GOLIATH: A Decentralized Framework for Data Collection in Intelligent Transportation Systems

Davide Maffiola, Stefano Longari, Michele Carminati, Mara Tanelli, *Senior Member, IEEE,*
and Stefano Zanero, *Senior Member, IEEE.*

Abstract—Intelligent Transportation Systems (ITSs) technology has advanced during the past years, and it is now used for several applications that require vehicles to exchange real-time data, such as in traffic information management. Traditionally, road traffic information has been collected using on-site sensors. However, crowd-sourcing traffic information from onboard sensors or smartphones has become a viable alternative. State-of-the-art solutions currently follow a centralized model where only the service provider has complete access to the collected traffic data and represent a single point of failure and trust.

In this paper, we propose *GOLIATH*, a blockchain-based decentralized framework that runs on the In-Vehicle Infotainment (IVI) system to collect real-time information exchanged between the network's participants. Our approach mitigates the limitations of existing crowd-sourcing centralized solutions by guaranteeing trusted information collection and exchange, fully exploiting the intrinsic distributed nature of vehicles.

We demonstrate its feasibility in the context of vehicle positioning and traffic information management. Each vehicle participating in the decentralized network shares its position and neighbors' ones in the form of a transaction recorded on the ledger, which uses a novel consensus mechanism to validate it. We design the consensus mechanism resilient against a realistic set of adversaries that aim to tamper or disable the communication. We evaluate the proposed framework in a simulated (but realistic) environment, which considers different threats and allows showing its robustness and safety properties.

Index Terms—Automotive, Blockchain, Smart Cities, Positioning Systems, Traffic Management

I. INTRODUCTION

In the last years, Intelligent Transportation Systems (ITSs) have gone through significant changes, mainly due to the introduction of new technologies deployed on vehicles, which allow advanced applications that often require vehicles to exchange real-time data to provide high-quality services. Modern vehicles are now equipped with In-Vehicle Infotainment (IVI) systems that allow executing custom applications on top-notch mobile processors [1]. Additionally, vehicle manufacturers are investing on Vehicle to Everything (V2X) communication [2], either based on cellular network [3] or on short range communication technologies [4]. These features allow vehicles to perform complex operations and take part in a vehicular network, thus enabling the deployment of lightweight blockchain-based solutions, which guarantee trust without relying on a single, centralized point of failure.

All authors are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy. E-mail: davide.maffiola@mail.polimi.it {stefano.longari, michele.carminati, mara.tanelli, stefano.zanero}@polimi.it

The most common field of application of ITSs is traffic information management, which requires the processing of a high volume of real-time traffic information. Currently, traffic information sources are local authorities or agencies that use costly on-site sensors such as Road Side Units (RSUs), or services based on Floating Car Data (FCD) and crowdsourcing like *Waze*, *TomTom HD Traffic*, and *Google Maps*. However, both technologies are based on centralized systems, where the service providers have complete control of the collected data and acts as a trusted element. Although these methods can provide crowdsourced traffic information, they suffer from high deployment and maintenance costs, and their effectiveness is limited to a specific area. Also, precise traffic information is challenging to measure and estimate, and the available sources are often partial or not easily accessible. In fact, in last years we assisted to the growth of blockchain-based solutions applied to various domains of the automotive ecosystem [5]–[21]. However, these solutions are usually only semi-decentralized since they rely on base stations (e.g., RSUs) and are not evaluated against advanced attacks in the automotive environment.

In this paper, we aim to solve the challenges mentioned above by proposing *GOLIATH*, a blockchain-based framework that relies on the resources offered by IVI systems to provide a decentralized alternative to centralized services commonly used to collect real-time information exchanged between network's participants. We demonstrate its feasibility in the context of vehicle positioning and traffic information management by processing real-time traffic information. In the proposed framework, each participating vehicle shares its position and its neighbors' ones, obtained through an on-board Global Navigation Satellite System (GNSS) module, in the form of blockchain transactions. To detect nearby vehicles, the participants periodically use short-range communication to broadcast a probe message. When the participants receive a reply to their probe message, they generate, sign, and broadcast, for each received answer, a transaction containing the information about the occurred interaction (i.e., the probing entity's position and the vehicles' identifiers). The transactions are then collected in a block, which goes through a novel consensus mechanism for validation. The consensus mechanism verifies the integrity of the received information and evaluates the participants' reliability and trustworthiness. By doing so, *GOLIATH* is resilient against adversaries aiming to tamper or disable the communication, which instead have been proven effective against current centralized solutions [22]. Once validated, all participants are updated with the new

block. The consensus mechanism is entirely managed by the participant vehicles with supporting RSUs and does require any entity to have special privileges or permissions after the break-in phase. This allows the framework to be fully decentralized, implying that a trusted intermediary – i.e., the service provider – is no longer required.

We validate *GOLIATH* by implementing a network simulator for realistic traffic generation. We demonstrate the resilience of the framework in realistic scenarios while also showing its capability of maintaining high-quality traffic information even in the presence of malicious entities.

Our contributions are the following:

- *GOLIATH*, a lightweight, decentralized, blockchain-based framework for real-time data collection in ITSs that mitigates the limitation of centralized systems and fully exploits the intrinsic distributed nature of vehicles. We also demonstrate its feasibility in the traffic information management domain.
- A novel consensus mechanism designed to be resilient, differently from existing centralized solutions [22], against a realistic set of adversaries that aim at tampering or disabling the communication. The consensus mechanism is entirely managed by the participant vehicles, removing the need for a centralized service provider.
- A simulation tool (available at https://github.com/necst/GOLIATH) which is, to the best of our knowledge, a first attempt to evaluate decentralized traffic information management systems through the simulation of realistic attack scenarios. This tool will help future researchers in the comparison with the state of the art and in the evaluation of the robustness of their approaches.

The paper is structured as follows: In Section IV we describe the problem we aim to tackle and our research goal. In Section II we offer an overview of blockchain technologies, while in Section III, we discuss the current status of positioning systems and related works that study the blockchain technologies in the automotive domain. Then, in Section V we provide an analysis of the threat model, fundamental to understand the choices we made in Section VI, where we present *GOLIATH*. In Section VII we show the experimental evaluation of our framework. Finally, in Section VIII we discuss the conclusions of our work, discussing the main limitations and outlining some future works.

II. BACKGROUND ON BLOCKCHAIN TECHNOLOGY

Blockchain is a vast research topic, for which we refer the reader to [23], [24]. In this section, we provide the basic concepts needed to understand our contribution.

A blockchain is a data structure employed in decentralized and distributed systems to obtain immutable and verifiable data in the form of blocks. This solves the issues deriving from the lack of a centralized trusted intermediary intrinsic of a Peer-to-Peer (P2P) network. A blockchain-based system is composed of three core components: **nodes**, which are the participants of the network, **transactions**, which represents the interactions between nodes (e.g., monetary transactions in blockchain-based cryptocurrencies), and **blocks**, which are

collections of valid transactions over a period of time. On top of these three components, blockchain-based systems require the definition of two processes to ensure authenticity and verification of data: the first is a **signature scheme** that grants the integrity and authenticity of transactions and blocks, and the second is a **consensus mechanism**, which has the goal of validating data. The consensus mechanism is composed of a **consensus algorithm** and a **validation algorithm**. The former is used to achieve a distributed consensus, and the latter checks the validity of the content of a block. One of the most known consensus algorithms is the Proof-of-Work (PoW), implemented initially by Bitcoin [25], which chooses the next valid block by rewarding nodes based on expended computation and time. A second class of consensus algorithms is the Proof-of-Stake (PoS) [26], where instead of computational power, an abstract resource imaginable as a currency is used by the nodes as a stake. The higher the stake a node offers, the higher is the chance of that node generating the next block. In this work, we opt for a PoS-based system in light of the strict computational requirements of the automotive field that would make PoW unfeasible.

From a high-level perspective, nodes (i.e., participants) of a blockchain generate transactions that express the interactions between each other and broadcast them. Harvesters then collect these transactions into candidate blocks and compete for the right of adding their candidate block to the blockchain through the consensus algorithm designed for that specific blockchain system. The block of the winning harvester is then added to the blockchain by attaching to it the hash of the previous block, officially chaining the new block to the blockchain. The robustness of a blockchain-based system depends on the number of legitimate participants in the network since they directly influence the consensus mechanism. Hence, it is of paramount importance for a blockchain-based system to reach critical mass (i.e., a high number of participants) in a short period and that the consensus mechanism rewards legitimate behaviors only.

