+}{}_{rs,e} \leq \beta \sum_{rs \in MRS} (conv_{rs} - rew_{rs}) \leq \beta \cdot M \cdot MRS. \quad (9)$$

Therefore, the malicious activity that affects the network is bounded. If there are no malicious nodes (M = *0*), then Equation (9) describes the ideal case of $\sum_{rs \in RS, e \in E} contr^{-}{}_{rs,e} - p \cdot \sum_{rs \in RS, e \in E} contr^{+}{}_{rs,e} \leq 0$. □

4.2.2 Correlation between the Legitimate User's Participation and the Robust Deduction Procedure of MobileTrust.

Lemma 4.1. *The MobileTrust's evaluation mechanism can decrease the attack rate.*

Proof. Consider M_{up} and M_{mt} as the number of malicious nodes per window for an unprotected (up) network and a setting that is protected by MobileTrust (mt), respectively. As the malfunctioning is detected with MobileTrust in place, the malicious nodes (M_{mt}) are denoted as untrustworthy, punished, and eventually expelled from the network based on Equation (3). Thus, it holds that $M_{mt} = M - \sum(User,\ where\ T_{User} \leq 0.4)$. Based on Equation (9), it is concluded that the malfunctioning that is caused by the malicious nodes for an unprotected setting, where $M_{up} = M$, can be higher, as it holds that $M_{up} \geq M_{mt}$. □

Theorem 4.2. *The MobileTrust's mitigation rate is directly related with the user's active participation in the crowdsourcing operations.*

Proof. Let $m_{rs} \subseteq M$ denote the malicious nodes that exist in a road segment $rs \in MRS$. The maximum volume of malicious notifications for a specific event entry e in rs is $m_{rs,e} = m_{rs}$.

Consider a faulty event *e* for a *rs*. The event's confidence $C_{rs,e}$ is given by Equation (10):

$$C_{rs,e} = \frac{contr_{rs,e}}{passing_{rs}} = \frac{p \cdot contr^{+}{}_{rs,e} - (1-p) \cdot contr^{+}{}_{rs,e} - m_{rs,e}}{passing_{rs}}. \quad (10)$$

Based on Equation (10), the malicious activity will be punished if the erroneous notifications are below *50%*; thus, if$((1-p) \cdot contr^{+}{}_{rs,e} + m_{rs,e}) \leq 0.5 \cdot passing_{rs}$. As the malfunction ratio *(1 − p)* of the legitimate entities will be low, the higher the user participation, the higher the difficulty and the effort for the attacker to avoid punishment. □

On the contrary, this also reflects the general limitation of the recommendation systems [37, 38]. If the majority of the nodes are compromised (≈*50%*), such approaches cannot provide guarantees against colluding malicious attacks. Networks with such high volume of attackers are in general discarded [20–22, 24, 34].

4.2.3 Attacker's Effort and User's Participation.

Lemma 4.2. *MobileTrust increases the effort that is required to exploit the crowdsourcing system.*

Proof. Based on Theorem *4.2*, it is concluded that to perform an attack, one would need to marshal a high portion of colluding malicious user-accounts that cross a $rs \in MRS$. As MobileTrust forces the colluding nodes to move around to avoid detection (step 6.c in Section 3.1), the attacker cannot know in advance the $passing_{rs}$ nodes from the *rs*, and thus, he/she could not know if the malfunctioning will be detected by the CA during the evaluation of an event entry or not. Therefore, the attacker could be burdened with a number of failed notifications. This fact will make the attacker devote a significant effort to ensure that this single *rs* will be mistreated, straining the wile resources. However, the falsely accused legitimate users can retain their trustworthiness by their rewards from non-faulty segments. □

Lemma 4.3. *MobileTrust does not require the absolute cooperation from all users to report events with high confidence and* effectively *mitigate the attackers.*

Proof. Taking into account privacy issues and the user's convenience, MobileTrust preconceives that users will not always be *participating* actively in the crowdsourcing activities. From Theorem *4.2* it is derived that if the low volume of periodically malfunctioning legitimate modes along with the malicious ones do not constitute the majority, there exists a threshold of non-contributing users ($NC\text{-}User_{thr}$) where the system continues deducing the actual events' status while retaining its capability in detecting/mitigating the erroneous functionality. □

5 SIMULATION STUDY

This section presents the simulation evaluation of MobileTrust and other related trust schemes (Vehicle Ad hoc network Reputation System (VARS) [20], Event-based Reputation System (ERS) [21], Trust Based Security Enhancement (TBSE) [22], and Social Network approach to Trust Management (SNTM) [24]) over a common setting for VANET communication, under normal operation and two attack scenarios. As the theoretical analysis results provoke, we test the MobileTrust's defense effectiveness for different ratios of contributing/malicious nodes.

