Is it really easy to detect sybil attacks in C-ITS environments: a position paper

Badis Hammi*, Mohamed Yacine Idir†, Sherali Zeadally‡, Rida Khatoun§, Jamel Nebhen¶
*EPITA Engineering School, France
badis.hammi@epita.fr
†Université Gustave Eiffel, France
myacine.idir@uge.fr
‡University of Kentucky, USA
szeadally@uky.edu
§Institut Mines Telecom Paris, France
rida.khatoun@telecom-paris.fr
¶Prince Sattam bin Abdulaziz University, KSA
j.nebhen@psau.edu.sa

***Abstract*—In the context of current smart cities, Cooperative Intelligent Transportation Systems (C-ITS) represent one of the main use case scenarios that aim to improve peoples' daily lives. Thus, during the last few years, numerous standards have been adopted to regulate such networks. Within a C-ITS, a large number of messages are exchanged continuously in order to ensure that the different applications operate efficiently. However, these networks can be the target of numerous attacks. The sybil attack is among the most dangerous ones. In a sybil attack, an attacker creates multiple identities and then disguises as several fake stations in order to interfere with the normal operations of the system or profit from provided services. We analyze recently proposed sybil detection approaches regarding their compliance with the current C-ITS standards as well as their evaluation methods. We provide several recommendations such as network and attack models as well as an urban and highway datasets that can be considered in future research in sybil attack detection.**

***Index Terms*—Certificate, C-ITS, PKI, Privacy, Pseudonym, Security, Sybil attack, VANET**

I. Introduction and problem statement

Cooperative Intelligent Transportation Systems (C-ITS) comprise emerging information and communication technologies that improve the transport of people and goods. Indeed, in recent years, several standards have been developed for vehicle-to-vehicle and vehicle-to-infrastructure communications and enable various ITS-related applications. More precisely, communications in C-ITS are mainly regulated by two international standardization organizations namely, the European Telecommunications Standards Institute (ETSI) standards in Europe [1][2] and the Institute of Electrical and Electronics Engineers (IEEE) in the United States as well as other countries such as China, India and others, commonly known as Dedicated Short-Range Radio (DSRC) [3][4]. Standardization ensures interoperability, supports regulations and legislation, and new developments.

Over the last few years, a lot of attention has been given to smart cities which has promoted interests in Cooperative Intelligent Transportation Systems, where they are involved in multiple scenarios [5]. This rise in interests is reflected by the numerous deployment projects that have been initiated and supported by governments, e.g, Security Credential Management System (SCMS) in the USA [6] and SCOOP@F in France [7]. Many car manufacturers such as Nissan, Volvo, Renault and Toyota, have provided numerous prototypes of vehicles equipped with sensors [8], dedicated computing hardware and Dedicated Short-Range Radio for communication with other nearby vehicles (called Intelligent Transportation System's Station-Vehicle (ITSS-V) in the C-ITS context)[1] or with road side infrastructure (Intelligent Transportation System's Station-Road Side Unit (ITSS-R))[2]).

C-ITS can facilitate the driver's decision making tasks (e.g., trip planning based on traffic congestion on the road) as well as the improvement of its safety through a plethora of applications such as intersection collision avoidance, cooperative collision warning, blind spot warning, emergency electronic brake lights, lane change assistance and traffic flow control [9][10]. To support different applications, a large number of messages are exchanged continuously. In ETSI based architectures, ITSSs use Cooperative Awareness Messages (CAM) [11] and Decentralized Environmental Notification Messages (DENM) [12]. In IEEE based architectures, ITSSs use Basic Safety Messages (BSM) [13]. One example of the importance of these messages is where BSM has the potential to prevent up to 75% of all roadway crashes according to [14][15]. Thus, the correctness and reliability of the exchanged messages have a direct impact on the efficiency and effectiveness of the proposed services and the applications used.

C-ITS applications and components can be the target of numerous security issues and attacks but the sybil attack is considered to be among the most dangerous ones [9][10][16][17]. In a sybil attack (as Figure 1 shows), the attacker node creates different virtual nodes (also called sybil ghosts) in order to

[1] In the remaining of this paper, we use the terms vehicle, node, and ITSS-V to refer to a connected vehicle.

[2] In the rest of this paper, we use the terms RSU and ITSS-R interchangeably to refer to a connected road side unit.

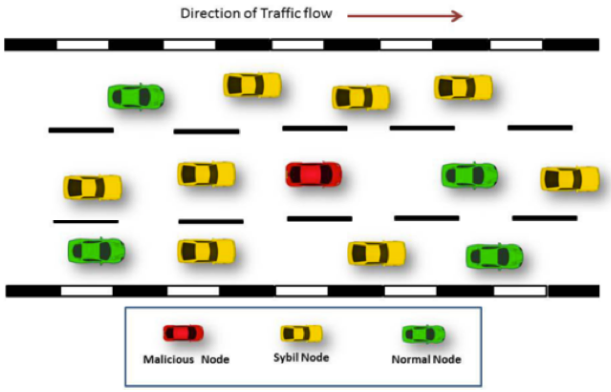


Fig. 1: Sybil attack: traffic congestion

have a certain influence on the network's decisions especially in voting based protocols and applications. The creation of the sybil ghosts is performed by creating different messages using different fake identities and different fake locations.

The strength of C-ITS relies on the strong cooperation of its stations. This requires one station to receive enough credible information from legitimate stations. In other words, most C-ITS based applications, such as hazard notification, collision warning, route navigation, traffic status, and so on, need the cooperation of stations. The similar view sensed by multiple distinct stations for a certain traffic situation can provide trustable correctness and a reliable proof about the traffic situation. However, in a sybil attack, a malicious node generates multiple fake identities to create many untrusted virtual nodes, which violates the fundamental assumption of receiving the real traffic situation, in implementing those applications [18][10]. Hence, the sybil attack can be applied to different scenarios such as: (1) to perpetrate different types of Distributed Denial of Service attacks by disrupting the normal operations of data dissemination protocols [19]. One possible attack variant occurs when the malicious node makes seemingly disjoint paths in multipath routing protocols wherein all converge to it via multiple sybil nodes. Then, the malicious node could drop all (or part of) the messages that go through it perpetrating the black hole attack (or grey hole attack) [10]; (2) Sybil attacks can cause more serious safety threats as reported by [19] in the deployment of deceleration warning systems [20]. For example, if a vehicle reduces its speed significantly or stops abruptly, it will broadcast a DENM message in an ETSI based project (or a BSM message in an IEEE based project) to warn the following stations. Recipients will relay the message to stations further behind. However, this forwarding process can be interrupted by a large number of malicious Sybil stations. In this way, the malicious adversary can create a massive pileup on the highway, potentially causing serious damages. (3) To influence voting and reputation systems, where a sybil attack can create enough malicious identities to report repeatedly which will falsify the voting results. (4) To influence the aggregation of data because a sybil identity may be able to report malicious readers and report incorrect sensor readings thereby influencing the overall computed result [17]. (5) To impact the distributed storage where the sybil attack causes huge data damages in a network where the storage is distributed because the data will be given to virtual nodes [21]. (6) To impact resource allocation such as the Time Division Multiple Access (TDMA) schedule which is based on the knowledge of the network topology; this allocation will be poorly distributed when there are fictitious nodes [17]. (7) To fake a traffic congestion for the road management platform and the vehicles nearby. Then, the latter may choose other routes [22].

Problem statement

Numerous solutions [16][9][23][24][25] have been proposed to detect and mitigate sybil attacks. Nevertheless, the vast majority of existing works are either outdated or are not adapted to current C-ITS infrastructures. In this section we discuss the different requirements that a security solution must ensure. We use these research requirements to analyze the different security proposals later in Section II and in the research methodology proposed in this work.

In the last few years, multiple regulations and standards have been adopted [26]. What is commonly named as beacon messages and alert messages have been replaced and standardized (in different standards) to Cooperative Awareness Message (CAM) [11], Decentralized Environmental Notification Message (DENM) [12], Basic Safety Messages (BSM), Signal Phase and Timing (SPAT), MapData Messages (MAP) [13], In Vehicle Information (IVI), Traffic Light Control (TLC) [27] [28], Point of Interest (POI), and others. Their structures have already been defined and only a standard amendment or a new version of the standard can modify them. **Thus, Req 1[3]: the proposal of a sybil detection solution that requires the complete modification of these structures or their replacement by other structures (e.g., [29][30]) cannot be considered.**

Moreover, the security mechanisms in C-ITS environments are already standardized including the security architecture and the secure message formats. Indeed, to handle security requirements, the Public Key Infrastructure (PKI) solution has been adopted by all the standards. The IEEE 1609.2 standard [31] specifies a set of security services to support ITS communications. It defines secure messages formats and processing for Wireless Access Vehicular Environments (WAVE) devices, including methods to secure WAVE management messages and methods to secure application messages. It also describes administrative functions necessary to support the core security functions. For the PKI infrastructure, the standard classifies all the entities that provide or use IEEE 1609.2 security services into two categories namely, Certificate Authority entities and End entities.

In 2014, the National Highway Traffic Safety Administration (NHTSA), Department of Transportation (DOT) published a Request for Information (RFI) called Vehicle-to-Vehicle

[3] Req stands for requirement

Security Credential Management System (V2V SCMS) [32]. The purpose of this RFI was to seek responses concerning the establishment of an SCMS, security approaches for a V2V environment, technical and organizational aspects of the SCMS. In short, the PKI system was selected as the security solution. Further, in 2016, DOT and NHTSA, along with Crash Avoidance Metrics Partners (CAMP LLC) Vehicle Safety Communications 5 (VSC5) Consortium[4] published parts of the SCMS Proof-of-Concept specification [6]. The latter extends the last RFI to V2I communications and consider RoadSide Unit (RSU) usage. The report [6] focuses on PKI description, the certificates used and their management.

Also, the ETSI ITS Technical Committee Working Group 5 developed the ITS security architecture, providing security standards as well as guidance on the use of security standards. ETSI TS 102 940 [33], ETSI TS 102 941 [34] and ETSI TS 102 731 [35] standards specify and describe the security services and security architecture for ITS communications, and the ETSI TS 103 097 standard [36] [37] specifies the V2X message security header and the various certificates' formats.

Consequently, Req 2: the proposal of a solution that requires a new security architecture or security mechanisms different from the standards cannot be considered.