III. RELATED WORKS

In the following section, we present related blockchain-based works that pave the way for the validity of our approach and, then, we illustrate decentralized solutions applied to the automotive field.

A. Related Blockchain Methodologies

In the last ten years, the need for decentralized solutions to share information has been answered most of the time by the blockchain technology. Initially in 2008 by Nakamoto with Bitcoin [25], followed by many others. We briefly discuss those that apply techniques related to or similar to ours. ByzCoin [27] proposes to improve the transaction acceptance delay of Bitcoin through the use of a consensus mechanism based on Practical Byzantine Fault Tolerance (PBFT) [28], where view-based leader election is decoupled from the validation algorithm. The Proof-of-Believability (PoB) consensus algorithm used in IOST [29], which is a modification of a PoS algorithm, divides the participants into a believable league and

a normal league, similarly to the method used in the proposed framework. The believable league, though, is only used to validate the transactions optimistically, and the normal league still has to run the modified version of PBFT used in ByzCoin. In particular, the blockchain technology has been applied in nearly all domains of the IoT [6], from the healthcare [30]–[32] and embedded devices [33], [34], to fog computing [35] and software-defined networking [36].

B. Related Blockchain-based works in the automotive field

Blockchain-based solutions have also been applied to the automotive ecosystem. In particular, the most common applications are related to trust-based dynamic data collection and exchange [5]–[9], secure storage of static data [10]–[14], key management for authentication and verification [15]–[17], and energy trading in vehicular networks [18]–[21].

Regarding the trust-based dynamic data collection and exchange domain, to which our work belongs, the blockchain is used for exchanging information in dynamic vehicular networks (i.e., traffic management systems). The security (e.g., integrity and availability) of the exchanged information is guaranteed through the concept of trust among the entities involved in the communication, where only entities that gain *trust* through legitimate interactions and can participate in the blockchain. Yahiatene et al. [5], [6] propose a framework based on Software Defined Vehicular Network (SDVN) and blockchain that builds trust through the feedback of satisfaction of other nodes regarding the received information. The process is handled in a semi-centralized manner by RSUs, and the miners' election is done dynamically through a novel distributed miners-connected dominating set algorithm (DM-CDS). Yang et al. [7], [8] propose a trust model based on blockchain where the information received by the RSU, which acts as a centralized controller, is given a credibility rating that builds the trust value of each of the nodes. Li et al. [9] propose Creditcoin, a framework to encourage privacy-preserving communication of alerts between vehicles with a cryptocurrency-like solution, where RSUs are used as a trusted element, and the alert generator is rewarded by receiving a tailored currency in a PoS-like fashion.

Regarding the secure storage of static data domain, multiple solutions have been designed to securely and publicly share long-term data from different sources, such as vehicle and parts chain of ownership, insurance data, and other VANET-related data (e.g., driving habits, on-board sensors data), through the use of various blockchain technologies. Distefano et al. [11], Jiang et al. [10], Sharma et al. [13], and Javaid et al. [12] study the feasibility and the requirements needed to build blockchain-based frameworks to accommodate the needs of the automotive domain, while Li et al. [14] tackle the specific issue of ad dissemination in vehicular networks.

The works that fall in the key management for authentication and verification domain [15]–[17], as the name suggests, tackle the issue of authentication in vehicular networks, which have tailored requirements such as privacy-preserving pseudonyms, fast key transfer time, and low computing and communication overheads.

Finally, multiple works focus on the issue of energy trading in vehicular networks [18]–[21], handling the exchange of data required to track the involved entities and transactions, again focusing on the preservation of the privacy of the vehicle owners while considering the requirements of vehicular networks.

Discussion on related works. In this paper, we propose a blockchain-based framework for real-time data collection in ITSs that exploits a novel consensus mechanism that falls into the trust-based dynamic data collection and exchange domain. It is evident how the scope and structure of solutions that fall in the other aforementioned categories, although being based on blockchain technologies, significantly differ from the approach of our paper, making a direct comparison unfeasible. Regarding the trust-based dynamic data collection and exchange domain, all the presented related works manage participants (e.g., miners, harvesters) differently. In fact, most of the time, they are based on RSUs or base station to which the *mining* process is delegated. Consequently, related works are based on a centralized or at most semi-decentralized architecture, which significantly differs from our approach, which is fully decentralized (the RSUs are needed only in the initialization phase). Indeed, the comparison of output results can not take place, besides being of little significance. Therefore, in line with the presented related works, we resort to demonstrating the robustness of the blockchain-based solution by studying the resistance against known attacks (e.g., Sybil attacks). In addition, in the presented work, we further evaluate the feasibility of our approach in the automotive scenario by exploiting a network simulator for realistic traffic generation, demonstrating the resilience of the framework in realistic scenarios while also showing its capability of maintaining high-quality traffic in-formation even in the presence of malicious entities. This contribution, in particular, is missing or only briefly discussed in all the above-presented works.

IV. PROBLEM STATEMENT AND RESEARCH QUESTION

Exchange of data between vehicles in V2X environments has currently been designed mainly as a centralized, proprietary communication that uses technologies such as 4g, 5g, and Dedicated Short Range Communications (DSRC) to transfer data. Regarding the context of vehicle positioning and traffic information, local authorities and agencies in large cities, e.g., Transport for London [37], often provide real-time traffic information on an open-access basis. This information is primarily collected through on-site sensors like CCTV cameras, Wi-Fi data, and induction loops, possibly paired with artificial intelligence techniques [38]. Several companies, like TomTom [39], Google [40], and Waze [41] have adopted crowdsourcing and FCD [42] to remove the need of on-site sensors. This approach relies on GNSS and Global System for Mobile Communications (GSM) to collect the real-time positions from the devices of active users.

Although these companies allow users to obtain aggregated traffic information, the raw data are not publicly accessible, and the offered services adopt a loose security model. These centralized systems have been proven vulnerable to several successful attacks that tamper with the collected traffic information, the most recent being from Simon Weckert [22].

Research Question. Based upon the same technologies premises (i.e., crowdsourced information collection and global positioning), the research presented in this paper aims to explore the feasibility of a fully decentralized framework for vehicle data collection based on a blockchain data structure, whose properties directly tackle the weaknesses of current centralized solutions while preserving their functioning. To achieve the research goal, we have to face two challenges. The first challenge regards ensuring the system's resilience (i.e., reliable and trusted information collected) against realistic attacks. The second challenge regards the compatibility of the proposed framework with existing automotive technology. In fact, computational power is a known constraint in the automotive domain. Even if *GOLIATH* can be applied to any exchanged data – with proper modification – in the remainder of this work, we apply our approach to the traffic information management domain since it is one of the most common and best fits the problem under analysis.

V. THREAT MODEL

Threat modeling [43] is essential to derive the non-functional security requirements in the design of a system.

Threat Agents. The main threat agents are individuals (or small groups) who aim at profiting from disrupting the service (e.g., ransomware attack) or wish to hijack traffic for personal gain or to affect society at large. Attackers may also wish to forge information for anti-forensic purposes (i.e., claiming they are or they have been in a different position). Another potential threat comes from competitors who may gain an advantage from a loss of reputation of the targeted service. Therefore, the most significant threats are a disruption or degradation of service or targeted manipulation of the results.

Attacker Goals. To disrupt the service, an attacker could aim to lower the overall quality of the collected information (e.g., broadcasting a high number of crafted artifacts) or deploy a Denial of Service (DoS) attack by exhausting the resources – computational or network related – of the participants (e.g., flooding the system with forged data). To manipulate the system results or to forge fake records, attackers could aim to make the system accept as valid maliciously crafted data (e.g., faking the position of vehicles). It should be noted that the two goals affect each other since the robustness of blockchain-based solutions depends on the number of participants. Disrupting the service lowers the (perceived) reliability of the system, which in turn lowers the interest in participating of legitimate nodes. If a minor number of nodes participates in the consensus mechanism, the capability of attackers to manipulate the system increases.