5.1 Network Assumptions and Simulation Setup

Regarding the network assumptions, this study concentrates on VANETs and other similar mobile ad hoc networks. Attacks on resource management are mainly studied. MobileTrust is modelled in the SUMO and is implemented in C++. As aforementioned, the assignment of the constant coefficients for the reputation and trust elements is based on relevant studies [20–22, 24] and in a previous analysis by the authors [37].

Figure 4 illustrates the simulation test for the map of Eichstatt, a small-size city with around *1K* citizens. The simulated area covers *4.5 km × 3.0 km*. For the evaluation of the different schemes, a VANET of *1,500* nodes is modelled. The vehicles' top speed varies from *18–80 km/h*. There are utilized the propagation model two-ray ground reflection and the IEEE 802.11p [10] as the MAC layer. Each node has a *2 Mbps* raw bandwidth with *100 m* physical radio range. Table 1 summarizes the simulation parameters.

The experiments include an initialization and an evaluation phase. At first, nodes start communicating with the default values of each scheme. Then, the normal operation and two attack scenarios are evaluated, measuring the overall performance and security, respectively. The simulated period lasts *5* mins for each phase. Every experiment is performed *10* times for each setting, and the average measurements are reported. The randomly located nodes for the first evaluated system are recorded to use the exact same environment in all schemes.

The first case examines the normal operation of each scheme, with no attacks taking place. Figure 4 depicts a snapshot from the simulation and its projection on Google Maps. The different colors illustrate the maximum speed of the vehicles and the current average speed in each road segment. In general, as the processed data are spatial, the performance of MobileTrust can be affected by the density of the mobile nodes. Nevertheless, one of the main contributions of this article is that the trust calculation can be distributed. Thus, if the same computational resources are devoted for every 13K citizens, the similar results will be concluded for medium-sized (50–100K residents) or even bigger (200–500K citizens) cities.

Then, malicious nodes are introduced. For each different attack, *five* experiments are performed for *10, 20, 30, 40*, and *50* percentages of malicious nodes, respectively (networks with higher volume of malicious entities are generally getting discarded in practice).

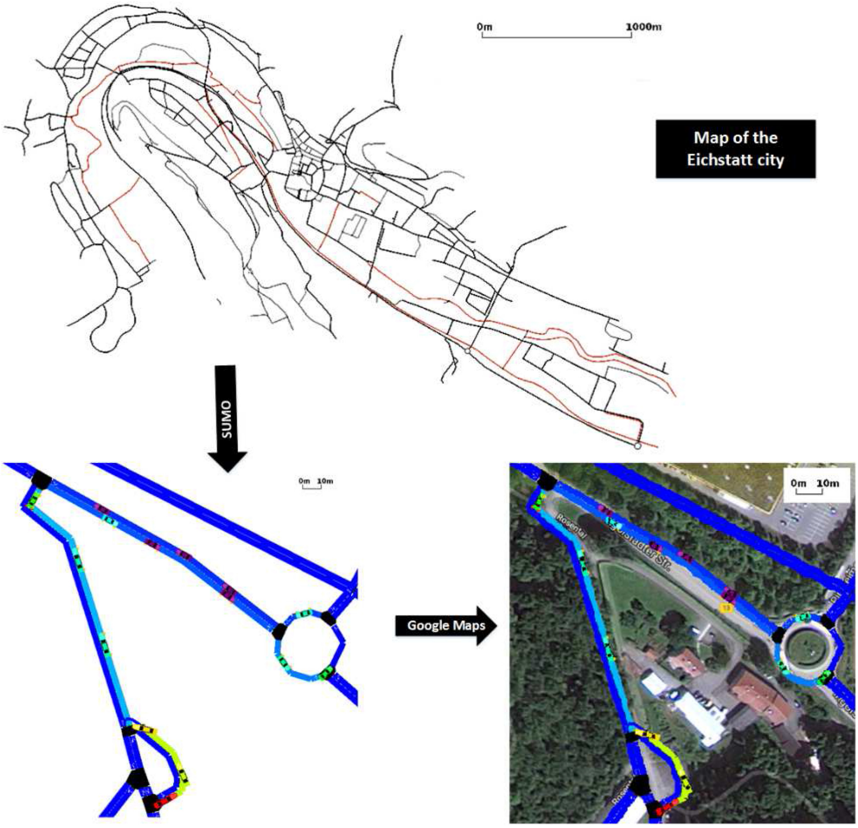

Fig. 4. (i) Simulation map of the German city Eichstatt in scale 1:1000m, (ii) simulation snapshot of a road segment in scale 1:10m, and (iii) its projection on Google Maps.