Finally, since ITSs periodically transmit messages that contain information about their position and localization, an attacker can, using such information, track the station or create detailed mobility patterns of individual drivers [38]. This problem is addressed by providing a station with a set of pseudonyms. The station uses each pseudonym for a limited duration. More precisely, by relying on the PKI, each ITSS uses two certificates simultaneously: (1) an Enrollment Certificate (EC) (also called Long Term Certificate (LTC)) and (2) a Pseudonym Certificate (PC) (also called Short Term Certificate (STC)). Known only by the EC Authority (ECA) and its owner (ITSS), the EC is not used in common communications, but is used only to authenticate the ITSS to the PKI in order to request new PCs. However, the PC is used for the ITSS communications. In order to protect the privacy of road users, a regular change of pseudonyms is required, for example in the SCMS project [14], an ITSS uses more than 1000 PCs per year and this number can even reach 100000 according to [39]. In the SCOOP@F project an ITSS uses 520 PCs per year [40]. When a station changes its PC, it changes all its credentials and all the network related information such as IP addresses, MAC addresses, Station IDs, and so on. **Therefore, Req 3: an approach that relies on a station's history or assumes that a station does not change its identity multiple times during a journey, cannot be considered.** However, some PKI architectures (such as SCMS) implements a Linkage Authority to link the different PCs of the same vehicle for some purpose such as certificates' revocation. This authority has a limited cooperation with the other PKI authorities (e.g., it can cooperate with the LTCA but not directly with the PCA. **Hence, Req 4: if the history of the vehicle must be considered in the detection process, the linkage of the different PCs must be provided by the Linkage authority while adhering to the PKI disclosure policies.**

Contributions of this work

The main research contributions of this work include:

1) We present a state-of-the-art review of solution aimed at detecting sybil attacks. We also provide a discussion and analysis to show that the majority of these works are not suitable for current C-ITS and prove that sybil attack still represent an open issue.
2) We provide network and attack models as well as recommendations for a research methodology that can be considered in future works.
3) We provide one dataset for an urban scenario and another dataset for a highway scenario. These datasets describe the activity of 86,747 vehicles and can be used by researchers in future works. We also present a quick statistical characterization of some sybil attack scenarios using the urban dataset.

The rest of the paper is organized as follows: Section II presents a review of sybil detection methods. Then, Section III provides network and attack models that can be adopted and considered in further sybil detection works as well as other research recommendations. Section IV describes our statistical characterization and analysis of sybil attacks. Finally, Section V concludes the paper and describes some future research directions.

II. State of the art of sybil detection approaches

The main goal of sybil attack detection schemes is to detect (1) the sybil nodes (virtual nodes) and (2) the attacker that creates these sybil nodes. Due to the wireless environment's features, the detection of the attacker remains a challenging task compared to the detection of sybil nodes. Thus, the majority of detection approaches aim at the detection of the sybil nodes and are not able to detect their creator.

We classify sybil detection schemes into three classes: (1) position verification, (2) reputation and data-driven systems and (3) resource testing. The majority of works and surveys such as [41][42][43] consider public cryptography based approaches as another class for sybil detection. However, we believe that these approaches are for prevention and not for detection because there is no detection engine that decides whether the activity is part of an attack or not.

Position verification

In the position verification approach, the claimed position of each station is verified. This verification is realized through different methods such as (1) the signal strength, where the Received Signal Strength Indication (RSSI) is used to estimate the position of a station. Next, this estimated position is compared with the position indicated in the station's message.

[4] The members of the consortium are Ford Motor Company, General Motors LLC, Honda R&D Americas Inc, Hyundai-Kia America Technical Center Inc, Mazda, Nissan Technical Center North America Inc., and Volkswagen Group of America

(2) Dedicated radars or sensors calculate the position of neighboring stations and compare them to their claimed positions in the messages they send. Position verification approaches rely on witness and verification by the neighbor nodes. Thus, it requires a continuous collaboration of the latter. Moreover, it requires additional equipment such as radars or special sensors, which makes these approaches really costly in terms of hardware, computation, and bandwidth.

In [44], *Xiao et al.* measure the signal strength of beacons received and compare them with the claimed position of a vehicle. These measures are performed by vehicles traveling in the opposite direction to avoid fake measures sent by the attacker. This work was later extended by the same authors in [19], where a Random Sample Consensus algorithm has been used in order to increase the estimation accuracy against outlier data created by sybil nodes. Besides, the authors applied a statistical method that performs hypothesis tests on accumulated measurements. Next this approach evaluates if the measurements match a normal distribution pattern. A sybil node is reported if its distribution pattern is inconsistent with its claimed physical position. In [45], Golle *et al.* propose a sensor-driven technique that allows nodes to detect incorrect information and identify the source of the incorrect information. Their approach relies on the network model wherein each vehicle contains all the knowledge about the network. The scheme focused on the reasoning of conflicting observations, but simply assumed the ability of the nodes in detecting the distance to other nodes or the precise locations of other nodes [19]. In [46], *Rabieh et al.* combines resource testing and position verification techniques. Their approach is applied at the RSU level where each RSU looks for anomalies such as wrong distance, overlaps, wrong vehicle count or contradictory radio signals, and then sends a challenge packet to the suspicious node using a directional antenna. If the node is at the expected location, it should be able to receive the challenge and send back a valid response. In [47], Yan *et al.* consider a vehicle model that relies on front and rear radars that detect neighboring stations within a line of sight in a radius of 200 meters. Moreover, according to the position-based cell-based approach [48], the road is divided into equal-sized location-based cells. Each vehicle can directly communicate with every other vehicle in the cell. If a vehicle receives a message, the receiver matches its Global Positioning System (GPS) position with the position calculated by the radars. If both positions match, the message is accepted and the sender is labeled as honest. A history is recorded for each station. A station without a history is not trustable. Therefore, this approach completely ignores the station's privacy and non-tracking concerns.

Benkirane *et al.* [17] proposed an approach where they assume that each vehicle on the road is linked to three reliable RSUs at a given time. Thus, when a vehicle broadcasts a message to other vehicles, the three RSUs also receive this message. The detection mechanism involves the collaboration of the RSUs. Indeed, based on the Received Signal Strength Indication (RSSI) measurements made by the three RSUs, the distances that separate the vehicle to each of the three RSUs at a given time is calculated. Since the messages of different sybil nodes are broadcasted by one physical node, each RSU receives the same RSSI values which allows the detection of the sybil nodes. However, due to the optimal positioning of the RSUs, it can be difficult if not impossible that each vehicle is always linked to three RSUs. In [49], *Kabbur et al.* proposed a similar approach. The proposed solution places the RSU in such a way that the position of any vehicle can be found using triangulation by using the RSSI to calculate the distance between the vehicle and each one of the three RSUs. Then, each RSU attaches a timestamp to the triangulated location of node and broadcasts a message containing the timestamp and the triangulated location to neighboring RSUs to build a path for each vehicle and track/detect the node generator of the sybil nodes. However, as in [17], due to the optimal position of the RSUs, it can be difficult to always keep each vehicle linked to three RSUs. Moreover, this approach does not consider the non-tracking requirement.

Data-driven systems

This technique relies on data collected from stations and generally does not require special hardware. In [50], Chang *et al.* proposed Footprint, an approach that uses vehicles' trajectories to identify them. More precisely, when a station approaches an RSU, it requests an authorized message from the RSU as a proof of the time it has appeared at this RSU. During a communication with another station, the participating station must provide the authorized messages collected, that represent its trajectory. A sybil attack is detected if there are similarities in the stations' trajectories. In order to be trusted, a station must go through numerous RSUs to collect numerous authorized messages. This approach has several shortcomings. Indeed, generally, the placement of RSUs is made in an optimal way, in order to maximize the coverage while minimizing the number of RSUs (for economical purposes). An RSU covers a diameter of more than 1 Kilometer (3.5 Km in some cases) [51] which represent a zone that can include numerous stations and roads in an urban environment. Similarly, numerous vehicles will obtain authorized messages from the same RSUs which makes them share the same trajectories. The same problem occurs in the highway scenario where numerous vehicles share the same authorized messages and trajectories which makes the detection process more complex. Moreover, for privacy purposes, the stations must change their credentials and all the network's information (e.g., IP addresses and MAC addresses). However, if a vehicle keeps its trajectory history, its privacy is compromised. Finally, the process of actively requesting and providing authorized messages may overload an already loaded network.

Similarly, Hussain *et al.* [52] proposed an RSU driven detection scheme. In this approach, the authors proposed a new PKI architecture, where each region is autonomous. In each region, stations dynamically receive tokens from the nearby RSUs and use them to report events in the area only once (one token per message). Then, a pre-assembly analysis on messages is provided to detect sibyl nodes that send messages without having tokens. Nonetheless, as in [50], this approach relies on providing additional messages for tokens which overload the network. Moreover, the detected sybil nodes are revoked using CRLs. Each region manages its own CRL, and this makes handling of the CRLs more complex for stations. Finally, this approach considers the proposal of a new PKI architecture which does not meet the requirements Req 1 and Req 2 that we discussed above. Following the exact same reasoning, Park *et al.* [18] proposed a detection approach wherein stations obtain a certified timestamp signed by each RSU the vehicle passes by. The communication messages sent by the stations must contain these certified timestamps. However, this approach suffers exactly from the same shortcomings as the previous RSU assisted approaches. Furthermore, these schemes rely on the idea that a vehicle obtains a certification from the RSU because it can correctly authenticate it although the only way for an RSU to authenticate a station is through its certificate. However, a station with numerous certificates can be authenticated as numerous stations because no linkage between certificates must be possible [38], which cannot really help the detection process. Also, Chen *et al.* [42] used the feature of gathering signed timestamps from RSUs and use them in communications with other stations. Each station performs the detection process independently. Relying on the signed timestamps, the detection scheme, within each station, constructs trajectories of the sending stations. The trajectories are analyzed according to a normal trajectory pattern. A station having a trajectory different from the pattern is considered as a sybil ghost. However, in addition to the shortcomings discussed above, in this approach, each station performs an independent detection, which means that the detector will make decisions without having a global view of the execution environment.

Grover *et al.* [53] proposed a detection scheme, where each station, based on the exchanged beacon messages, keeps periodically a record of all its neighbors. Then, each station exchanges groups of its neighbors periodically with other stations and performs the intersection of these groups to determine the stations that appear in multiple groups simultaneously. If some nodes observe that they share the same neighbors for a significant period of time, these similar neighbors are identified as sybil nodes. Nonetheless, this approach is not very efficient in an urban environment due to the coverage range of stations ($\approx 1Km$) because the vehicles will probably share the same neighbors, especially considering that the stations' speeds are low.