Attack Modeling. Attackers can perform their attacks on the blockchain core components: at a transaction level, block, or consensus level. At a transaction level, the attacker can forge or spoof fake transactions. To name a few, they can selectively reply only to DSRC probes sent by some participants, or broadcast transactions with incorrect information, or send spam transactions. At a block level, attackers attempt to tamper with the blockchain validation algorithm by altering the validity tags of transactions, the reputation scores, or the
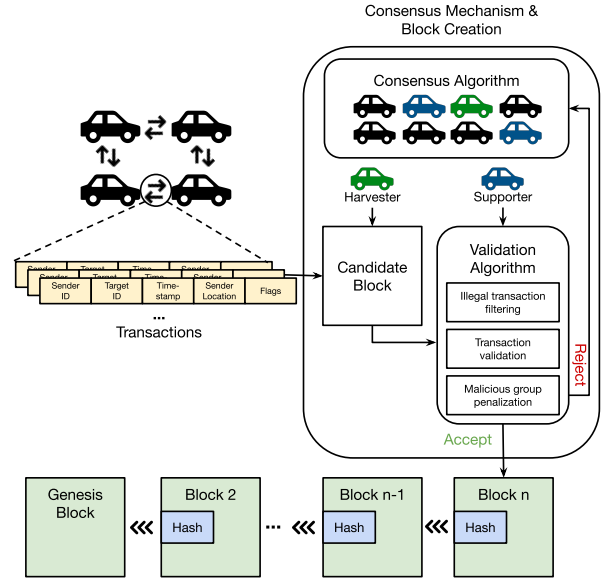

Fig. 1: Overview of *GOLIATH* Approach.

activity flags. Also, attackers can try to gain the majority during the execution of the consensus algorithm and reject a legit block or force a forged block to be accepted. These block-level attacks fall into the category of Sybil attacks [44].

VI. *GOLIATH* APPROACH

In Figure 1 we provide an overview of *GOLIATH*. Each participating vehicle shares (i.e., broadcasting a message) its position and its neighbors' ones in the form of blockchain transactions. In the meantime, following the consensus mechanism, the participants elect a harvester, which generates a valid candidate block by aggregating the broadcasted transactions, and a set of supporters, which run the validation algorithm. The candidate block, generated by the harvester, is evaluated by the supporters following the validation algorithm and, if valid, is broadcasted and added to the blockchain.

The consensus mechanism determines how the blockchain grows by validating blocks. It performs primarily two operations. First, it executes a *consensus algorithm* to ensure that the choice between accepting or rejecting a new block is indisputable by electing harvesters and supporters. This operation permits decentralization and the removal of trusted intermediaries, allowing the consensus mechanism to work in a trust-less environment.

The validation algorithm ensures the integrity of the received information and evaluates the participants' reputations. We define the *reputation* as a measure of the reliability and trustworthiness of a participant that depends on the correctness of the history of its past transactions, and we use it as the abstract resource (i.e., stake) in our PoS. In fact, participants may broadcast incorrect or malicious transactions. Therefore, validation is required to decide whether a transaction is valid or not, and the reputation of the transmitting node is updated accordingly. Unlike the consensus algorithm, the validation is domain-specific.

A. Blockchain Data Structure

The blockchain data structure stores traffic information in the form of transactions and blocks. It guarantees the properties of immutability, transparency, decentralization, persistence, and pseudo-anonymity [45].

Transaction. The primary entity in a blockchain data structure is the *transaction*, representing the interaction between two or more participants. In our context, transactions are generated by the participants through short-range communication. Participants – also called nodes – periodically broadcast a probe message that is captured by nearby vehicles (i.e., vehicles in communication range), which respond with a reply message. Regarding the communication range, it can be defined by both distance-bounding protocols [46] or by considering the maximum detection distance of short-range communication devices. Since, according to the US Federal Communications Commission (FCC) classification, a low-power DSRC transmitter has a maximum range of about 15 meters [47], from this point on, we will consider as "short-range" a distance in the order of a few tens of meters. Then, the probing vehicles generate a *transaction* for each reply received, which contains their identifier, the identifier of the detected participant, their position obtained through GNSS (e.g., GPS), and a timestamp. The sender digitally signs its transactions individually, making them immutable, and broadcasts them to all the other participants using a long-range communication mean (e.g., cellular network).

Block. The second element that composes the blockchain is the *block*, which contains all the information that defines the current state of the blockchain and references the previous one through its hash value, forming a structure similar to a linked list. Each block contains up to a maximum number of transactions, which is defined by a "block-size". Blocks are added to the blockchain after being validated by the consensus mechanism (see Section VI-B), and each participant is rewarded by increasing their reputation.

A new block is inserted after a fixed amount of time from the previous insertion, which we refer to as "block-time" (i.e., the interval between two executions of the consensus mechanism). If more transactions than the block size are generated in the block-time, we apply a "greedy" selection strategy. When a new transaction is received, our strategy randomly selects a transaction from the participant with the highest number until it finds one that is older and removes it. Blocks also include a tag that indicates each transaction validity status, the list of identifiers of the participants, an integer score representing their reputation, and a flag that indicates if the participant has been active during the "block-time" (A node is considered active if at least one of its transactions is evaluated as valid). The metadata stored in the block includes a timestamp indicating when the block was created, the digital signature of the node that harvested the block, the block height, which is the index of the block from the genesis block (i.e., the first validated block), and the attempt number, that is derived from the elapsed time since the last block as

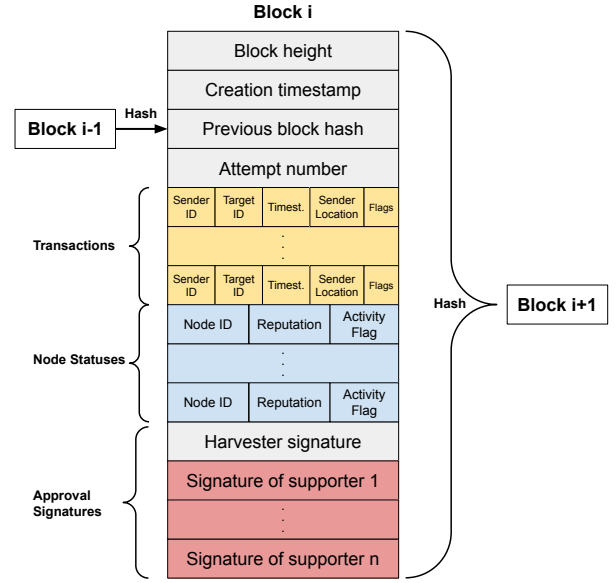$$a = \lfloor \frac{\Delta time}{block - time} \rfloor$$

Fig. 2: Structure of the blockchain and of the blocks used in the proposed framework.

The attempt number is necessary to distinguish different attempts when the consensus mechanism fails, causing a block to be rejected. Figure 2 graphically summarizes the structure of the blocks and the transactions.

Genesis Block. A crucial aspect to consider is the generation of the first block in the blockchain, which is called the *genesis block*. It has to satisfy the assumptions of the validation phase (see Section VI-C) and allows *GOLIATH* to start correctly. Consequently, the genesis block contains a set of randomly generated transactions between initially trusted participants (e.g., RSUs). In other words, the genesis block must be trusted in the break-in phase. After this phase, these trusted participants are no longer required and are treated as normal ones.

Resources Optimization. The amount of memory required to store the whole blockchain grows constantly. This represents a critical issue for embedded systems, such as In-Vehicle Infotainment (IVI) systems, where the amount of memory is limited. It is possible to introduce the concept of *state blocks* [29], [48], leaving the task of memorizing the full blockchain only to the participants with more resources, like RSUs. When there are enough regular blocks, the consensus mechanism produces a new state block that summarizes and aggregates the oldest regular blocks, allowing participants with limited resources to store only state blocks and the most recent regular blocks. Moreover, new nodes that want to join the blockchain have to download and verify less data, granting a faster join.

Key distribution and Pseudonymity. The node identifiers and their respective asymmetric keys are bound to one vehicle. Their distribution may be performed by car dealers or by the local Department of Motor Vehicles. Therefore, strict anonymity cannot be enforced. However, it is possible to provide *pseudonymity* by creating identifiers uncorrelated with real entities [49]. The definition of a method to distribute

identities and keys is beyond the scope of this paper.

B. The Consensus Algorithm

For the consensus algorithm design, we evaluate established ones [25]–[27], [29], [50]–[52] and customized them given the requirements of the automotive context. Consensus algorithms can be systematized into election-based or voting-based.

Election-based algorithms rely on a resource that is difficult or expensive to obtain, like computational power or invested money. A leader, called the block harvester or the block miner, is then elected considering this resource. The harvester performs the validation of the block, digitally signs it, and broadcasts the resulting block to all the other participants, which only have to check the eligibility of the harvester to accept the new block. The main alternatives for this approach are PoW, like in Bitcoin [25], and PoS and its variants, like in Peercoin [26]. PoW can not be executed on embedded systems such as IVI systems due to the computational requirements [53]. On the other hand, PoS is robust only if the resource that is at stake does not vary too quickly [54], which is not easily achievable in a network composed of vehicles where participants can join and leave frequently.