Table 1. Simulation Parameters of MobileTrust

Parameter	Value
Number of Vehicles	1,500
Vehicle top speed (20% of vehicles for each speed value)	[18, 36, 50, 60, 80] km/h
Frequency	2,400 MHz
Path loss model	Two ray ground
MAC layer	IEEE 802.11p
Antenna height	1.5 m
Transmission power	15 dbm
Transmission range	100 m
Type of antenna	360 degrees
TxGain of antenna	1
RxGain of antenna	1
Fading channel	Ricean
Width of road	20 m

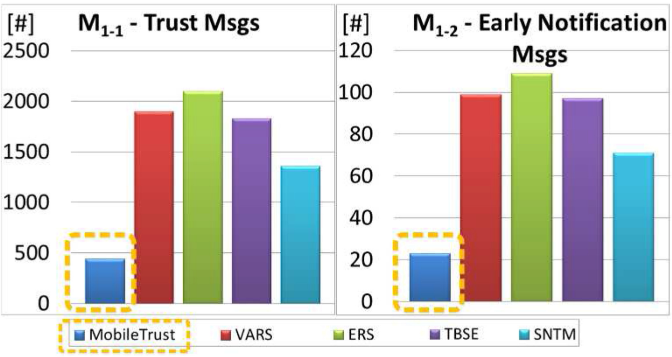

Fig. 5. Performance evaluation results for the metrics $M_{1\text{-}1}$ and $M_{1\text{-}2}$.

5.2 Normal Operation

To evaluate the trust systems under normal operation, we emulate around *100* traffic congestion events and *10* car crashes in each experiment. Two metrics estimate the performance and the energy consumption of each system. $M_{1\text{-}1}$ measures the number of trust messages that are sent during normal operation, representing a rough estimation of the involving communication and computational overheads. $M_{1\text{-}2}$ calculates the number of messages that are required to identify a road event, revealing the early notification capabilities of each scheme. For both metrics, the lower the value, the better. Figure 5 illustrates the relevant metrics.

For $M_{1\text{-}1}$, the MobileTrust authority receives one message by each involved vehicle. The rest P2P systems require a much higher number of messages to form the trust parameters, based on the information propagation scope. The SNTM's closest node strategy has moderate performance, while TBSE and VARS are less efficient. ERS requires the highest amount of interaction.

For $M_{1\text{-}2}$, the MobileTrust publishes instantly a new road event when it is reported by a trusted or a legitimate user with high event reputation along with the current confidence level. Then, it takes a decent amount of incoming contributions to notify the event with high confidence. The rest systems take several messages until a node obtains sufficient knowledge regarding the event's occurrence. Moreover, as direct interaction gains higher weight when integrating direct and indirect knowledge, the nodes in the P2P schemes usually have to detect the event by direct observation before being able to reason about it with high confidence. Among these systems, SNTM provides the earliest notification, as indirect knowledge gives higher weight to the nodes that are closest to the event. TBSE, VARS, and ERS exhibit similar low performance.

As is evident by the two metrics, MobileTrust is the most efficient system with effective early notification of road events. Nonetheless, MobileTrust's performance is normally even better, as public authorities can report several types of scheduled events, helping the drivers to avoid the relevant segments without requiring any further contribution from them. The actual performance of MobileTrust is measured in a preliminary implementation on real devices and is detailed in the next section.

5.3 Attack Scenarios

In the first attack, the malicious nodes launch a social-based on-off attack. They cooperate at the initialization phase to gain high trust values, while at the evaluation phase they start injecting false notifications. We evaluate the effectiveness of the knowledge integration mechanism of each system in detecting bogus data and producing accurate notifications regarding the road state by