Sowattana *et al.* [29] proposed a distributed consensus based scheme for sybil detection. In their approach they proposed a novel format for beacon messages, which includes the list of stations' neighbors. Each node performs the detection mechanism after receiving the required number of messages. After analyzing the neighbors lists, a node is considered as a sybil node if its position is inside the intersected area of two communication nodes. However, this assumption is not always verified. Indeed, due to stations' coverage zones (some efforts have recently been proposed to improve this coverage area [54] [55]), numerous legitimate stations can be in the identified intersection. Moreover, this approach proposes a novel beacon message which does not meet the requirement Req 1 discussed above. Finally, relying on the resource-constrained stations for the execution of the detection process can represent a weakness.

In [43] Gu *et al.* proposed a detection scheme that exploits a station's messages to build a Driving Pattern Matrix (DPM). Then, a minimum distance classifier is used to detect the unusual patterns. However, no further details about the detection protocol and its implementation were given.

Bißmeyer *et al.* [56] proposed a central approach in which vehicles send Misbehavior Reports (MRs) to a central entity when detecting overlaps. These MRs contain signed evidence of the overlap and trust statements toward neighbors. The central entity analyzes all received MRs and then decides whether a node is a sybil ghost or not. However, this approach requires the linkage of pseudonym certificates of stations, which is contrary to standard requirements for non-tracking requirement (Req 4).

In [57] Ayaida *et al.* proposed a detection approach whose key idea is that each vehicle monitors its neighborhood in order to detect an eventual sybil attack. This is achieved by comparing the real accurate speed of the vehicle and the one estimated using the Vehicle-to-Vehicle (V2V) communications with vehicles in the vicinity. This estimated speed is obtained using the traffic flow fundamental diagram of the road's portion where the vehicles are moving.

Boeira et al. [30] proposed an approach for sybil mitigation that uses the Vouch location proof scheme [58]. The idea of Vouch is to use the built-in ability of fifth generation cellular networks to locate mobile clients independently of the information provided by the client. Vouch uses RSUs to provide trusted location proofs for vehicles. A proof essentially is digitally signed data that enables a vehicle to attest its position to neighbors in a secure and trusted manner. When neighbor vehicles broadcast beacons, Vouch employs a plausibility model to classify the received positions according to the proofs that have been disseminated by those entities. However, this approach needs a 5G network to be deployed and is therefore not compatible with current standards. Moreover, it does not satisfy the non-tracking requirement.

Hamdan et al. [59] proposed a detection approach that relies on the idea that an RSU can link pseudonym certificates belonging to the same vehicle via the use of hashing algorithms. Thus, for the detection, the RSU monitors all the traffic messages and tries to find two or more certificates belonging to the same vehicle. To confirm if this result is valid for an attack, the proposed algorithm depends on the trajectory of the vehicle to check whether it is a sybil node or a genuine node. To check the trajectory of the vehicle, each station has a series of link-tags that is obtained by each RSU that is passing by. Thus, the vehicles have a series of link-tags. The sybil attack detection is done by checking this series of link-tag. If two vehicles have the same series of link-tags then a sybil attack is happening. However, this approach suffers from privacy issues because it provides a method to link a vehicle's pseudonym certificates. It also provides a method to create vehicles' trajectories which allow their tracking. Moreover, the approach is not effective in an urban scenario where numerous vehicles share the same trajectories.

Iwendi et al. [60] proposed a biologically inspired spider-monkey time synchronization technique for large-scale VANETs to improve packet delivery time synchronization with low energy consumption. The proposed technique is based on the metaheuristic stimulated framework approach by the natural spider-monkey behavior [61]. An artificial spider-monkey technique [60] is used to examine the sybil attack strategies on VANETs to predict the number of vehicular collisions in a densely deployed challenge zone. However, this approach assumes that when an RSU fails to synchronize its clock with legitimate vehicles, then this is the main cause for a sybil attack. Thus, they focus on synchronizing the RSU clock with the vehicles to avoid attacks. Nonetheless, by using different legitimate pseudonym certificates, an attacker can still launch an attack without affecting the clock.

in [62] *Anwar et al.* proposed a cloud-based detection scheme for connected vehicles against sybil attacks. The scheme integrates a cloud-based authorization unit to authenticate legitimate nodes using symmetric cryptography and enables real-time location tracking. However, this approach does not meet the current ITS standards for PKI. Furthermore, it relies on creating a history of the vehicles paths which allows their tracking.

Yang et al. [63] proposed a classifier to detect sybil attackers according to their mobility behaviors. Specifically, three levels of sybil attackers are first defined according to their attack abilities. By analyzing the mobility behaviors of vehicles, a learning-based model is used in the Central Server (CS) to extract mobility features and distinguish sybil attackers from benign vehicles. Nonetheless, this approach relies on a centralized server to enable the detection process which can make it a communication bottleneck as well as reduce its scalability.

Ressource testing

The resource testing approach assumes that physical entities are limited in resources such as computation, storage, and radio channels. Thus, in this approach, a typical puzzle is given to all stations to evaluate their resource availability. If one station is used to create and simulate multiple entities, then, it will be limited in responding to all puzzles.

This technique was mainly used in detecting sybil attacks in Mobile Ad hoc Networks (MANET) [64] and Sensor networks [65]. However, it is not suitable for a heterogeneous environment [50] such as C-ITS. Furthermore, an attacker can easily have more computational resources compared to legitimate nodes or have more radio transmitters [19]. Finally, this technique involves a high number of requests/responses which can cause network congestions in a lossy environment, where communication exchanges must be minimized.

In [66], *Raj et al.* proposed a detection method that relies on proofs of work and location. The main goal here is that when a vehicle encounters an RSU, it will be authorized by a timestamped tag which is a concatenation of time of appearance and the anonymous location tag of that RSU. As the vehicle keeps moving, it creates its trajectory by incorporating a set of consecutive authorized timestamped tags that are chronologically chained to each other. This trajectory is used as an anonymous identity of the vehicle. Hence RSUs have the main authority to provide proof of location to vehicles. Moreover, threshold signature [67] is adopted so that each RSU is only able to generate a partial signature on a set of timestamped tags. If a vehicle travels along a valid threshold number of RSUs, a standard signature representing a proof of location can be generated. Upon receiving an authorized message from an RSU, the vehicle should use it as a seed to solve a puzzle using a proof-of-work algorithm, similar to the one used in Bitcoin. The core idea of Proof of Work is to provide a proof to RSUs so that they can ensure that the vehicle solves the puzzle correctly. However, this technique is greedy in computational power and energy. Moreover, continuously computing Proof of Work (PoW) to validate vehicles' positions and their messages limits the scalability of the network.

Similarly, *Baza et al.* [68] proposed a sybil attack detection scheme using proofs of work and location. The idea is that each RSU issues a signed time-stamped tag as a proof for the vehicle's anonymous location. Proofs sent from multiple consecutive RSUs are used to create the vehicle trajectory which is used as an anonymous identity for the vehicle. Also, one RSU cannot issue trajectories for vehicles. Instead, several RSUs are needed. In this way, attackers need to compromise a large number of RSUs to create fake trajectories. Moreover, upon receiving the proof of location from an RSU, the vehicle should solve a computational puzzle by running PoW algorithm. So, it should provide a valid solution (proof of work) to the next RSU before it can obtain a proof of location. Using the PoW can prevent the vehicles from creating multiple trajectories in case of low-dense RSUs. Then, during any reported event, (e.g., road congestion) the event manager uses a matching technique to identify the trajectories sent from sybil vehicles. The scheme depends on the fact that the sybil trajectories are physically bounded to one vehicle. Thus, their trajectories should overlap. The proposed approach suffers from two main shortcomings: (1) the continuous computation of the proof of work, a hard computational puzzle, by vehicles which are considered as constrained devices, and (2) the approach relies on the history of vehicles, presented by their proofs of location provided by the RSUs which is contrary to the requirements (Req 3 and Req 4) discussed above and that requires that the privacy and non-tracking of vehicles must be respected.

Public cryptography based approaches

There have been some solutions such as [69] [70] that rely on symmetric cryptography.