Voting-based algorithms are variations of the Practical Byzantine Fault Tolerance (PBFT) [28] method. All the participants take part in the voting process and decide whether the candidate block proposed by the harvester is acceptable or not. Because of this feature, voting-based algorithms perform better in small networks, where they typically have higher throughput than election-based algorithms. Using this approach, all the participants execute the validation phase at least once per block. However, this overhead is undesirable because the framework would be much more resource-intensive. Moreover, the amount of exchanged messages required to achieve consensus is proportional to the square of the total number of participants, introducing significant overhead.

In blockchains, the election can be deterministic (e.g., ByzCoin [55]) or probabilistic (e.g., NEM [51]). A deterministic approach uses a synchronized view-number to perform the election. Instead, a probabilistic approach relies on a time-dependent condition evaluated locally by each participant to determine if it is eligible to harvest the next block. This means that, in this case, explicit synchronization among the participants is not required. However, several participants may try to harvest the next block simultaneously, causing the blockchain to fork, which does not occur in deterministic approaches. Eventually, a fork policy solves the issue, but the participants have already wasted their limited resources in discarded operations.

Proposed Consensus Algorithm. In *GOLIATH*, we propose a consensus algorithm that combines election- and voting-based approaches adapting their functioning to the domain under analysis and finding a trade-off between resource efficiency and performance. Instead of involving all the participants in the voting process, only a subset of them, which we defined as *supporters*, perform the PBFT algorithm on the new block. The harvester produces a new block; then, it broadcasts the new block to the rest of the participants, which stores the block and waits for its approval by the supporters. The supporters vote on whether to accept or reject the proposed block following the PBFT method and broadcast their decisions, through approval signatures, to the non-supporting participants. Once it receives enough approval signatures, each participant adds the corresponding block with the received approval signatures to its local copy of the blockchain. Unlike voting-based methods, the proposed consensus algorithm requires a robust method to determine when and how to elect the harvester and the supporters. For this reason, we introduce an election step inspired by election-based approaches. Since probabilistic approaches may waste resources due to the simultaneous block harvesting and the consequential forking of the blockchain, we adopt a deterministic approach based on the elapsed time since the creation of the previous block. To avoid an easy to predict view-number, we derive its value from both the hash of the previous block – on which there is already consensus – and the elapsed time – which can be kept synchronized with various solutions, e.g., Global Positioning System (GPS). The harvester and the supporters are elected based on the view-number and their reputation. In a lottery-like fashion, each candidate receives several lottery tickets proportional to its reputation. Then, winning tickets (one for each supporter and harvester to be elected) are extracted based on the view-number. More formally, for each election, the view-number is concatenated with an increasing integer, and the resulting value is hashed using SHA-256 (i.e., winning tickets). The hash function is used as a uniform pseudo-random number generator. The algorithm then assigns to each node a partition of the image of the hash function used, proportionally to their reputation (i.e., number of lottery tickets per candidate). Finally, the algorithm elects the participant that owns the partition in which the hash is included. Since the output of the SHA-256 function is comparable to a uniform distribution, the probability of electing a participant is proportional to its reputation score. Regarding the number of elected participants, following the PBFT method, we select a total of $n = 3f + 2$ candidates either as harvester or supporters, where f represents the fault tolerance parameter, which corresponds to the minimum number of approval signatures required to add a new block. By doing so, we reduce the number of average validation phase executions per block to n.

Proposed Algorithm Evaluation. We demonstrate the robustness of the proposed consensus algorithm by analyzing the probability of a successful malicious block insertion in the case of a Sybil attack [44]. To achieve the malicious goal, an attacker should be elected as harvester and control a number of supporters sufficient to win the voting procedure. Considering $3f + 1$[1] supporters, a malicious block insertion is successful if the harvester node is malicious and the attackers own over 33% of the supporter nodes. To perform the analysis, we model the problem by assigning a probability distribution to each involved variable. We define as H the probability that an attacker has been elected as a block harvester, which is a Bernoulli random variable with success probability p, where p

---

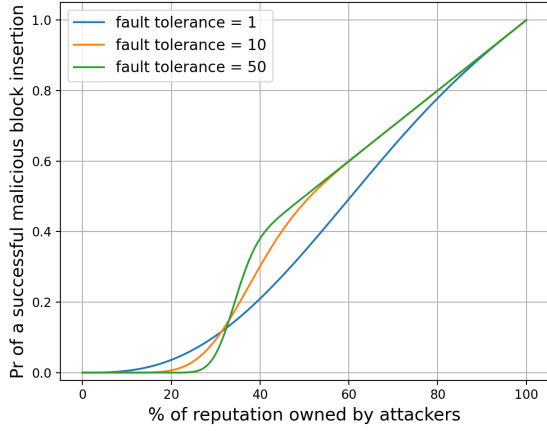[1] f is the fault tolerance parameter of Practical Byzantine Fault Tolerance (PBFT)

Fig. 3: Probability of a successful malicious block insertion in relation with the amount of reputation owned by attackers.

is the ratio of the attackers' reputation with respect to the total reputation. We define as M the number of nodes controlled by attackers elected as supporters and will participate in the voting process. Since $n = 3f + 2$ is the total number of elected nodes between harvester and supporters and f corresponds to the minimum number of approval signatures required to add a new block, we model M as a binomial random variable with $3f + 1$ trials, each having a success probability p, which is the same as in H. Equation 1 shows how the formulas for the probability of success of attackers (i.e., malicious block insertion) are derived.

$$
\begin{aligned}
H &\sim Bernoulli(p) \\
M &\sim Binomial(3f + 1, p) \\
Pr(&\text{``A malicious block is inserted''}) = \\
&= Pr(H = 1 \land M \geq f + 1) = \\
&= Pr(H = 1)(1 - Pr(M \leq f)) = \\
&= p \cdot \left(1 - \sum_{k=0}^{f} \binom{3f + 1}{k} p^k (1 - p)^{3f+1-k}\right)
\end{aligned}
\tag{1}
$$

To evaluate the robustness of the proposed approach, we, therefore, study the probability of a successful malicious block insertion by varying p (i.e., as the amount of reputation owned by attackers grows) and f, as shown in Figure 3. It is evident how by keeping p in the interval $[0; 33\%]$ (i.e., the intersection between the f curves), an increase in the fault tolerance parameter f maintains the success rate of attackers low, if not close to zero.

C. The Validation Algorithm

In light of the strict computational requirements of IVI systems, we designed three heuristics that achieve the same result – i.e., the validation of the block in the ledger performed by elected participants – of algorithms implemented in contexts where the computation power is not limited (e.g., Bitcoin [53]). In particular, in this phase, we validate a block by removing invalid transactions and updating participants' reputations following a *reputation update strategy*.

1) Illegal transaction filtering: The first heuristic filters out illegal transactions and apply a penalty (i.e., lower participants' reputations) to their owners. First, it filters out transactions that have the same target and sender IDs because the sender could otherwise exploit the validation algorithm to increase its reputation score by itself. Then, it removes transactions that are not "acknowledged" by participants. In other words, for each transaction, there must be another transaction with sender and receiver IDs swapped, i.e., from the detected vehicle to the original sender. By doing so, we prevent a single attacker (i.e., not organized in a group) from masking itself as another vehicle.

2) Transaction validation: The second heuristic validates each transaction individually. It considers the transactions contained in the previous N blocks and checks if the information contained (i.e., position and timing) is compliant with the laws of physics. By assuming a linear motion, it is possible to find an upper bound to the travel distance of a vehicle for a given interval of time Δt and a maximum velocity v_{max}. Therefore, we reject a transaction depending on participants' reputations and whether the travel distance exceeds these bounds. Also, we apply a penalty to transactions' owners. By comparing the identifiers contained in old and new transactions, we define four bounds, depending on whether the participant under analysis is sender or detected node in the transactions: the *sender-sender bound*, the *sender-detected bound*, the *detected-sender bound*, and *detected-detected bound*. For the sender-sender bound, we compute the euclidean distance between the sender positions in different transactions since the values of its positions are directly indicated in the transaction. This is not true for all other bounds, where the node under analysis is a detected one. In these cases, the position of the detected node must be approximated based on the position of its sender, the geometry of the problem, and the maximum range of detection $range_{max}$. For example, the detected-sender bound is:

$$
distance(sender_{old}, sender_{new}) \leq v_{max}\Delta t + range_{max}
$$

Then, we assign a score for each new transaction under analysis: when the travel distance is not coherent with the bounds, we lower the score assigned to the transaction of a value equal to the reputation of the sender of the older transaction. Otherwise, the score is increased by the same amount. Finally, we normalize the score with respect to the number of bound checks made and compare it with a threshold. The new transaction is rejected if the normalized score is lower than the threshold. The evaluation of the transaction is not executed if the number of matches is equal to zero.