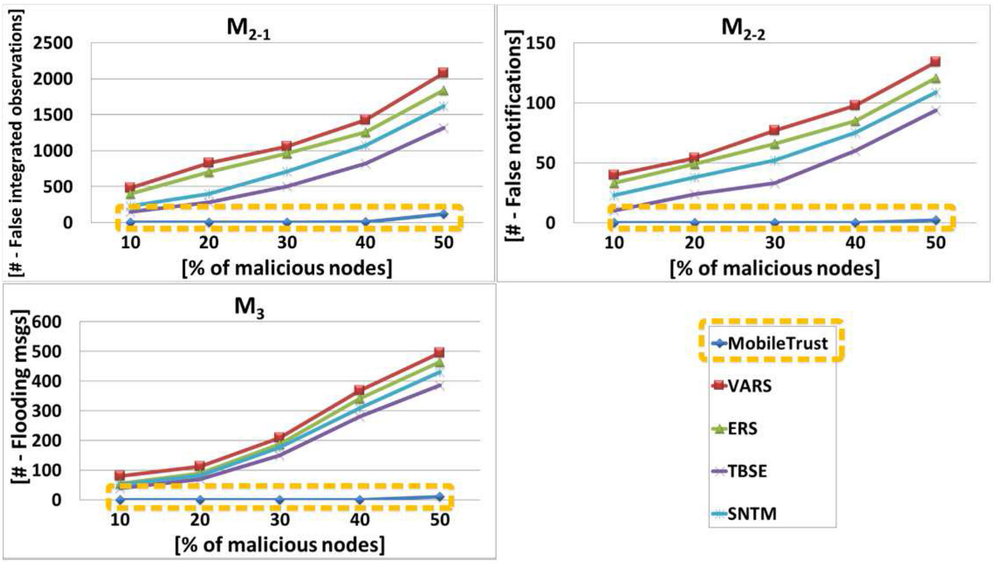

Fig. 6. Security evaluation results for the metrics $M_{2\text{-}1}$, $M_{2\text{-}2}$, and M_3.

measuring the volume of the false observations that are integrated by the trust system, defined as $M_{2\text{-}1}$, and the number of false notifications that are made by the system, defined as $M_{2\text{-}2}$. For both metrics, the lower the value, the better. Figure 6 illustrates the evaluation metrics $M_{2\text{-}1}$ and $M_{2\text{-}2}$.

For $M_{2\text{-}1}$, the single and centralized authority of MobileTrust collects a high volume of legitimate votes and thus is able to recognize the malicious entities and reject their contributions. As the attackers increase, the system may integrate a small number of bogus observations at the beginning of the attack phase. Nevertheless, the authority can detect the misbehaving nodes fast, even for a high volume of attackers (*50%*). All relevant schemes perform poorly for a large number of attackers *(>40%)*. TBSE rejects a high volume of bogus observations for a low and moderate percent of attackers (*10%–30%*) due to its enhanced trust evaluation process. SNTM, ERS, and VARS exhibit the worst results in all cases.

For $M_{2\text{-}2}$, the central authority of MobileTrust reports only a small number of false notifications for a high volume of malicious nodes (*50%*). When many attackers participate in the attack (*>40%*) the relevant schemes perform badly, as they integrate a large number of bogus observations ($M_{2\text{-}1}$). TBSE produces moderate results for low to moderate attackers' volumes (*10%–30%*) as it processes less malicious messages. SNTM, ERS, and VARS have the lowest performance across all attack setups.

In the second scenario, the attackers exploit the ad hoc communication to perform rushing attacks by flooding messages and exhaust the resources of the other nodes. We estimate the robustness of the ad hoc communication against rushing attacks by calculating the number of flooding request messages that are processed by the legitimate nodes, defined as M_3; the lower the value, the better. We consider that all messages contain valid data (not erroneous). Figure 6 also illustrates the metric M_3.

In MobileTrust, the nodes can instantly recognize the involved road segments from the notification format and ignore irrelevant requests. As the nodes exchange only public information, they can reason when a specific notification has just been sent and they will not retransmit it as long as they remain inside the same road segment. Moreover, each node imposes a maximum number

Table 2. Comparison of Trust-based Schemes for VANETs

Feature	MobileTrust	VARS	ERS	TBSE	SNTM
Architecture	Hybrid	P2P	P2P	P2P	P2P
Trust scope	Hybrid	Data	Node	Node	Data
Authentication	YES	NO	NO	NO	NO
Privacy	YES	NO	NO	NO	NO
Early notification	YES	NO	NO	NO	NO
Secure Communication	YES	NO	NO	NO	NO

of responses in the current time-window. Thus, MobileTrust nodes will process a low volume of different requests, constraining the effectiveness and propagation of flooding. The P2P systems serve a high number of flooding requests under many colluding attackers. TBSE can degrade the flooding attack for a low to moderate volume of malicious participants (*10%–20%*), as it detects selfish behavior and ranks negatively the relevant nodes. SNTM and ERS also provide adequate defense against a small amount of misbehaving nodes (*10%*). VARS offers little protection.

5.4 Security and Comparison with Other Systems

As is evident by the above analysis, MobileTrust implements efficient and robust mechanisms for knowledge integration and ad hoc communication that outperform the current state-of-the-art. The proposed system offers immunity-by-design against some types of threats and attacks. It is not vulnerable to information cascading and oversampling, as only first-hand observations are evaluated by a single centralized entity. Moreover, badmouthing and ballot-stuffing attacks are not applicable, as no third-party recommendations about trust are supported. As nodes either advertise their observations directly to the authority or broadcast the public notifications to the other cars, the deletion of transmitted messages is not applicable, in contrast to the P2P trust schemes. Regarding privacy, P2P communication does not disclose any vehicle information, while restricted anonymous usage is supported for the centralized interaction. Eavesdropping is also ineffective, as the transmitted messages are either encrypted communication between the authority and the vehicles or publicly available data.