	Requires additional hardware?	Considers C-ITS standards?	Feasible on current C-ITS systems?	Data superposition on digital map?	Allows tracking?	Scalability	Was there an evaluation?	Eval. on 1 road or on map?	Eval data type?	Eval. environment type	Sybil generation algo provided?	Year
Piro et al. [64]	No	No	No	No	Yes	No	Yes	/	Synthetic	/	No	2006
Xiao et al. [44]	Yes	No	Yes	Yes	No	No	Yes	1 road	Synthetic	/	No	2006
Yu et al. [19]	Yes	No	Yes	Yes	No	No	Yes	1 road	Synthetic	/	No	2013
Golle et al. [45]	Yes	No	No	No	No	No	No	/	/	/	/	2004
Rabieh et al. [46]	Yes	Yes	Yes	Yes	No	No	Yes	1 road	Synthetic	/	No	2015
Yan et al. [47]	Yes	No	No	Yes	Yes	No	Yes	1 road	Synthetic	/	No	2008
Chang et al. [50]	No	No	No	No	Yes	No	Yes	Map	Synthetic	Urban	No	2012
Hussain et al. [52]	No	No	No	No	No	No	No	/	/	/	/	2012
Park et al. [18]	No	No	No	No	No	No	No	/	/	/	/	2009
Chen et al. [42]	No	No	No	No	Yes	No	Yes	Map	Synthetic	Urban	No	2009
Grover et al. [53]	No	Yes	Yes	No	No	No	Yes	Map	Synthetic	Urban / Highway	No	2011
Grover et al. [71]	No	Yes	Yes	No	No	No	Yes	Map	Synthetic	Urban / Highway	Yes	2014
Sowattana et al. [29]	No	No	No	Yes	No	Yes	Yes	1 road	Synthetic	Highway	No	2017
Gu et al. [43]	No	Yes	Yes	No	No	Yes	Yes	1 road	Synthetic	Urban	Yes	2016
Bißmeyer et al. [56]	No	No	Yes	No	Yes	Yes	Yes	/	Synthetic	/	No	2012
Rahbari et al. [41]	No	No	No	No	No	No	Yes	/	/	/	No	2011
Jin et al. [72]	Yes	Yes	No	Yes	No	No	Yes	1 road	Synthetic	Highway	No	2014
Abu-Elkheir et al. [73]	Yes	Yes	Yes	Yes	No	No	Yes	Map	Synthetic	Urban	Yes	2011
Naveed et al. [74]	Yes	Yes	No	No	No	No	Yes	1 road	Synthetic	Highway	No	2015
Naveed et al. [75]	Yes	Yes	No	No	No	No	Yes	1 road	Synthetic	Highway	No	2015
Murugan et al. [76]	No	No	No	No	Yes	No	Yes	/	Synthetic	/	No	2015
Zhou et al. [77]	Yes	No	Yes	No	No	No	Yes	1 road	Synthetic	/	Yes	2011
El Zoghby et al. [78]	No	Yes	Yes	No	No	No	Yes	1 road	Synthetic	Highway	Yes	2012
De Sales et al. [79]	No	No	No	No	No	No	No	/	/	/	/	2014
Hao et al. [80]	Yes	Yes	Yes	Yes	No	No	Yes	1 road	Synthetic	Highway	No	2011
Feng et al. [81]	No	Yes	Yes	No	No	No	Yes	/	Synthetic	/	No	2017
Lal et al. [82]	Yes	No	Yes	No	No	No	Yes	1 road	Synthetic	/	Yes	2015
Bouassida et al. [83]	No	No	Yes	Yes	Yes	No	Yes	/	Synthetic / Real	/	No	2009
Yao et al. [10][84]	Yes	Yes	Yes	Yes	No	No	Yes	1 road / Map	Synthetic / Real	Urban / Highway	Yes	2018
Benkirane et al. [17]	Yes	No	Yes	No	No	Yes	Yes	1 road	Synthetic	/	No	2019
Ayaida et al. [57]	No	Yes	Yes	No	No	Yes	Yes	1 road	Synthetic	/	No	2019
Khalil et al. [85]	No	No	No	No	Yes	No	Yes	/	Synthetic	/	No	2020
Baza et al. [68]	No	Yes	Yes	No	Yes	Yes	Yes	Map	Synthetic	Urban / Highway	No	2020
Hamdan et al. [59]	No	No	No	No	Yes	Yes	Yes	1 road	Synthetic	/	No	2019
Iwendi et al. [60]	No	No	Yes	No	No	Yes	Yes	/	Synthetic	/	No	2018
Anwar et al. [62]	No	No	Yes	No	Yes	No	Yes	1 road	Synthetic	/	No	2019
Parham et al. [69]	No	No	Yes	No	Yes	Yes	Yes	Map	Real	Urban	No	2020
Boeira et al. [30]	Yes	No	No	No	Yes	Yes	Yes	1 road	Synthetic	/	No	2018
Kabbur et al. [49]	Yes	No	No	No	Yes	No	Yes	Map	Synthetic	/	No	2020
Trauernicht et al. [86]	No	Yes	Yes	No	No	Yes	Yes	1 road	Synthetic	/	No	2019
Khalil et al. [70]	No	No	No	No	Yes	No	Yes	/	Synthetic	/	No	2018
Lim et al. [87]	No	No	No	Yes	Yes	No	Yes	1 road	Synthetic	/	No	2020
Raj et al. [66]	No	No	No	No	No	No	No	/	/	/	/	2020
Kamel et al. [22]	No	Yes	Yes/ No	No	No	/	Yes	Map	Synthetic	Urban	No	2019
Yang et al. [63]	No	Yes	Yes	No	No	No	Yes	/	Real	Urban	No	2018

TABLE I: Summary of sybil detection approaches; Yes: supported; No: Not supported. Green color is used for a suitable feature and the red color is for an unsuitable feature.

However, symmetric cryptography is not well-suited to support scalability and privacy requirements in such an environment. In public cryptography based approaches, certificates are provided to stations for authentication purposes during communications. However, for privacy and non-tracking purposes, these certificates must be changed continuously. Moreover, due to the lossy features of the network, the stations must download a set of certificates in advance. Having numerous active certificates in the same time facilitates the sybil attack. To the best of our knowledge, only the proposal named Issue First Activate Later (IFAL)[88], have addressed this issue. IFAL provides the station with only one valid certificate at a given time. However, in this case, all the station's certificates are easily linkable, which is in contrary with the non-tracking requirement.

Trauernicht et al. [86] proposed a concept for the long-term exclusion of sybil attackers based on a deterministic mechanism called Sybil attack Alternation Check (SAC). The key idea of the approach is to forbid the reuse of a recently used pseudonym certificate for a short time period. Based on this rule proof, an attack can be recorded and reported to invalidate certificates of the misbehaving station.

Kamel et al. [22] proposed the integration of a Misbehavior Authority (MA) into the PKI structure. The MA attempts to link the pseudonyms related to the same reported physical ITSS. If no link is found, the process is complete and the misbehavior type is returned. If a link is found, then a sybil attack is suspected and the linked pseudonyms are candidates for sybil attack type detection in the next phase. In this phase the linked pseudonyms are treated as one and the evidences collected from all the linked pseudonyms are used in a specific sybil type detection process. This approach can be easily integrated with IEEE based PKIs because the IEEE architecture comprises a linkage authority as well as a misbehavior authority. However, the ETSI based architecture does not have a linkage authority nor a misbehavior authority.

Thus, the implementation of this approach incurs additional work of modifying the existing PKI architecture. Besides, this approach relies on linking pseudonym certificates to find those of the same vehicle. As a result, it is not sufficient against sybil attacks where the attacker replays other vehicles' messages.

Summary and discussion

Table I presents the works discussed above and other existing ones. The table does not provide a comparison or a classification of the approaches and their performances as they have been presented already in past surveys [16][9][23][24][25]. However, the table discusses the features related to their adoption and deployment in the current context and deployment projects. We highlighted in green what we think are positive aspects and in red negative ones.

The first columns [1-6] of the table describe the ability of the approach in meeting some needed requirements such as: (1) their consideration of C-ITS standards and their feasibility on current systems. Indeed, several works do not take into account the current standards that each project must respect, regarding the sent and received messages' formats, security, types, and frequency. It is worth noting that numerous approaches are technically feasible on current systems. However they do not satisfy the communication standards as they were published before the current standards were adopted. Thus, they cannot be considered as solutions by current and future deployment projects based on these standards. (2) Even if we add additional hardware such as antennas, lidars or radars to enhance the localization and detection accuracy, we will incur additional costs. Indeed, even if the majority of current vehicles are embedded with radars, to be efficient, the existing approaches require a certain number of directional antennas to send and receive targeted data, which will increase costs. Also, adding these types of hardware leads to additional computational costs, due to additional data processing on the collected data [89]. (3) Several approaches have assumed that the vehicle has only one identity or rely on the linking of the different vehicle's identities without resorting to a dedicated Linkage Authority, which can allow its tracking. But it is prohibited by current standards and such approaches cannot be adopted. (4) Multiple approaches consider the position of the vehicle to achieve the detection. This requirement implies the usage of a digital map (or its equivalent with road boundaries coordinates) in order to overlay the vehicle's positions on these maps, because if the sybil identities are forged and not replayed, they can be out of roads' boundaries. This feature is considered in position verification approaches, even if it is not mentioned. Moreover, this positions' overlay on maps is implicitly managed by simulation tools. However, in real life systems, adding such a layer can lead to additional costs. (5) ITSs are considered as the first use case scenario in smart cities [90]. The number of connected vehicles and devices related to ITS scenarios is in exponential growth [91][92]. Hence, the detection solution deployed must be scalable to support such a load. In the table provided, we consider a solution to be scalable if it is not centralized, does not introduce additional messages, and does not incur high processing overheads from the vehicle's side.

The rest of the columns in Table I are dedicated to how the discussed approaches were evaluated. Indeed, the decision of a solution's adoption mainly lies on its performances' results although the results can differ according to the methodology followed for the evaluation. All the works presented relied on simulation tools for their evaluations. The majority of them used synthetic data. We think that using real datasets can produce more accurate indicators about the evaluated approach's efficiency. Moreover, the majority of works were evaluated on the scenario of one road. In contrast, in real life scenarios, a vehicle can have neighbors along the same road or along parallel roads due to the coverage range of stations ($\approx 1Km$), especially in urban scenarios, which may mislead the detection. Without a digital map position superposition, the vehicle cannot know if its neighbor is along same road or not because the roads are not always straight especially in urban areas. We discussed earlier the limits of considering a digital map in the detection process. Consequently, we argue that an evaluation must consider a scenario that contains adjacent roads with different topologies and not just different lanes along the same road. Finally, the majority of the proposed solutions are evaluated regarding one sybil scenario. But there are numerous attack models and possible ways [9][81][22] to launch a sybil attack. Thus, we argue that providing the sybil generation algorithm will improve the understanding of the solution.

From the Table I and the analysis above, we can conclude that **most the proposed detection solutions cannot be deployed in current C-ITS systems mainly because: (1) they are not scalable; (2) they do not meet privacy and non-tracking requirements; (3) they do not satisfy the requirements of current standards especially regarding the formats of messages, security, and PKI architecture and (4) of their limited evaluations where only a few use case scenarios (such as single lane) were tested.**

III. Research Methodology

In this section we propose, according to the analysis of the different existing works, a methodology that allows a better evaluation of the sybil detection approaches.

A. Network model

The overall purpose of a security scheme is to allow multiple nodes to communicate in a trustworthy way over a non-trusted network. In this work we consider a network that owns a set of ITSS offering and using different ITS services in a centralized or a distributed architecture. Each ITSS communicates with a large number of other ITSSs. Exchanged messages pass through an unreliable and potentially lossy communication network, such as 802.11p or ITS-G5. We also assume that all participants cannot be trusted. Indeed, a high number of stations in the network increases the risk of including compromised ones. Furthermore, the existing stations are of heterogeneous types.

According to the standards, the network implements a PKI. The latter comprises a Long Term Certificate Authority and a Pseudonym Certificate Authority that supply ITSSs with certificates. The ITSSs never use the LTC for communication but only to authenticate to the PKI in order to request new PCs. However, the PCs are continuously used because each packet must be signed by a private key associated with a public key certified by a PC. To comply with the privacy (and non-tracking) requirements, each ITSS must change its PC as well as all the network identifiers (e.g., IP address, MAC address, station ID, and so on) multiple times during a trip. The PCs of a given ITSS can only be linkable by dedicated authorities (e.g., the Linkage Authority) and cannot be linkable by other stations.