3) Malicious groups penalization: The third heuristic penalizes the misbehavior of malicious groups, i.e., groups of participants that support each other to boost their reputations while faking localization by forging transactions. This heuristic is necessary since this behavior would pass unnoticed through the previous heuristics. We adopt a strategy to penalize misbehaving participants over time by analyzing each participant independently and detecting conflicts. There is a conflict between two participants whenever they claim to be in two close positions, but they do not detect each other (i.e., there is no such transaction within the block).

For each conflicting pair of participants, the one with the lowest reputation score receives a penalty. When a participant loses more than a defined amount of conflicts, all of its transactions are blacklisted. This strategy is based on the assumption that misbehaving participants have, on average, less reputation than legitimate ones. Finally, it is essential to highlight that this heuristic does not claim to provide an exact representation of the traffic stream but only approximates it by analyzing the transactions exchanged between participants. As previously explained, each block contains a limited number of transactions, and the remaining are dropped. Consequently, this heuristic may have a high number of false positives. However, experiments have shown that it does not deteriorate the overall performance since it is unlikely that a correctly behaving participant generates many conflicts systematically.

Reputation update strategy. We update the reputation of the ledger participants based on the outcome of the previous heuristics and their activity in the blockchain. To do so, we first define as active a participant with at least one transaction in the block. The reputation of active participants is updated by summing to its original value the difference between their total reward and total penalty. The total penalty is computed by multiplying a penalty base value with the number of rejected transactions. Instead, the total reward is computed by multiplying a reward base value with the number of accepted transactions and a scaling factor that considers the number of frequent neighbors (i.e., participants that often appear together in the transactions). The scaling factor mitigates the capability of groups of malicious participants to boost each others' reputation: the more a group of participants interacts only with each other, the lower the scaling factor and, hence, the reward is. The scaling factor is computed as:

$$S_f = 1 - w \frac{|new_i \cap old_i|}{|old_i|}$$

where w is the neighbor variance weight, new_i is the set of participants detected by participant i in the new block, and old_i is the participants detected in the previous M blocks. A low neighbor variance is more common in malicious groups, as they are only a few and require to interact frequently with each other to gain support and carry out the attack. Regarding the reward and penalty base value, we select a fixed value over a dynamic one (i.e., a value that changes depending on the node history) since it performed better in our experiments. Dynamic value often zeroes the reputation of legit participants, making them unable to contribute to the validation of transactions. We also defined a maximum value for the reputation score so that participants that never go offline, such as RSUs, cannot gain infinite reputation and, hence, monopolize the execution of the consensus algorithm. Inactive participants, instead, periodically lose their reputation to give more weight to recent transactions over older ones. This guarantees that malicious participants have a lower reputation score than legit ones and that it is more difficult for them to increase it.

TABLE I: Values of the parameters used in the experiments obtained through the performance and robustness analysis.

Component	Parameter	Value
Scenario	No. of vehicles	1,500
	No. of RSUs	40
	Max speed	130 km/h
Blockchain	Blocksize	50,000
	Blocktime	60 s
Transaction validation	No. of previous blocks	3
	Threshold	0.1
Misbehavior penalization	Variance weight	0.67
	Tolerable lost conflicts	2
Consensus algorithm	Fault tolerance	6
Reward system	Max reputation	4,096
	Initial reputation	64
	Base reward	256
	Base penalty	512

VII. EXPERIMENTAL RESULTS

We evaluate *GOLIATH* in a simulated (but realistic) environment. First, we analyze the performance and the robustness of the proposed framework through parameter studies. Second, we test the resistance of the framework against attacks derived from the threat model.

A. Simulation Setup

The experiments conducted on *GOLIATH* are carried out through simulations designed to be as realistic as possible and provide reliable results. For this purpose, we developed a modular simulator based on the Veins vehicular network simulation framework [56]. This choice allowed us to use the OMNeT++ [57] simulation library to simulate the network model and SUMO [58], a microscopic traffic simulator. Also, we adopt the Monaco SUMO Traffic (MoST) scenario [59] which is based on the Principality of Monaco and simulates 46,000 trips in 10 hours. This setup guarantees a realistic road layout, vehicle interactions, and traffic dynamics. Our experiments consider 1,500 vehicles interacting for one hour to keep the computational and storage requirements manageable.

B. Experiment 1: Performance and Robustness Analysis

The framework needs to have a stable and robust configuration that allows it to function correctly. For this reason, in the first experiment, we analyze the impact of the *GOLIATH* parameters to empirically estimate their optimal values and their sensitivity in a context where no attack occurs. For each of the parameters under analysis, we analyze the average reputation of participants (i.e., the stake of our PoS) as a metric of the goodness of our system. In particular, we run different simulations varying the value of the parameters[2]. In Table I, we show the final values obtained for each parameter that we then use for subsequent experiments.

Number of previous blocks for transaction validation. First, we study the number of preceding blocks (and their related transactions) used to validate new transactions and,

---

[2]In the following experiments, we highlight only the most relevant values

(a) No. of previous blocks in transaction validation.

(b) Threshold in transaction validation.

(c) Initial reputation.

(d) Base penalty.

(e) Tolerable lost conflicts.

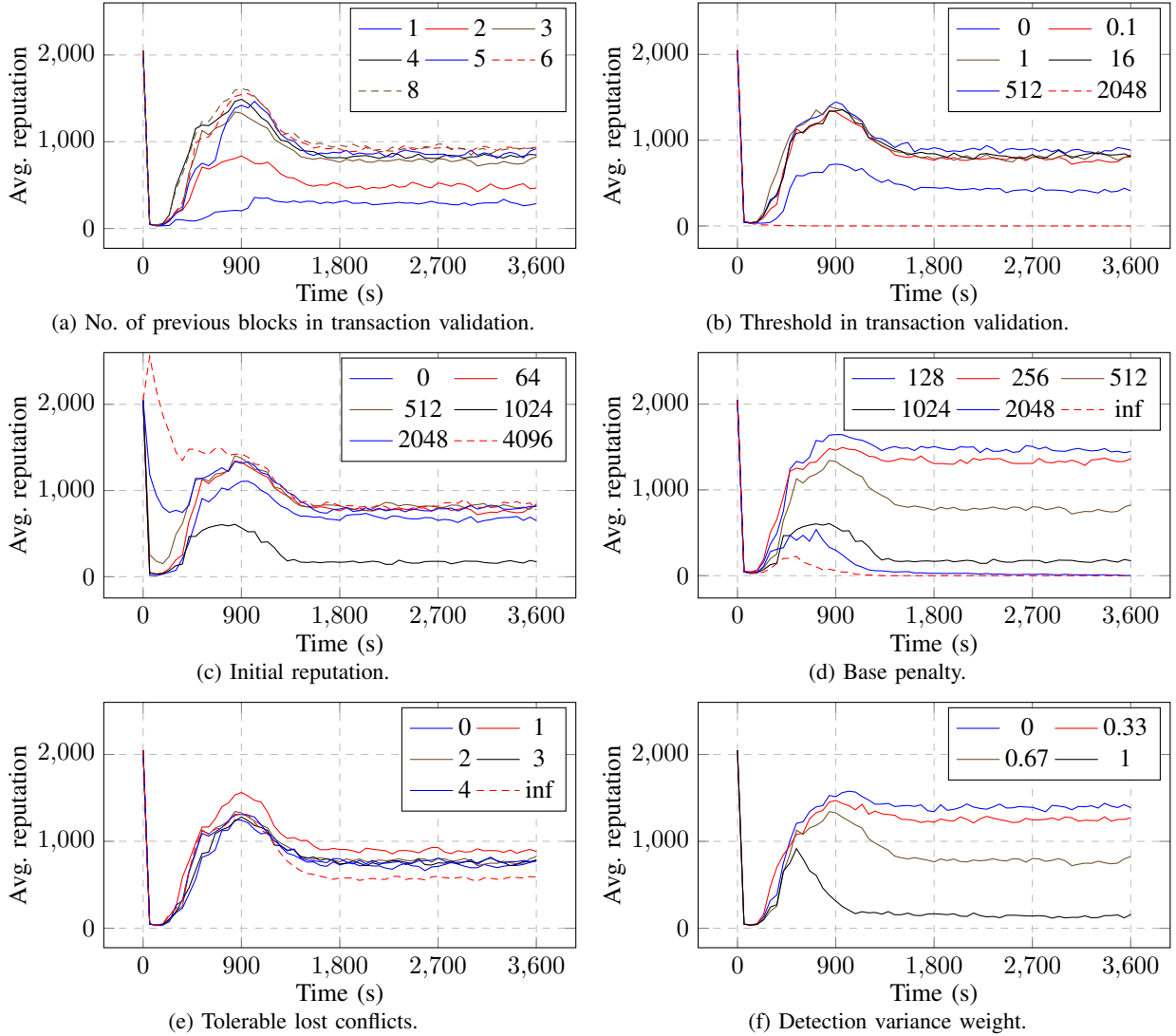(f) Detection variance weight.