The cryptographic solutions provide the mainstream security properties. Identity-based attacks (e.g., impersonation, newcomer, HELLO flooding, and Sybil attacks) are countered as the creator of each message is properly authenticated and evaluated. Session symmetric keys encrypt the transmitted data between the authority and the contributors, offering confidentiality and preventing information disclosure. The HMAC primitive enables integrity checks and detects message modification attempts. Random nonces are also included in the message data to avoid replay attacks. The system accomplishes non-repudiation regarding the actions of the authority and the knowledge contributors. Each entity performs a specific set of actions, and role-based authorization is imposed for the central authority, the authenticated contributors, and the passive anonymous users.

Moreover, automatic access control techniques (i.e., the plaintext notification identifiers) and back-off mechanisms degrade the effectiveness of DoS attacks. The low computational/communicational overheads for each task and edge computing further enhances protection.

Table 2 summarizes the comparison results of MobileTrust and the *four* related schemes.

6 IMPLEMENTATION

This section presents the implementation details of MobileTrust on a smart city setting with a real vehicle. We evaluate the system's performance under normal operation and measure the actual runtime overhead, which was sketched in the simulation study (Section 5.2, Normal Operation).

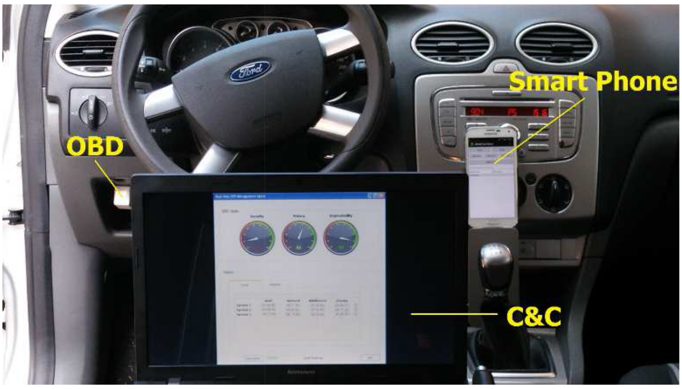

Fig. 7. Smart-car deployment.

6.1 The Car Installation

A real smart-car setting is deployed by the authors in Reference [39]. A smart phone, acting as the vehicles' infotainment, collects information of the vehicles electronic control units (ECU) via a Bluetooth-enabled OBD scan tool and communicates to Internet via a Global System for Mobile Communications (GSM) connection with a command-and-control (C&C) center located in the cloud. The research platform of GRNET Virtual MAchines (ViMA)[5] is utilized for cloud computing. The VMs of the cloud service are monitored through a laptop. The emulated scenario implements the European Union's road safety system eCall [40], where the system detects car accidents and automatically informs the involved emergency services. Figure 7 depicts the smart-car setting.

In this article, we further extend this deployment to implement MobileTrust and model the event types that are described in Section 3. The in-vehicle component is an Android application, running in the smart phone, and the CA is implemented as a web service in the C&C.

6.2 Testbed

Under this setting, a testbed was evaluated on a VM (Intel Core i7 at 2.1 GHz CPU, 8 GB RAM, 64-bit OS Windows 8.1 Pro), where *1K* event notifications were advertised. Figure 8 illustrates the response time of MobileTrust in ms for the smart phone client application and the central authority at the backend, with spikes depicting TLS processing. The additional communication delay through GSM was on average *1–4* seconds. The processing resource consumption of the internal components is summarized in Table 3. The V2V interactions also include the referred cryptographic services and exhibit similar results.

6.3 Parallelization in GPUs

As aforementioned, the core knowledge integration mechanism of MobileTrust can be parallelized in a fine-grained manner. Except from the CPU parallelization on multiple cores, the GPUs is a well-tried practice that can drastically reduce the processing time in computation-intensive operations. Recommendation systems perform well in this setting with *30–32* times speedup [38]. In this study, we utilize the CUDA General Purpose GPU (GPGPU) to parallelize the MobileTrust algorithm. The GPU-accelerated deployment is implemented on the same machine as the CPU version, equipped

[5] GRNET ViMA: https://vima.grnet.gr/about/info/en/.