The network function only forwards packets and does not provide any security guarantee such as integrity or authentication. Thus, a malicious user can read, modify, drop or inject network messages.

Algorithm 1: Basic operations of an attacker

```
Function CollectPackets () : List of Packet
    // Sniffs the network and collects all the packets
Function ChooseRandomPackets (List of Packet $lPkt$, Integer $ghostsPercentage$) : List of Packet
    // Provides the needed number of fake packets for the sybil attack, e.g., if there are 100 vehicles and the attacker needs to simulate 10% vehicles for the sybil attack, this function will choose randomly 10 packets
Function CreatePackets (Integer $ghostsPercentage$) : List of Packet
    // Creates the needed number of fake packets for the sybil attack. e.g., if there are 100 vehicles and the attacker needs to simulate 10% vehicles for the sybil attack, this function will create 10 packets
Function Broadcast (List of Packet $ghosts$) : Void
    // Broadcasts the messages of the sybil attack
Function Sign (Packet $pkt$, PrivateKey $privKey$ ) : Packet
    // Signs a packet
Function PacketWithoutSignature (Packet $pkt$) : Packet
    // Creates the same packet but without the signature field
Function RandomCoord (Packet $pkt$, List of Packet $ObservedNetwork$) : Packet
    // Replaces the Longitude and Latitude values in the packet by random values in the coverage area. the attacker considers values existing in other sniffed genuine packets
Function RandomSpeed (Packet $pkt$, List of Packet $ObservedNetwork$) : Packet
    // Replaces the speed value in the packet by a random value. This value must be between the maximum and minimum speed values observed in the genuine packets
Function RandomAcceleration (Packet $pkt$, List of Packet $ObservedNetwork$) : Packet
    // Replaces the acceleration value in the packet by a random value. This value must be between the maximum and minimum acceleration values observed in the genuine packets
Function NewTimestamp (Packet $pkt$) : Packet
    // Replaces the timestamp value in the packet by the current timestamp
Function StaticSpeedAndAcceleration (Packet $pkt$) : Packet
    // Modifies the speed and acceleration fields of a packet to zero
```

B. Attacker model

The C-ITS environment relies on wireless communications. Therefore, in this work, we assume that an attacker or malicious user has total control over the network used, i.e., the attacker can selectively sniff, drop, replay, reorder and delay messages arbitrarily with negligible delay. We also assume that the attacker has a pool of valid PCs. For instance, the attacker can obtain these PCs by tampering with the storage device of an ITSS. Besides, the attacker can benefit from increased computation power and storage than the existing devices.

All the messages are signed. Thus, the attacker cannot modify existing messages. However, since the attacker has a set of valid certificates the attacker can change the signature and modify the fields as needed, or can create new packets. Knowing that the certificates are pseudonym identities and are not linkable, the majority of the receiving entities (stations and services) will not notice that these are packets sent from an attacker.

Within a network, devices can receive unaltered and altered messages. Thus, for a better evaluation of sybil detection proposals, we recommend that the researchers evaluate their detection approach regarding different rates of altered messages. For example, they evaluate their approach when the network comprises 10% of additional sybil ghosts. Then, for 20%, 30% and so on. We believe that this method will accurately reflect the efficacy of a detection system with respect to the number of sybil ghosts.

We also recommend to evaluate the detection proposals against different sybil scenarios with different difficulties. Hence, in this work we propose different scenarios (we assume that the attacker provides basic protocol primitives to execute attacks as we have described above. Algorithm 1 depicts such an Application Programming Interface (API)):

1) Sybil scenario with random values: this scenario represents the case where the attacker does not replay captured packets, but just forge new packets, with random values in the fields and broadcast them. Algorithm 2 describes this scenario. As described by [22], this scenario can be used to launch a Denial of Service (DoS) attack where the motivation behind such an attack could be to overwhelm the misbehavior detection system of neighboring ITSS or that of the platform or just to disturb the network's communications. Moreover, it is one of the most commonly used scenarios for the evaluation of numerous approaches discussed in Section II. Thus, to be stealthy than the discussed approaches, we propose that the attacker does not generate any random data that can make the detection easy. Instead, we propose that the attacker uses the same geolocalisation data as other captured packets. Moreover, for the speed, acceleration and so on, the attacker can use a random value between the maximum and minimum values that he observes on the network during the attack period.

2) Sybil scenario with static values: this scenario represents the case where the attacker simulates a traffic congestion. To be stealthy, the attacker captures some packets in the targeted area, then changes some fields in the packets (e.g., sets the speed and the acceleration to zero and modifies the signature and the timestamp). Next, the attacker broadcasts these modified packets and repeats the process of changing the signature but without changing the coordinates, heading, and so on, until the end of the attack. Algorithm 3 describes this scenario.

3) Sybil scenario with replayed values: in this scenario, the attacker continuously captures traffic packets, changes their signatures, and timestamps, but keeps their movement data such as coordinates, speed, heading, acceleration and so on and broadcasts them. This scenario can be considered as the highest difficulty level for a detection scheme because it uses a realistic traffic model. Algorithm 4 depicts this scenario.

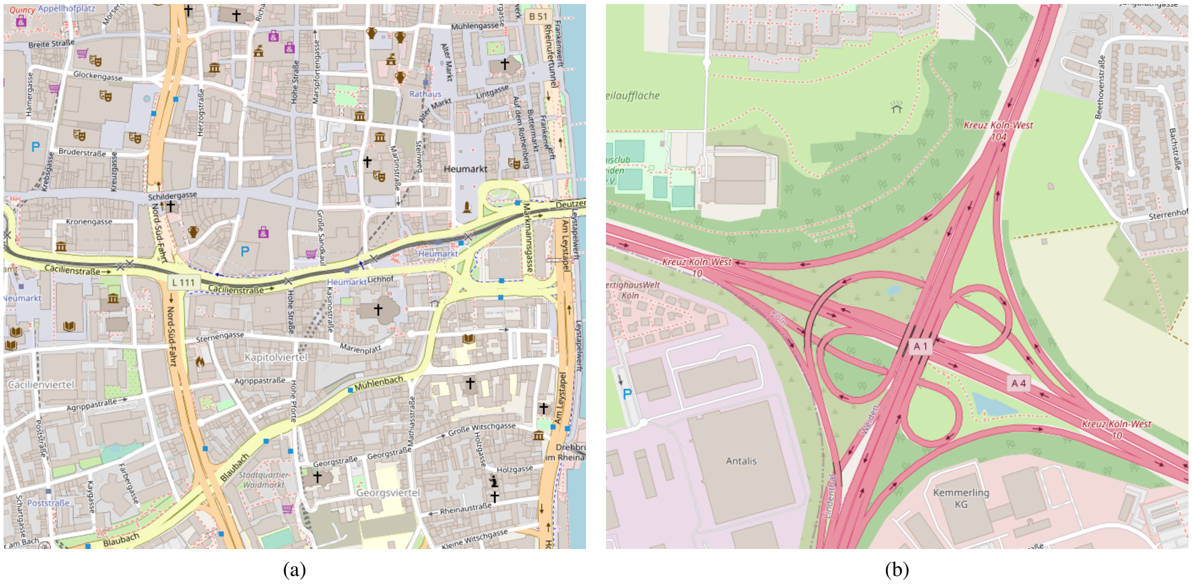
Finally, we recommend that researchers evaluate their approaches in both urban and highway scenarios. Moreover, we recommend, whenever possible, the use of real data. Other-

Algorithm 2: Sybil scenario: random values

```
attackerPosition : Coordinates // Coordinates
    (Longitude and Latitude) of the attacker
coverage : Integer // Coverage zone of the
    attacker
CertificateStructure : SEQUENCE { certificate: Certificate;
    privateKey: Integer } // A structure that
    contains a certificate and its private key
certificateStructures : List of CertificateStructure
    // List of certificates that the attacker
    will use during the sybil attack
ghostPercentage: Integer // Percentage of the
    additional sybil ghosts to add
lPkt: List of Packet
chosenList: List of Packet
ghosts: List of Packet
ghost: Packet
begin
    while AttackIsOngoing() do
        ObservedNetwork ← CollectPackets ()
        lPkt ← CreatePackets (ghostPercentage)
        foreach ghost In lPkt do
            ghost ← RandomCoord (ghost,
             ObservedNetwork)
            ghost ← RandomSpeed (ghost,
             ObservedNetwork)
            ghost ← RandomAcceleration (ghost,
             ObservedNetwork)
            ghost ← NewTimestamp (ghost)
            ghost ← Sign (ghost,
             certificateStructures.Next ().privateKey)
            ghosts.Append (ghost)
        Broadcast (ghosts)
```

Algorithm 3: Sybil scenario: static values

```
CertificateStructure : SEQUENCE { certificate: Certificate;
privateKey: Integer }
certificateStructures : List of CertificateStructure
ghostPercentage: Integer
lPkt: List of Packet
chosenList: List of Packet
ghosts: List of Packet
packet: Packet
ghost: Packet
begin
    lPkt ← CollectPackets ()
    chosenList ← ChooseRandomPackets (lPkt,
     ghostPercentage)
    while AttackIsOngoing() do
        foreach packet In chosenList do
            ghost ← PacketWithoutSignature
             (packet)
            ghost ← StaticSpeedAndAcceleration
             (packet)
            ghost ← NewTimestamp (ghost)
            ghost ← Sign (ghost,
             certificateStructures.Next ().privateKey)
            ghosts.Append (ghost)
        Broadcast (ghosts)
```



(a)

(b)

Fig. 2: (a) Map of the urban dataset (b) Map of the highway dataset

Algorithm 4: Sybil scenario: replayed values

```
$CertificateStructure$ : SEQUENCE { $certificate$: Certificate; $privateKey$: Integer }
$certificateStructures$ : List of CertificateStructure
$ghostPercentage$: Integer
$lPkt$: List of Packet
$chosenList$: List of Packet
$ghosts$: List of Packet
$packet$: Packet
begin
    while $AttackIsOngoing()$ do
        $lPkt \leftarrow$ CollectPackets ()
        $chosenList \leftarrow$ ChooseRandomPackets ($lPkt$, $ghostPercentage$)
        foreach $packet$ In $chosenList$ do
            $ghost \leftarrow$ PacketWithoutSignature ($packet$)
            $ghost \leftarrow$ NewTimestamp ($ghost$)
            $ghost \leftarrow$ Sign ($ghost$, $certificateStructures$.Next ().$privateKey$)
            $ghosts$.Append ($ghost$)
        Broadcast ($ghosts$)
```

wise, synthetic data generated according to realistic traffic models. In this context, we provide two datasets[5] that can be used by researchers to evaluate their detection solutions. These datasets are retrieved from the datasets made available by the *TAPASCologne* initiative[6] of the Institute of Transportation Systems at the German Aerospace Center (ITS-DLR), that aims at reproducing car traffic in the greater urban area of the city of Köln, Germany, with the highest level of realism possible.