Fig. 4: Experiment 1: Analysis of the average reputation of active participants, varying one parameter at a time.

therefore, recognize the attacks. The higher is the number of blocks used, the more the transaction validation algorithm complexity and the computation time increase. On the other hand, the lower is the number of blocks used, the higher is the chance of rejecting legitimate transactions (due to the lack of a robust transaction history), lowering the reputation of legitimate nodes. Figure 4a shows the average reputation of active participants under different values of this parameter. When the framework reaches the steady-state, all but two configurations produce similar outcomes, suggesting that using more than three previous blocks does not improve the transaction validation enough to justify the increased complexity.

Threshold in transaction validation. The threshold used in the transaction validation (see Section VI-C2) is necessary to define the minimum required support to accept a new transaction. A threshold too high would compromise the system's functioning since it would lead to a high number of legitimate rejected transactions (i.e., the score assigned to new transactions would rarely satisfy the threshold) and, therefore, an increase in the penalization of legitimate participants. Figure 4b shows a comparison of the average reputation as the

value of the threshold varies. Low threshold values (i.e., 0, 0.1, 1, 16) produce similar outcomes (i.e., the average reputation shows an almost identical trend and a low variance). However, as the thresholds increase in value, the average reputation drops, as expected. In particular, the highest tested threshold (i.e., 2048) completely prevents the correct functioning of the framework, forcing the average reputation to drop to zero abruptly. Considering a scenario where no attacks occur, a low value of the threshold guarantees the highest performance. Consequently, we choose a threshold of 0.1. The goodness of the threshold is also confirmed by results obtained in experiment 2.

Initial reputation. The initial reputation given to participants when they join the system for the first time should not significantly affect the proposed framework's long-term performance. A high initial reputation would lead to a conveniently fast convergence to a steady-state. However, this configuration could be exploited by attackers that, with a high initial amount of reputation, would have higher chances of succeeding with their attacks. Instead, the convergence to a steady-state is slower with a low initial reputation, but attackers would not

have a multiplier factor to exploit since there would be no difference between new participants and untrusted ones. The results of this analysis are presented in Figure 4c, which show that there is no significant difference in the steady-state by selecting different values of initial reputation. Only the first ten minutes of the simulation are significantly affected. The configuration with no initial reputation, however, exhibits worse behavior than the others. This is caused by the fact that the participants with zero reputations cannot support others' transactions, causing more transactions to get rejected. For this reason, we opt for a low but non-zero value of the initial reputation (i.e., 64).

Reward/Penalty base values. The most relevant parameters in the reward system are the $rewardbasevalue$ and $penaltybasevalue$. The $rewardbasevalue$ should be significantly smaller than the $penaltybasevalue$ to positively reward a consistently correct behavior and heavily penalize misbehaviors, thus reducing the chances of attack success. To study the effects of these two parameters, we fixed the $rewardbasevalue$ – to 256 – and let the $penaltybasevalue$ vary. Figure 4d displays the comparison of the outcomes of the simulations. As expected, higher penalties result in a lower average reputation.

However, when the penalty is too high, the average reputation decreases significantly, increasing the chances of a transaction being rejected and leading to the worst-case scenario in which the reputation of participants becomes zero. In this case, attackers can exploit the low average reputation of legitimate participants to carry out a successful attack. Therefore, excessively high penalties are not advisable. On the other side, as we tested on attack scenarios, a base penalty not high enough does not penalize the attackers for their misbehavior, enabling them to send false information before being ignored. We choose a value of 512 for the base penalty, which has proven to be a good trade-off value.

Maximum conflicts for the malicious group penalization. The malicious group penalization simulation must be carefully analyzed to prevent the framework from collapsing into undesirable states. The parameter that drives the simulation is the number of tolerable lost conflicts, which determines when to blacklist a participant based on its behavior. If too many lost conflicts are allowed, malicious behavior is more likely to pass unnoticed. It is essential although to consider that spontaneous conflicts may arise naturally. However, our simulations suggest that the number of spontaneous conflicts is, in general, low.

The average reputation of active participants under different values of this parameter is shown in Figure 4e, where it is possible to notice that the least restrictive condition produces the lowest average reputation. This is because if transactions are blacklisted, the evaluation of other transactions is avoided due to the lack of support. Therefore, overall, less penalty is distributed because rejected transactions are not used to detect conflicts. For this reason, we choose two as the maximum number of conflicts before being blacklisted.

Reputation update strategy variance weight. The weight given to the neighbors' variance directly affects the variation of overall reputation (see Section VI-C3). A low neighbor variance is typical of cooperative malicious since they communicate with each other only to increase their reputation. However, also legitimate participants can interact with the same vehicles or RSUs more than once, especially in case of traffic congestion. Figure 4f confronts the average reputations obtained by changing the neighbor variance weight. When the framework reaches a steady-state, the average reputations under the different configurations are distinct and well-separated. This shows that even in non-attack scenarios, neighbors often repeat transactions between each other and that the weight needs to be tuned to obtain a good average reputation. Therefore, a trade-off between usability (i.e., low weight value) and security (i.e., high weight value) must be considered. Since it remains essential to ensure that attackers do not feed each other reputation, we choose 0.67 as an in-between weight that keeps the average reputation high enough while not rewarding repeated neighbor detection.

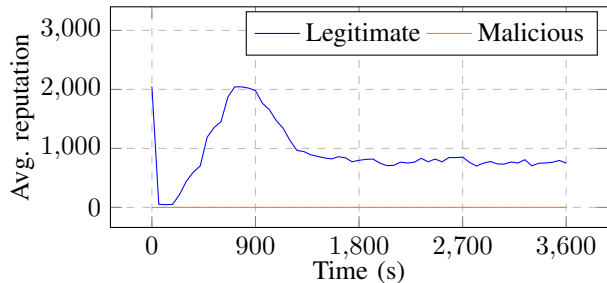### C. Experiment 2: Resistance to Attacks

In this experiment, we analyze the four attack scenarios modeled from the threat model described in Section II: three of them involve uncoordinated attackers that act individually, while the last attack involves a coordinated group of attackers. We analyze each attack independently since they do not affect each other. For each attack scenario, we analyze the value of the average reputation of legitimate and malicious participants as a metric of the goodness of our system. In Figure 6, we represent a graphical representation of the attack scenarios.

Uncoordinated attacks scenarios. We considered three attack scenarios that require no coordination among the attackers and no particular interactions with legitimate participants.
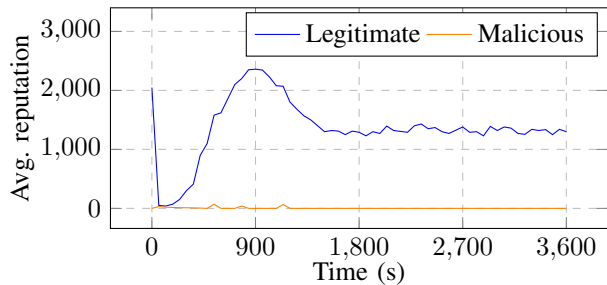
The first attack involves the generation of a significant amount of valid-looking transactions with random sender position and detected vehicle/RSU identifier to exhaust the space in the block. In our experiments, attackers generate and broadcast three times the transactions of the average user. Figure 5a shows the result of the simulation of this attack scenario. The average reputation of attackers is consistently zero, while legitimate participants have, on average, enough reputation for supporting correct traffic information. As visible in Table II, as long as the number of attackers does not exceed 30% of the total number of users, *GOLIATH* is resilient to this attack. This is coherent with the threat model, which considers a relatively small number of attackers, and with the limitations of similar solutions that apply PBFT-based algorithms, which requires that for m attackers, at least $3m + 1$ total nodes exist to ensure stability.