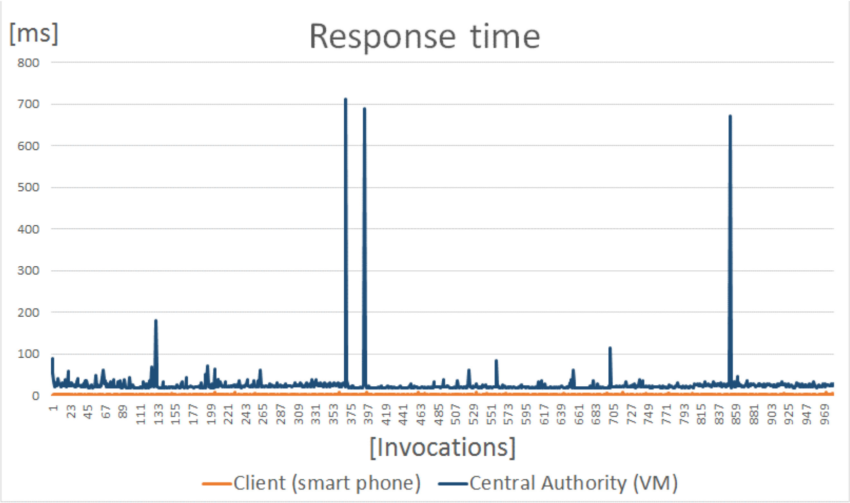

Fig. 8. Response time of MobileTrust applications in ms.

Table 3. Resource Consumption of MobileTrust Components

Component	RAM [KB]	CPU [ms]
Cryptographic service		
TLS handshake	1,400	692.8
TLS session	65	30.45
RSA-2048 sign	6.1	3.229
RSA-2048 verify	3.2	0.117
AES-256 in CBC	3.4	0.004
SHA512	2.4	0.009
Trust scheme		
Reputation estimation (per node)	0.08–1.1	3.4–54.48
Trust computation (per node)	1.4	0.4
Event evaluation		
Event confidence calculation (per 500 nodes in a road segment)	1.5	0.14

with the NVIDIA GeForce GTX 1050 GPU (640 cores, 2 GB buffer, 6 Gbps memory speed, 1.4 GHz clock).

Consider the most frequent and resource-demanding traffic congestion events for road segments with *500* vehicles on average. After the event's active period, the event's confidence must be evaluated (see Table 3). Then, the reputation histories and the trust structures of the *500* users must be updated. Figure 9 depicts the average processing time of evaluating a road event for the CPU, CPU-accelerated, and GPU-accelerated versions of MobileTrust. With the parallel processing on *4* CPU cores (*8* threads), each thread processes one event for a specific road segment and updates the users' data. The CPU parallelism busts performance by *23%* in comparison with the single CPU version. This CPU-accelerated implementation is further improved by parallelizing the users' data update in the GPU (*500* cores of *560 KB* each). This setting is around *32* times faster than the single CPU one and can serve *1.16* events per second. The optimized proposal can be deployed by every VM in both cloud and MEC architectures, providing an overall efficient and globally scalable system with real-time response.

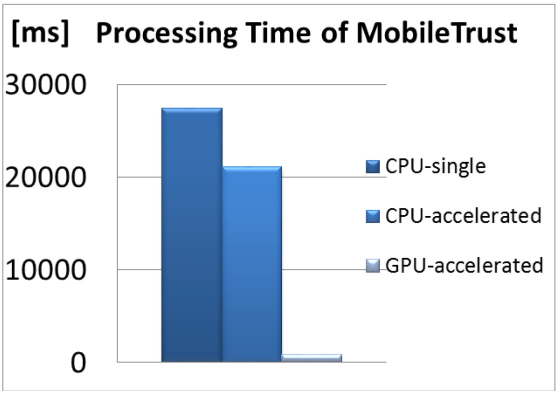

Fig. 9. Processing time of MobileTrust versions in ms.

7 CONCLUSIONS

In this work MobileTrust—a trust-based resource-sharing system for VANETs is presented. The system is modelled in the SUMO simulator along with *four* relevant schemes, and a comparative analysis is conducted. MobileTrust is an energy-efficient solution that provides the higher level of protection. On real embedded devices in the context of an emulated VANET and smart-city CPS, the overhead of the trust scheme is low and acceptable for the added security and the overall protection level that is achieved.

One limitation for MobileTrust and all the relevant systems concerns the accuracy of the location positioning system [41]. For example, the discrimination of the different lanes in a road section may not be possible. Thus, if only one lane faces a specific event (i.e., traffic jam), the notification and rating mechanisms can produce erroneous results. Nevertheless, these issues will be contained, as the Global Positioning System (GPS) technology is evolved and integrated with 5G pioneering frequency communications (such as 26 and 60 GHz).