The *TAPASCologne* provides mobility data of the city of Köln which may represent a far too large test area. Thus, to facilitate the task for researchers, we extracted two datasets, each on 1 Km^2 and for 24 hours. The first dataset that we provide presents an urban scenario that describes the activity of 62,421 vehicles, and the second presents a highway scenario that describes the activity of 24,326 vehicles. Figure 2.a shows the area[7] from which the urban dataset was extracted and Figure 2.b highlights the area from which the highway dataset was extracted.

C. Summary

We summarize our recommendations that will help better evaluate sybil detection proposals as follows:

1) The use of a realistic network model where a C-ITS PKI is deployed.
2) The ability to meet the privacy and non-tracking requirements by considering the continuous change of the identification data such as the PC, the IP address, the MAC addresses and so on.
3) The use of real mobility data or at least synthetic data generated according to realistic traffic models.
4) The evaluation of the detection approach regarding different rates of altered messages (e.g., 10%, 20%, 30% and so on).
5) The evaluation of the detection approach against different sybil scenarios (in this work, we proposed three scenarios).

IV. Characterization of the Sybil Dataset

Many researchers believe that big data represent a promising solution to numerous C-ITS problems [93] [94]. In this section,

[5] https://github.com/BadisHammi/C-ITS_Datasets

[6] https://sumo.dlr.de/docs/Data/Scenarios/TAPASCologne.html

[7] from openstreetmap.org

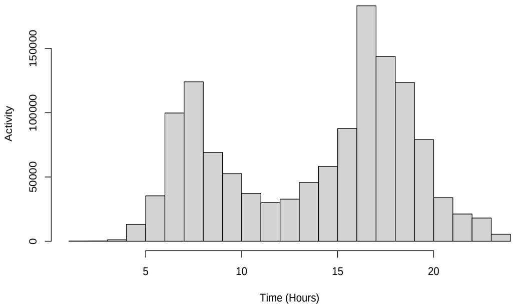


Fig. 3: Histogram of the vehicles activity (number of messages) collected in the 24 hours

we provide a quick statistical characterization of the dataset that we provide after considering the different sybil scenarios in order to prove the efficiency of our attack scenarios in generating realistic sybil data and to show the difficulty of detecting such attacks. More precisely, we describe three experimentations. For each experimentation, we used the highway dataset, to generate a sybil attack that adds 10% of sybil ghosts to the traffic. Each experimentation corresponds to one of the attack scenarios described in Section III-B.

Figure 3 presents the activity of the vehicles of our highway dataset regarding the number of messages for each hour. For visibility purposes we limit our experiences to the hour between 11:00 am and 12:00 am because it represents the weakest activity. Thus, we only consider the networks' data of this period. Also, the sybil attack scenarios are executed during this period.

For this characterization, we used the Principal Component Analysis. Principal Component Analysis (PCA) [95] is a descriptive statistical method belonging to the factorial category. It is aimed at easing the exploration and analysis of high-dimensional vectors of input data by reducing their dimensions and enabling the extraction of effective features. Given a data matrix of n observations, also called individuals, composed of p variables, PCA describes the variance-covariance structure of the set of variables through a few new variables, called principal components or factors, which are functions of the original variables. Principal components represent linear combinations of the p variables with important properties: the computed principal components, which are in general 2 or 3, have the highest variances so that they best represent the data in a reduced dimension space and highlight their linear relations. Also, components are uncorrelated and the total variance of all the principal components is equal to the total variance in the original variables.

Apart from reduction of data dimensions, PCA is also used for simplification, data reduction, modeling, variable selection, classification, prediction, and outlier detection. Many other works such as in [96][97][98][99] in network intrusion detection relied on PCA, .

In this work we use the individuals projection method of PCA (individual scatter plot). Indeed, as explained above, the PCA creates new individuals, where each of these represents a linear combination of the p parameters collected in the same given instant. In other words, each line of the initial matrix $(x_1, x_2, ..., x_p)$ will be transformed into one individual that represents the linear combination of the other p values. Thus, if the set of data relative to the attack has different statistical characteristics from the genuine data, it will create a separate cluster when projected on the new PCA factors. However, if this data is not easily characterizable, then, it will be mixed with the other data of the network.

Figure 4.a presents the PCA's individual scatter plot of the experimentation that involves the static sybil scenario. In this case, we note that the ellipse that surrounds the sybil ghosts cluster is more concentrated around the origin O than the ellipse of the genuine nodes cluster. We also note the superposition of different sybil ghosts because of the replay of the same data. Finally we observe that the sybil nodes are not easily characterizable compared to genuine nodes. We recall that this case describes a highway mobility scenario and if the same attack were executed in an urban scenario where numerous genuine vehicles can also be static for a defined time, then the sybil ghosts will be less characterizable than in the presented highway mobility scenario.

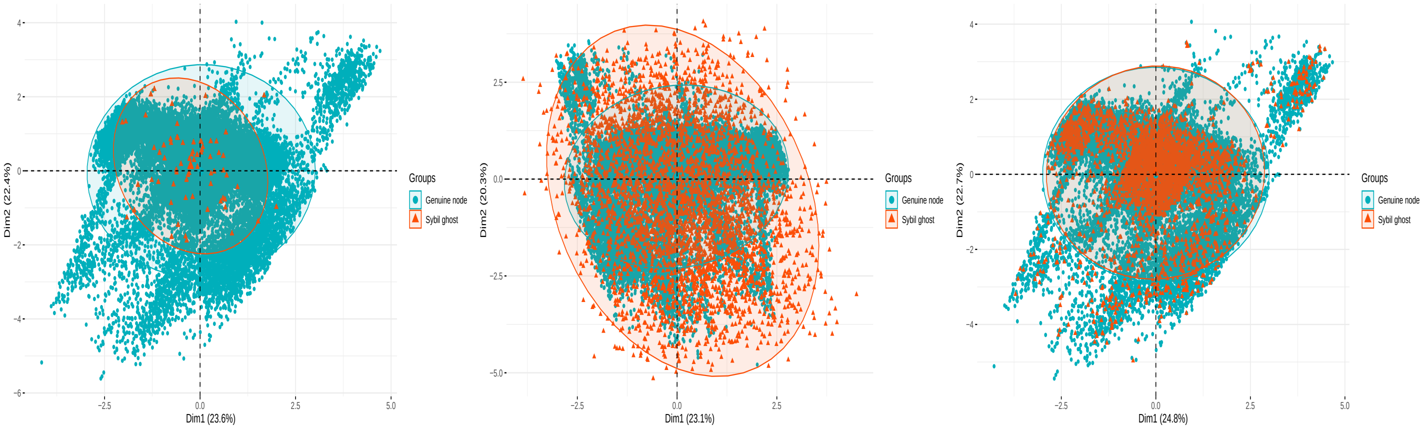
Figure 4.b and Figure 4.c describe the PCA's individual scatter plots of the random and replayed experimentation scenarios respectively. In both scenarios the sybil ghosts are mixed with genuine nodes. We also note in the replayed scenario that the ellipse that surrounds the sybil ghosts cluster perfectly overlays the ellipse of the genuine nodes cluster .

For the three presented scenarios, we observe that the sybil ghosts are completely mixed with the genuine vehicles and cannot be detected which demonstrate the efficiency and how realistic the given scenarios are.

Given the last characterization, we recommend that future proposals evaluate their approaches for the different scenarios that we propose in order to have a better idea about the detection approach's efficiency and not to evaluate it using only the random scenario because it is the case with the majority of existing works.

V. Conclusion and future works

Cooperative Intelligent Transportation Systems play a vital role in our daily life. They improve traffic safety and our driving experience. However, these benefits are subjected to different security issues and attacks. In this work we considered sybil attacks which are considered among the most dangerous ones. We provided a comprehensive survey on the different sybil detection approaches that have been recently proposed. We analyzed their compliance with the different existing network and security standards adopted and deployed by the C-ITS. We found that the majority of these proposals are not compatible with the current C-ITS context and cannot be deployed mainly due to: (1) their limits on scalability; (2) the fact that they do not satisfy privacy and non-tracking



(a)

(b)

(c)

Fig. 4: PCA's individual scatter plot: (a) Static Scenario; (b) Random scenario; (c) Replayed scenario

requirements; (3) their incompatibility with current standards especially regarding messages formats, security and PKI architecture and (4) their limited evaluations as only a few use case scenarios such as the use of single lane were tested, which makes it hard to assess their efficacies in real systems. There are other challenges that a sybil detection system must address. Indeed, vehicles from different manufacturers or countries will likely get their credentials from different PKIs. Therefore, the identities of the nodes managed by a detection system can be heterogeneous, which make the detection process more complex. Another challenge is dealing with revoked credentials. Indeed, the detection process must work closely with the authentication process to achieve efficient and optimal detection.

In this work, we provided a network model, an attacker model that comprises three attack scenarios, a set of recommendations and two datasets, urban and highway to help further research. The statistical characterization provided demonstrates the feasibility and efficiency of our attacker model.

This paper is the first step of our work which will propose a fully distributed sybil detection approach that can address scalability issues and seamlessly integrate with different security standards of C-ITS environments.

Acknowledgments

We thank the anonymous reviewers for their valuable comments which helped us improve the content, quality, and presentation of this paper. Sherali Zeadally was supported by a Fulbright U.S. scholar grant award administered by the U.S. Department of State Bureau of Educational and Cultural Affairs, and through its cooperating agency the Institute of International Education ("IIE").

References

[1] ETSI TS 102 636-1 V1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 1: Requirements. March 2010.

[2] ETSI TS 102 636-3 V1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network architecture. March 2010.