TABLE II: Experiment 2: Analysis of the resilience of *GOLIATH* in terms of the maximum percentage of attackers that can be managed by our framework. For each attack scenario, we show the % of accepted malicious block transactions of the successful attack.
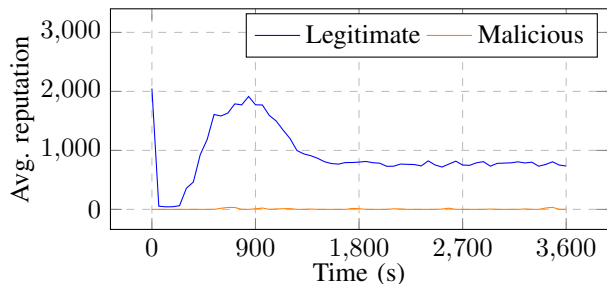
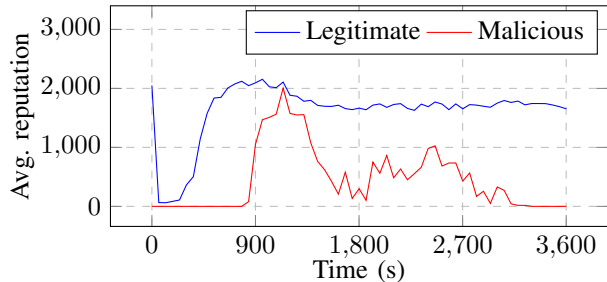| Attack Scenario	% of attackers	% of blocksize
Random (1)	30%	26,80%
Position Changing (2a,b)	80%	80%
Replay (3)	45%	16,4%

(a) Randomly generated transaction attacks.
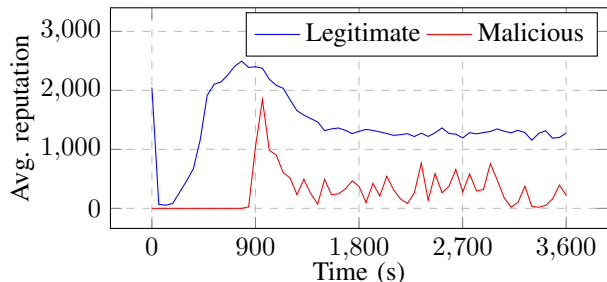

(b) Offset position transaction attacks.

(c) Replayed transaction attacks.

(d) Self-sustaining group of 5 members.

(e) Self-sustaining group of 20 members.

Fig. 5: Experiment 2: Analysis of the average reputation of active legitimate and malicious participants in different attack scenarios.

The second attack involves altering the sender's position to spoof a real position or lower the quality of traffic information. Regarding altering the sender's position, attackers add a constant offset vector to their real positions. Regarding lowering the quality of traffic information, the offset is changed for each spoofed transaction so that the attacker appears to be moving unrealistically. Both attacks generate similar results, shown in Figure 5b, which follow a similar pattern to the first attack. As shown in Table II, *GOLIATH* resists these attacks as long as the number of attackers is lower than 80% of the total users.

The third uncoordinated attack consists in replaying the content of the transactions generated by legitimate participants. In this scenario, attackers wait for legit participants to broadcast their transactions to copy the content except for the sender's identifier. Then, the attackers sign and broadcast the copied transaction. By doing so, an attacker can impersonate a legitimate participant, spoofing its position. Figure 5c shows the outcome of the attack. Similar to the previous scenarios, the attack fails, and the reputation of the attackers remains zero with small spikes during the initial transient. The copied transactions usually are not confirmed by the detected participant, as the attacker is at another position and, hence, filtered by the illegal transactions filtering process of the validation algorithm. As in the previous attacks, the minimum number of attackers required for the framework to not detect the attacks is significantly higher than the one considered in the threat model, as visible in Table II.

Coordinated attack scenario. In this attack scenario, attackers operate as a group. One of the attackers is on-road while the others are in a different position and behave properly until the attack begins. At this point, the malicious group generates fake transactions (between each other), pretending to be close to the one on-road. The on-road attacker eventually stops interacting with the rest of the group, which now has enough reputation to sustain itself autonomously. Then, the self-sustaining malicious group generates fake traffic congestion by changing the advertised position. This attack affects the usability of *GOLIATH* even if a few attackers can succeed consistently. Figure 5d shows the outcome of the attack when the self-sustaining malicious group is composed of 5 members. After an initial peak, the average reputation of the attackers starts dropping and eventually reaches zero. Consequently, the attack partially succeeds only for a limited period. The average reputation of the attackers is kept lower than the one of legitimate participants thanks to both the misbehavior penalization heuristic and the reputation update heuristic. Figure 5e shows the same attack with 20 attackers. Although the attackers do not reach zero reputation during the simulated time, their reputation remains very low and easily distinguishable from the legitimate ones. The number of accepted malicious transactions is less than 50% of the malicious ones in the block, implying that the attack remains contained although not completely blocked.

VIII. Conclusions

This paper presents *GOLIATH*, a blockchain-based decentralized framework that can be used as an alternative to
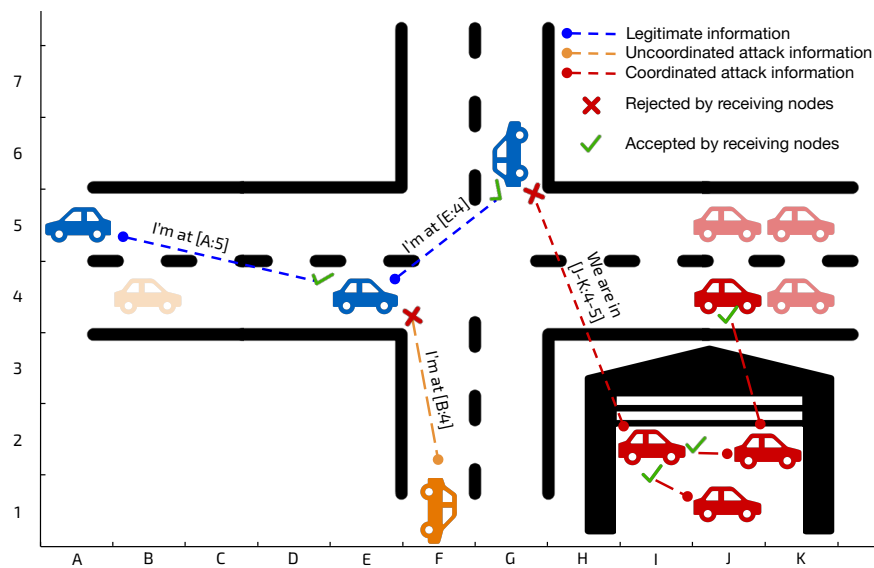
Fig. 6: Experiment 2: Graphical representation of attack scenarios. In blue legitimate participants, in orange an uncoordinated attacker, and in red coordinated attackers. Transparent vehicles represent the fake positions claimed by attackers.

widespread centralized solutions to collect real-time vehicular information. We demonstrated its feasibility in the context of vehicle positioning and traffic information management. Consequently, we defined a threat model tailored to the automotive domain. We validated *GOLIATH* by implementing a network simulator for realistic traffic generation. The results show that the framework can function stably in realistic scenarios while also maintaining high-quality traffic information even in the presence of malicious entities.

One of the limitations of this work is related to the limited availability of real traffic dynamics. Therefore, we resorted to use the SUMO traffic framework and the MoST scenario to produce a realistic simulation. However, MoST contains only one sequence of simulated events. To avoid overfitting, we evaluated our framework in different time windows of the MoST scenario. Another limitation is the computational complexity of the consensus mechanism, which has a quadratic dependency on the number of transactions stored in a block. Therefore, the proposed framework may not scale well with big blocks. The quadratic dependency is determined by the fact that every transaction in the new block has to be compared with all the transactions contained in older blocks or the new blocks themselves. Although it is possible to parallelize the critical loops of the validation algorithm, the computational complexity is still quadratic on the size of the block and cannot be reduced without redesigning the algorithms. An alternative approach and future work is to rely on a statistical approach for the consensus mechanism, where a node has only to check a smaller and fixed portion of the blocks. This approach could effectively reduce the computational complexity to linear, but it would carry the challenge of probabilistic guarantees that need to be carefully proved and enforced.

Overall, this research lays the foundations for further investigation toward decentralized blockchain-based vehicular systems, since to the best of our knowledge, there is no proposal for a decentralized traffic monitoring framework that can be used in a real environment.