As future extensions of our work, we consider the further inclusion of modern smart transportation technologies and circular economy business models. The goal is to extend the trust mechanism to evaluate application-specific features for ecosystems that provide operations such as ride-service hailing and car-sharing.

REFERENCES

[1] N. Trent. 2017. Cyber physical system 2017 market segmentation, application, technology & market analysis research report to 2021. Wiseguyreports.com. Retrieved from http://www.abnewswire.com/pressreleases/cyber-physical-system-2017-market-segmentationapplicationtechnology-market-analysis-research-report-to-2021_150981.html.

[2] A. Dahiya and R. K. Chauhan. 2010. A comparative study of MANET and VANET environment. *J. Comput.* 2, 7 (2010), 87–92.

[3] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni. 2011. A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). *Wirel. Commun. Mobile Comput.* 11, 7 (2011), 813–828.

[4] P. Chouhan, G. Kaushal, and U. Prajapath. 2016. Comparative study MANET and VANET. *Int. J. Eng. Comput. Sci.* 5, 4 (2016), 16079–16083.

[5] N. Saxena, S. Grijalva, V. Chukwuka, and A. V. Vasilakos. 2017. Network security and privacy challenges in smart vehicle-to-grid. *IEEE Wirel. Commun.* 24, 4 (2017), 88–98.

[6] C. K. Toh. 2008. Future application scenarios for MANET-based intelligent transportation systems. In *Proceedings of the International Conference on Future Generation Communication and Networking (FGCN'08).* IEEE, 1–4.

[7] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni. 2010. Emergency services in future intelligent transportation systems based on vehicular communication networks. *IEEE Intell. Transport. Syst. Mag.* 2, 2 (2010), 6–20.

[8] B. Cici, A. Markopoulou, and N. Laoutaris. 2016. SORS: A scalable online ridesharing system. In *Proceedings of the 9th International Workshop on Computational Transportation Science (IWCTS'16)*.

[9] S. Grafling, P. Mahonen, and J. Riihijarvi. 2010. Performance evaluation of IEEE 1609 WAVE and IEEE 802.11p for vehicular communications. In *Proceedings of the IEEE 2ndInternational Conference on Ubiquitous and Future Networks (ICUFN'10)*. 344–348.

[10] Y. Wang, A. Ahmed, B. Krishnamachari, and K. Psounis. 2008. IEEE 802.11p performance evaluation and protocol enhancement. In *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES'08)*. 317–322.

[11] D. Eckhoff, N. Sofra, and R. German. 2013. A performance study if cooperative awareness in ETSI ITS G5 and IEEE WAVE. In *Proceedings of the IEEE 10th Conference on Wireless On-demand Network Systems and Services (WONS'13)*. 196–200.

[12] X. Gong and N. B. Shroff. 2017. Trustful mobile crowdsensing for strategic users with private qualities. In *Proceedings of the 15thInternational Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'17)*.

[13] M. Mejia and R. Haparro-Vargas. 2013. Distributed trust and reputation mechanisms for vehicular ad hoc networks. In *Vehicular Technologies—Deployment and Applications*. Chapter 6, 117–140. INTECH, UK.

[14] N. J. Patel and R. H. Jhaveri. 2015. Trust-based approaches for secure routing in VANET: A survey. In *Proceedings of the International Conference on Advanced Computing Technologies and Applications (ICACTA'15)*.

[15] M. A. Ngadi, R. H. Khokhar, and S. Mandala. 2008. A review of current routing attacks in mobile ad hoc networks. *Int. J. Comput. Sci. Sec.* 2, 3 (2008), 18–29.

[16] D. Airehrour, J. Gutierrez, and S. K. Ray. 2016. Secure routing for internet of things. *J. Netw. Comput. Applic.* 66, 198–213. Elsevier.

[17] E. Carrara and G. Hogden. 2007. *ENISA: Reputation-based System: A Security Analysis*. European Network and Information Security Agency (ENISA) Position Paper 2. ENISA, Athens, Greece.

[18] J. Zhang. 2012. Trust management for VANETs: Challenges, desired properties and future directions. *Int. J. Distrib. Syst. Technol.* 1, 1 (2012), 48–62.

[19] S. A. Soleymani, A. H. Abdullah, W. H. Hassan, M. H. Anisi, S. Goudarzi, M. A. R. Baee, and S. Mandala. 2015. Trust management in vehicular ad hoc network: A systematic review. *EURASIP J. Wirel. Commun. Netw.* 146, 1–22. Springer.

[20] F. Dotzer, L. Fischer, and P. Magiera. 2005. VARS: A vehicle ad hoc network reputation system. In *Proceedings of the 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)*. 454–456.