[3] John B Kenney. Dedicated short-range communications (DSRC) standards in the United States. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[4] Andreas Festag. Standards for vehicular communication—from IEEE 802.11 p to 5G. *e & i Elektrotechnik und Informationstechnik*, 132(7):409–416, 2015.

[5] Ragunathan (Raj) Rajkumar, Insup Lee, Lui Sha, and John Stankovic. Cyber-physical systems: The next computing revolution. In *Proceedings of the 47th Design Automation Conference*, DAC '10, pages 731–736. ACM, 2010.

[6] Security Credential Management System Proof-of-Concept Implementation. EE Requirements and Specifications Supporting SCMS Software Release 1.1. Technical report, Vehicle Safety Communications 5 Consortium, May 2016.

[7] Monteuuis JP, Hammi Badis, Salles Eduardo, Labiod Houda, Blancher Reemi, Abalea Erwan, and Lonc Brigitte. Securing PKI Requests for C-ITS systems. In *3rd International Workshop on Vehicular Networking and Intelligent Transportation Systems (VENITS 2017)*, page 8. IEEE, 2017.

[8] Juan Guerrero-Ibáñez, Sherali Zeadally, and Juan Contreras-Castillo. Sensor technologies for intelligent transportation systems. *Sensors*, 18(4):1212, 2018.

[9] Fatih Sakiz and Sevil Sen. A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov. *Ad Hoc Networks*, 61:33–50, 2017.

[10] Yuan Yao, Bin Xiao, Gaofei Wu, Xue Liu, Zhiwen Yu, Kailong Zhang, and Xingshe Zhou. Multi-channel based sybil attack detection in vehicular ad hoc networks using rssi. *IEEE Transactions on Mobile Computing*, 2018.

[11] ETSI EN 302 637-2 V1.3.2: Intelligent Transport Systems (ITS), Vehicular Communications. Basic Set of Applications, Part 2: Specification of Cooperative Awareness Basic Service. November 2014.

[12] ETSI EN 302 637-3 V1.2.2: Intelligent Transport Systems (ITS), Vehicular Communications. Basic Set of Applications, Part 3: Specifications of Decentralized Environmental Notification Basic Service. November 2014.

[13] DSRC Committee. Dedicated short range communications (DSRC) message set dictionary. *SAE Standard J*, 2735:267, 2016.

[14] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn. A security credential management system for v2v communications. In *2013 IEEE Vehicular Networking Conference*, pages 1–8. IEEE, 2013.

[15] U.S. Department of Transportation. Reseach and Innovative Technology Administration. Vehicle-to-Vehicle (V2V) Communications for Safety. https://www.its.dot.gov/research/v2v.htm.

[16] Jyoti Grover, MS Gaur, and V Laxmi. Sybil attack in vanets. *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*, 269, 2010.

[17] Said Benkirane. Road safety against sybil attacks based on rsu collaboration in vanet environment. In *International Conference on Mobile, Secure, and Programmable Networking*, pages 163–172. Springer, 2019.

[18] Soyoung Park, Baber Aslam, Damla Turgut, and Cliff C Zou. Defense against sybil attack in vehicular ad hoc network based on roadside unit support. In *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pages 1–7. IEEE, 2009.

[19] Bo Yu, Cheng-Zhong Xu, and Bin Xiao. Detecting sybil attacks in vanets. *Journal of Parallel and Distributed Computing*, 73(6):746–756, 2013.

[20] Bryan Parno and Adrian Perrig. Challenges in securing vehicular networks. In *Workshop on hot topics in networks (HotNets-IV)*, pages 1–6, 2005.

[21] S. Abbas, M. Merabti, and D. Llewellyn-Jones. Signal Strength Based Sybil Attack Detection in Wireless Ad Hoc Networks. In *2009 Second International Conference on Developments in eSystems Engineering*, pages 190–195, 2009.

[22] Joseph Kamel, Farah Haidar, Ines Ben Jemaa, Arnaud Kaiser, Brigitte Lonc, and Pascal Urien. A Misbehavior Authority System for Sybil Attack Detection in C-ITS. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 1117–1123. IEEE, 2019.

[23] Yunpeng Zhang, Bidit Das, and Fengxiang Qiao. Sybil Attack Detection and Prevention in VANETs: A Survey. In *Proceedings of the Future Technologies Conference*, pages 762–779. Springer, 2020.

[24] NITHA C VELAYUDHAN, A ANITHA, MUKESH MADANAN, and VINCE PAUL. Review on avoiding sybil attack in vanet while operating in an urban environment. *Journal of Theoretical and Applied Information Technology*, 97(20), 2019.

[25] Zaid A Abdulkader, Azizol Abdullah, Mohd Taufik Abdullah, and Zuriati Ahmad Zukarnain. A survey on sybil attack detection in vehicular ad hoc networks (VANET). *Journal of Computers*, 29(2):1–6, 2018.

[26] Lan Lin and James A Misener. Message sets for vehicular communications. In *Vehicular ad hoc Networks*, pages 123–163. Springer, 2015.

[27] ETSI TS 103 301 v1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities Layer protocols and communication requirements for infratstructure services. *Technical specification, European Telecommunications Standards Institute*, page 42, November 2016.

[28] ETSI TS 102 894 v1.2.1-2: Intelligent Transport Systems (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary. *Technical specification, European Telecommunications Standards Institute*, page 94, November 2016.

[29] Chea Sowattana, Wantanee Viriyasitavat, and Assadarat Khurat. Distributed consensus-based sybil nodes detection in vanets. In *Computer Science and Software Engineering (JCSSE), 2017 14th International Joint Conference on*, pages 1–6. IEEE, 2017.

[30] Felipe Boeira, Mikael Asplund, and Marinho P Barcellos. Mitigating position falsification attacks in vehicular platooning. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–4. IEEE, 2018.

[31] Intelligent Transportation Systems Committee & others. IEEE Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages. *IEEE Vehicular Technology Society Standard*, 1609.2:1–884, January 2016.

[32] Vehicle-to-Vehicle Security Credential Management System; Request for Information. Technical report, National Highway Traffic Safety Administration (NHTSA), Department of Transportation (DOT), October 2014.

[33] ETSI TS 102 940: Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management. *Technical specification, European Telecommunications Standards Institute*, page 29, Jun 2012.

[34] ETSI TS 102 941: Intelligent Transport Systems (ITS); Trust and Privacy Management. *Technical specification, European Telecommunications Standards Institute*, page 71, May 2018.

[35] ETSI TS 102 731 v1.1.1: Intelligent Transport Systems (ITS); security; security services and architecture. *Technical specification, European Telecommunications Standards Institute*, page 68, 2010.

[36] ETSI TS 103 097 V1.2.1: Intelligent Transport Systems (ITS), Security header and certificate formats. page 35, June 2015.

[37] ETSI TS 103 097 V1.3.1: Intelligent Transport Systems (ITS), Security header and certificate formats. page 35, October 2017.

[38] Jonathan Petit, Florian Schaub, Michael Feiri, and Frank Kargl. Pseudonym schemes in vehicular networks: A survey. *IEEE communications surveys & tutorials*, 17(1):228–255, 2015.

[39] Virendra Kumar. Special Cryptographic Primitives in SCMS. SCP1: Butterfly Keys. https://wiki.campllc.org/display/SCP, March 2017.

[40] Security System: Integration Guide V4, SCOOP@F Delivrable 2.4.4.8. Technical report, December 2016.

[41] Mina Rahbari and Mohammad Ali Jabreil Jamali. Efficient detection of sybil attack based on cryptography in VANET. *arXiv preprint arXiv:1112.2257*, 2011.

[42] Chen Chen, Xin Wang, Weili Han, and Binyu Zang. A robust detection of the sybil attack in urban vanets. In *Distributed Computing Systems Workshops, 2009. ICDCS Workshops' 09. 29th IEEE International Conference on*, pages 270–276. IEEE, 2009.

[43] Pengwenlong Gu, Rida Khatoun, Youcef Begriche, and Ahmed Serhrouchni. Vehicle driving pattern based sybil attack detection. In *High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on*, pages 1282–1288. IEEE, 2016.

[44] B. Xiao, B. Yu, and C. Gao. Detection and localization of sybil nodes in VANETs. In *DIWANS'06 Proceedings of the 2006 Workshop on Dependability issues in Wireless Ad hoc Networks and Sensor Networks*, pages 1–8, Los Angeles, California, September 2006.

[45] Philippe Golle, Dan Greene, and Jessica Staddon. Detecting and correcting malicious data in vanets. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 29–37. ACM, 2004.

[46] K. Rabieh, M. M. E. A. Mahmoud, T. N. Guo, and M. Younis. Cross-layer scheme for detecting large-scale colluding sybil attacks in VANETs. In *IEEE International Conference on Communications (ICC)*, pages 7298–7303, London, UK, June 2015.

[47] Gongjun Yan, Stephan Olariu, and Michele C Weigle. Providing vanet security through active position detection. *Computer communications*, 31(12):2883–2897, 2008.

[48] Maxim Raya, Adel Aziz, and Jean-Pierre Hubaux. Efficient secure aggregation in vanets. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 67–75. ACM, 2006.

[49] Mahabaleshwar Kabbur and V Arul Kumar. MAR_Sybil: Cooperative RSU Based Detection and Prevention of Sybil Attacks in Routing Process of VANET. In *Journal of Physics: Conference Series*, volume 1427, page 012009. IOP Publishing, 2020.

[50] S. Chang, Y. Qi, H. Zhu, J. Zhao, and X. Shen. Footprint: Detecting sybil attacks in urban vehicular networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(6):1103–1114, June 2012.

[51] Connected Vehicle Roadside Unit (RSU). Technical report, Siemens, 2018.

[52] Rasheed Hussain, Heekuck Oh, and Sangjin Kim. Antisybil: standing against sybil attacks in privacy-preserved vanet. In *Connected Vehicles and Expo (ICCVE), 2012 International Conference on*, pages 108–113. IEEE, 2012.

[53] Jyoti Grover, Manoj Singh Gaur, Vijay Laxmi, and Nitesh Kumar Prajapati. A sybil attack detection approach using neighboring vehicles in vanet. In *Proceedings of the 4th international conference on Security of information and networks*, pages 151–158. ACM, 2011.