[21] N.-W. Lo and H.-C. Tsai. 2009. A reputation system for traffic safety event on vehicular ad hoc networks. *EURASIP J. Wirel. Commun. Netw.—Spec. Issue Enabl. Wirel. Technol. Green Pervas. Comput.* 9 ID 125348, 1–10. Springer.

[22] Z. Wei, F. R. Yu, and A. Boukerche. 2014. Trust based security enhancements for vehicular ad hoc networks. In *Proceedings of the 4thACM International Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications (DIVANet'14)*. 103–109.

[23] G. Shafer and J. Pearl. 1990. *Readings in Uncertain Reasoning*. 1–768. Morgan Kaufman Publishers.

[24] Z. Huang, S. Ruj, M. A. Cavenaghu, M. Stojmenovic, and A. Nayak. 2014. A social network approach to trust management. *Peer-to-Peer Netw. Applic.* 7, 3 (2014), 229–242. Springer.

[25] Q. Kong, R. Lu, M. Ma, and H. Bao. 2019. A privacy-preserving sensory data sharing scheme in internet of vehicles. *Fut. Gen. Comput. Syst.* 92, 644–655. Elsevier.

[26] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis. 2016. Migrating running applications across mobile edge clouds. In *Proceedings of the ACM 22nd International Conference on Mobile Computing and Networking (MobiCom'16)*. 435–436.

[27] L. Gkatzikis and I. Koutsopoulos. 2014. Mobiles on cloud nine: Efficient task migration policies for cloud computing systems. In *Proceedings of the IEEE 3rd International Conference on Cloud Networking (CloudNet'14)*. 204–210.

[28] S. Ioannidis and P. Marbach. 2008. On the design of hybrid peer-to-peer systems. In *Proceedings of the ACM International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS'08)*. 157–168.

[29] Y. Zhang, L. Xu, and X. Wang. 2008. A cooperative secure routing protocol based on reputation system for ad hoc networks. *J. Commun.* 3, 6 (2008), 43–50.

[30] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. 2010. A view of cloud computing. *Commun. ACM* 53, 4 (2010), 50–58.

[31] W. Tian, Y. Zhao, Y. Zhong, M. Xu, and C. Jing. 2011. A dynamic and integrated load-balancing scheduling algorithm for cloud datacenters. In *Proceedings of the IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS'11)*. 311–315.

[32] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. 2000. OceanStore: An architecture for global-scale persistent storage. *ACM SIGPLAN Not.* 35, 11 (2000), 190–201.

[33] B. Blanchet. 2014. Automatic verification of security protocols in the symbolic model: The verifier ProVerif. In *Foundations of Security Analysis and Design (FOSAD) VII. LNCS*, 8604, 54–87. Springer.

[34] I. Ristic. 2016. OpenSSL cookbook. Feisty Duck, 1–94, March 2016. Retrieved from https://www.feistyduck.com/books/openssl-cookbook/.

[35] S. Yang, S. Ji, and R. Beyah. 2018. DPPG: A dynamic password policy generation system. *IEEE Trans. Inf. Forens. Sec.* 13, 3 (2018), 545–558.

[36] Y. S. Han, H.-T. Pai, R. Zheng, and W. H. Mow. 2014. Efficient exact regenerating codes for Byzantine fault tolerance in distributed networked storage. *IEEE Trans. Commun.* 62, 2 (2014), 385–397.

[37] G. Hatzivasilis, I. Papaefstathiou, and C. Manifavas. 2014. ModConTR: A modular and configurable trust and reputation-based system for secure routing in ad hoc networks. In *Proceedings of the 11th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'14).*

[38] A. V. Rodrigues, A. Jorge, and I. Dutra. 2015. Accelerating recommender systems using GPUs. In *Proceedings of the 30th ACM Symposium on Applied Computing (SAC'15).* 879–884.

[39] K. Fysarakis, G. Hatzivasilis, I. Papaefstathiou, and C. Manifavas. 2016. RtVMF: A secure real-time vehicle management framework with critical incident response. *IEEE Pervas. Comput. Mag.—Spec. Issue Smart Vehic. Spaces* 15, 1 (2016), 22–30.

[40] J. Hayes. 2014. Calling all cars. *IET Eng. Technol.* 9, 3 (2014), 58–61.

[41] Q. Pei, B. Kang, L. Zhang, K.-K. R. Choo, Y. Zhang, and Y. Sun. 2018. Secure and privacy-preserving 3D vehicle positioning schemes for vehicular ad hoc network. *EURASIP J. Wirel. Commun. Netw.* 1, 271 (2018), 1–12. Springer.

Received December 2018; revised July 2019; accepted September 2019