[54] Seppo Olavi Hamalainen, Haitao Tang, Achim Franz Wacker, and Osman Nuri Can Yilmaz. Method of improving coverage and optimisation in communication networks, jul 2014. US Patent 8,774,791.

[55] Jing Zhang, Xiaohu Ge, Qiang Li, Mohsen Guizani, and Yanxia Zhang. 5g millimeter-wave antenna array: design and challenges. *IEEE Wireless Communications*, 24(2):106–112, 2017.

[56] N. Bißmeyer, J. Petit, J. Njeukam, and K. M. Bayarou. Central misbehavior evaluation for VANETs based on mobility data plausibility. In *VANET'12 Proceedings of the 9th ACM International Workshop on Vehicular Inter-networking, Systems, and Applications*, pages 73–82, Lake District, UK, June 2012.

[57] Marwane Ayaida, Nadhir Messai, Sameh Najeh, and Kouamé Boris Ndjore. A macroscopic traffic model-based approach for sybil attack detection in vanets. *Ad Hoc Networks*, 90:101845, 2019.

[58] Felipe Boeira, Mikael Asplund, and Marinho P. Barcellos. Vouch: A Secure Proof-of-Location Scheme for VANETs. In *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWIM '18, page 241–248. Association for Computing Machinery, 2018.

[59] Salam Hamdan, Amjad Hudaib, and Arafat Awajan. Detecting Sybil attacks in vehicular ad hoc networks. *International Journal of Parallel, Emergent and Distributed Systems*, pages 1–11, 2019.

[60] Celestine Iwendi, Mueen Uddin, James A Ansere, Pascal Nkurunziza, Joseph Henry Anajemba, and Ali Kashif Bashir. On detection of Sybil attack in large-scale VANETs using spider-monkey technique. *IEEE Access*, 6:47258–47267, 2018.

[61] S Archana and NP Saravanan. Biologically inspired QoS aware routing protocol to optimize lifetime in sensor networks. In *2014 International Conference on Recent Trends in Information Technology*, pages 1–6. IEEE, 2014.

[62] Anika Anwar, Talal Halabi, and Mohammad Zulkernine. Cloud-based Sybil Attack Detection Scheme for Connected Vehicles. In *2019 3rd Cyber Security in Networking Conference (CSNet)*, pages 114–121. IEEE, 2019.

[63] Zhe Yang, Kuan Zhang, Lei Lei, and Kan Zheng. A novel classifier exploiting mobility behaviors for sybil detection in connected vehicle systems. *IEEE Internet of Things Journal*, 6(2):2626–2636, 2018.

[64] Chris Piro, Clay Shields, and Brian Neil Levine. Detecting the sybil attack in mobile ad hoc networks. In *Securecomm and Workshops, 2006*, pages 1–11. IEEE, 2006.

[65] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 259–268. ACM, 2004.

[66] G Anitha F Stephen Raj. Detection of Sybil attack in VANET. *Karpagam JCS*, 14(2), 2020.

[67] Sungwook Kim, Jihye Kim, Jung Hee Cheon, and Seong-ho Ju. Threshold signature schemes for elgamal variants. *Computer Standards & Interfaces*, 33(4):432–437, 2011.

[68] Mohamed Baza, Mahmoud Nabil, Mohamed Mohamed Elsalih Abdelsalam Mahmoud, Niclas Bewermeier, Kemal Fidan, Waleed Alasmary, and Mohamed Abdallah. Detecting sybil attacks using proofs of work and location in vanets. *IEEE Transactions on Dependable and Secure Computing*, 2020.

[69] Mahdiyeh Parham and Ali A Pouyan. An Effective Privacy-Aware Sybil Attack Detection Scheme for Secure Communication in Vehicular Ad Hoc Network. *Wireless Personal Communications*, pages 1–34, 2020.

[70] Mohamed Khalil and Marianne A Azer. Sybil attack prevention through identity symmetric scheme in vehicular ad-hoc networks. In *2018 Wireless Days (WD)*, pages 184–186. IEEE, 2018.

[71] Jyoti Grover, Vijay Laxmi, and Manoj Singh Gaur. Sybil attack detection in vanet using neighbouring vehicles. *International Journal of Security and Networks*, 9(4):222–233, 2014.

[72] Dongxu Jin and JooSeok Song. A traffic flow theory aided physical measurement-based sybil nodes detection mechanism in vehicular adhoc networks. In *2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS)*, pages 281–286. IEEE, 2014.

[73] Mervat Abu-Elkheir, Sherin Abdel Hamid, Hossam S Hassanein, I Brahim M Elhenawy, and Samir Elmougy. Position verification for vehicular networks via analyzing two-hop neighbors information. In *2011 IEEE 36th Conference on Local Computer Networks*, pages 805–812. IEEE, 2011.

[74] Muhammad Saad Naveed and M Hasan Islma. Detection of sybil attacks in vehicular ad hoc networks. *Universal Journal of Communications and Network*, 3(1):15–25, 2015.

[75] Muhammad Saad Naveed and M Hasan Islam. Detection of sybil attacks in vehicular ad hoc networks based on road side unit support. *Int. J. Sci. Eng. Res*, 6(2):817–827, 2015.

[76] K Murugan, R Nandhakumar, and P Varalakshmi. Localization of sybil nodes and detection of malicious node in vanets. *Int. J. Adv. Inf. Eng. Technol*, 2(7):17–21, 2015.

[77] Tong Zhou, Romit Roy Choudhury, Peng Ning, and Krishnendu Chakrabarty. P2dap—sybil attacks detection in vehicular ad hoc networks. *IEEE journal on selected areas in communications*, 29(3):582–594, 2011.

[78] Nicole El Zoghby, Véronique Cherfaoui, Bertrand Ducourthial, and Thierry Denoeux. Distributed data fusion for detecting sybil attacks in vanets. In *Belief Functions: Theory and Applications*, pages 351–358. Springer, 2012.

[79] M Thiago, Hyggo O Almeida, Angelo Perkusich, Leandro de Sales, and Marcello de Sales. A privacy-preserving authentication and sybil detection protocol for vehicular ad hoc networks. In *Consumer Electronics (ICCE), 2014 IEEE International Conference on*, pages 426–427. IEEE, 2014.

[80] Yong Hao, Jin Tang, and Yu Cheng. Cooperative sybil attack detection for position based applications in privacy preserved vanets. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–5. IEEE, 2011.

[81] Xia Feng, Chun-yan Li, De-xin Chen, and Jin Tang. A method for defensing against multi-source sybil attacks in vanet. *Peer-to-Peer Networking and Applications*, 10(2):305–314, 2017.

[82] Anu S Lal and Reena Nair. Region authority based collaborative scheme to detect sybil attacks in vanet. In *Control Communication & Computing India (ICCC), 2015 International Conference on*, pages 664–668. IEEE, 2015.

[83] Mohamed Salah Bouassida, Gilles Guette, Mohamed Shawky, and Bertrand Ducourthial. Sybil nodes detection based on received signal strength variations within vanet. *IJ Network Security*, 9(1):22–33, 2009.

[84] Yuan Yao, Bin Xiao, Gaofei Wu, Xue Liu, Zhiwen Yu, Kailong Zhang, and Xingshe Zhou. Voiceprint: A novel sybil attack detection method based on rssi for vanets. In *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on*, pages 591–602. IEEE, 2017.

[85] Mohamed Khalil and Marianne A Azer. Crypto-SAP Protocol for Sybil Attack Prevention in VANETs. In *Advances in Computer, Communication and Computational Sciences*, pages 143–152. Springer, 2020.

[86] Jan Trauernicht and Norbert Bißmeyer. Deterministic sybil attack exclusion in cooperative-intelligent transportation systems. In 17^{th} *escar Europe : embedded security in cars (Konferenzveröffentlichung)*. 2019.

[87] Kiho Lim, Tariqul Islam, Hyunbum Kim, and Jingon Joung. A Sybil Attack Detection Scheme based on ADAS Sensors for Vehicular Networks. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–5. IEEE, 2020.

[88] Eric R Verheul. Activate later certificates for v2x-combining its efficiency with privacy. *IACR Cryptology ePrint Archive*, 2016:1158, 2016.

[89] Guangjie Han, Liangtian Wan, Lei Shu, and Naixing Feng. Two novel doa estimation approaches for real-time assistant calibration systems in future vehicle industrial. *IEEE Systems Journal*, 11(3):1361–1372, 2017.

[90] John A. Stankovic. Research Challenges and Solutions for IOT/CPS. Keynote speech of Prof. John A. Stankovic at the 26th International Conference on Computer Communications and Networks (ICCCN 2017). http://icccn.org/icccn17/wp-content/uploads/2017/08/ICCCN17-Jack-Stankovic-PPT-file.pdf .

[91] Mashrur Chowdhury, Amy Apon, and Kakan Dey. *Data analytics for intelligent transportation systems*. Elsevier, 2017.

[92] Juan Contreras-Castillo, Sherali Zeadally, and Juan Antonio Guerrero-Ibañez. Internet of vehicles: architecture, protocols, and security. *IEEE internet of things Journal*, 5(5):3701–3709, 2017.

[93] Juan Contreras-Castillo, Sherali Zeadally, and Juan Antonio Guerrero Ibañez. Solving vehicular ad hoc network challenges with big data solutions. *IET Networks*, 5(4):81–84, 2016.

[94] Wenchao Xu, Haibo Zhou, Nan Cheng, Feng Lyu, Weisen Shi, Jiayin Chen, and Xuemin Shen. Internet of vehicles in big data era. *IEEE/CAA Journal of Automatica Sinica*, 5(1):19–35, 2018.

[95] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2:37 – 52, 1987. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[96] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, DTIC Document, 2003.

[97] Hammi Badis, Guillaume Doyen, and Rida Khatoun. Toward a source detection of botclouds: a pca-based approach. In *IFIP International Conference on Autonomous Infrastructure, Management and Security*, pages 105–117. Springer, 2014.

[98] Xin Xu and Xuening Wang. An adaptive network intrusion detection method based on pca and support vector machines. In *Advanced Data Mining and Applications*, volume 3584 of *Lecture Notes in Computer Science*, pages 696–703. Springer Berlin Heidelberg, 2005.

[99] Rémi Cogranne, Guillaume Doyen, Nisrine Ghadban, and Badis Hammi. Detecting botclouds at large scale: A decentralized and robust detection method for multi-tenant virtualized environments. *IEEE Transactions on Network and Service Management*, 15(1):68–82, 2017.