REFERENCES

[1] S. E. Co. (2019) Exynos auto 8890 powers ivi system in the new audi a4. [Online]. Available: https://www.samsung.com/semiconductor/minisite/exynos/newsroom/pressrelease/exynos-auto-8890-powers-ivi-system-in-the-new-audi-a4/

[2] K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of DSRC and cellular network technologies for V2X communications: A survey," *IEEE transactions on vehicular technology*, vol. 65, no. 12, pp. 9457–9470, 2016.

[3] Q. Technologies. (2020) Audi of america, virginia dot and qualcomm announce initial c-v2x deployment in virginia. [Online]. Available: https://bit.ly/3s8zyXL

[4] Volkswagen. (2020) Car2x in the new golf: A "technological milestone". [Online]. Available: https://www.volkswagen-newsroom.com/en/stories/car2x-in-the-new-golf-a-technological-milestone-5919

[5] Y. Yahiatene and A. Rachedi, "Towards a blockchain and software-defined vehicular networks approaches to secure vehicular social network," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2018, pp. 1–7.

[6] Y. Yahiatene, A. Rachedi, M. A. Riahla, D. E. Menacer, and F. Nait-Abdesselam, "A blockchain-based framework to secure vehicular social networks," *Transactions on emerging telecommunications technologies*, vol. 30, no. 8, p. e3650, 2019.

[7] Z. Yang, K. Zheng, K. Yang, and V. C. Leung, "A blockchain-based reputation system for data credibility assessment in vehicular networks," in *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*. IEEE, 2017, pp. 1–5.

[8] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2018.

[9] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.

[10] T. Jiang, H. Fang, and H. Wang, "Blockchain-based internet of vehicles: Distributed network architecture and performance analysis," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4640–4649, 2018.

[11] S. Distefano, A. Di Giacomo, and M. Mazzara, "Trustworthiness for transportation ecosystems: The blockchain vehicle information system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2013–2022, 2021.

[12] U. Javaid, M. N. Aman, and B. Sikdar, "Drivman: Driving trust management and data sharing in vanets with blockchain and smart contracts," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–5.

[13] P. K. Sharma, N. Kumar, and J. H. Park, "Blockchain-based distributed framework for automotive industry in a smart city," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4197–4205, 2018.

[14] M. Li, J. Weng, A. Yang, J.-N. Liu, and X. Lin, "Toward blockchain-based fair and anonymous ad dissemination in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 248–11 259, 2019.

[15] X. Huang, C. Xu, P. Wang, and H. Liu, "Lnsc: A security model for electric vehicle and charging pile management based on blockchain ecosystem," *IEEE Access*, vol. 6, pp. 13 565–13 574, 2018.

[16] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, 2017.

[17] D. Wang and X. Zhang, "Secure data sharing and customized services for intelligent transportation based on a consortium blockchain," *IEEE Access*, vol. 8, pp. 56 045–56 059, 2020.

[18] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017.

[19] H. Liu, Y. Zhang, and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Network*, vol. 32, no. 3, pp. 78–83, 2018.

[20] V. Hassija, V. Chamola, S. Garg, D. N. G. Krishna, G. Kaddoum, and D. N. K. Jayakody, "A blockchain-based framework for lightweight data sharing and energy trading in v2g network," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5799–5812, 2020.

[21] F. Knirsch, A. Unterweger, and D. Engel, "Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions," *Computer Science-Research and Development*, vol. 33, no. 1, pp. 71–79, 2018.

[22] S. Weckert, "Google Maps Hacks," *Artwork: performance and installation*, 2020. [Online]. Available: simonweckert.com/googlemapshacks.html

[23] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117 134–117 151, 2019.

[24] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*. IEEE, 2017, pp. 557–564.

[25] S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system," *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, vol. 4, 2008.

[26] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, p. 1, 2012.

[27] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *25th USENIX security symposium*, 2016, pp. 279–296.

[28] M. Castro, B. Liskov *et al.*, "Practical Byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[29] IOST. (2017) IOST: The Next-generation, Secure, Highly Scalable Ecosystem for Online Services. [Online]. Available: https://iost.io/624/

[30] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*. IEEE, 2017, pp. 1–5.

[31] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Computing*, vol. 5, no. 1, pp. 31–37, 2018.

[32] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE access*, vol. 6, pp. 11 676–11 686, 2018.

[33] B. Lee and J.-H. Lee, "Blockchain-based secure firmware update for embedded devices in an internet of things environment," *The Journal of Supercomputing*, vol. 73, no. 3, pp. 1152–1167, 2017.

[34] J. Gu, B. Sun, X. Du, J. Wang, Y. Zhuang, and Z. Wang, "Consortium blockchain-based malware detection in mobile devices," *IEEE Access*, vol. 6, pp. 12 118–12 128, 2018.

[35] H. Huang, X. Chen, Q. Wu, X. Huang, and J. Shen, "Bitcoin-based fair payments for outsourcing computations of fog devices," *Future Generation Computer Systems*, vol. 78, pp. 850–858, 2018.

[36] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 78–85, 2017.

[37] T. for London. Our open data. [Online]. Available: https://tfl.gov.uk/info-for-open-data-users/our-open-data

[38] N. Buch, M. Cracknell, J. Orwell, and S. A. Velastin, "Vehicle localisation and classification in urban CCTV streams," *16th ITS World Congress*, pp. 1–8, 2009.

[39] H. Rehborn, B. S. Kerner, and R.-P. Schäfer, "Traffic jam warning messages from measured vehicle data with the use of three-phase traffic theory," in *Advanced Microsystems for Automotive Applications 2012*. Springer, 2012, pp. 241–250.

[40] D. Barth. (2009) The bright side of sitting in traffic: Crowdsourcing road congestion data. [Online]. Available: https://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html

[41] M. Amin-Naseri, P. Chakraborty, A. Sharma, S. B. Gilbert, and M. Hong, "Evaluating the reliability, coverage, and added value of crowdsourced traffic incident reports from Waze," *Transportation research record*, vol. 2672, no. 43, pp. 34–43, 2018.

[42] G. Leduc, "Road traffic data: Collection methods and applications," *Working Papers on Energy, Transport and Climate Change*, vol. 1, no. 55, pp. 1–55, 2008.

[43] A. Shostack, *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[44] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 251–260.

[45] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, "Everything you wanted to know about the blockchain: Its promise, components, processes, and problems," *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6–14, 2018.

[46] S. Brands and D. Chaum, "Distance-bounding protocols," in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1993, pp. 344–359.

[47] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[48] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 583–598.

[49] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym schemes in vehicular networks: A survey," *IEEE communications surveys & tutorials*, vol. 17, no. 1, pp. 228–255, 2014.

[50] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[51] NEM, *NEM Technical Reference*, 2018. [Online]. Available: https://nemplatform.com/wp-content/uploads/2020/05/NEM_techRef.pdf

[52] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[53] K. J. O'Dwyer and D. Malone, "Bitcoin mining and its energy footprint," in *25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*. IET, 2014, pp. 280–285.

[54] P. Daian, R. Pass, and E. Shi, "Snow white: Robustly reconfigurable consensus and applications to provably secure proof of stake," in *International Conference on Financial Cryptography and Data Security*. Springer, 2019, pp. 23–41.

[55] P. Jovanovic, "Byzcoin: Securely scaling blockchains," *Hacking, Distributed, August*, 2016.

[56] "Veins." [Online]. Available: https://veins.car2x.org/

[57] "Omnet++." [Online]. Available: https://omnetpp.org/

[58] "Sumo." [Online]. Available: https://www.eclipse.org/sumo/

[59] L. Codeca and J. Härri, "Towards multimodal mobility simulation of C-ITS: The Monaco SUMO traffic scenario," in *2017 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2017, pp. 97–100.

Davide Maffiola Davide Maffiola is a graduated student from Politecnico di Milano. There, he earned both his BSc and his MSc with honors in Computer Science and Engineering in 2018 and 2020 respectively. His main interests are in Computer Network Security, Embedded Systems, and Cyber-Physical Systems Security. He is currently working in the area of Risk Advisory.

Stefano Longari Stefano Longari received a Ph.D. in Information Technology from Politecnico di Milano, where he currently works as a Postdoctoral Researcher in NECST laboratory as part of the System Security group inside the Dipartimento di Elettronica, Informazione e Bioingegneria. His main research focus is automotive security and the use of novel technologies such as machine learning intrusion detection to secure the development of smart mobility and smart city ecosystems.

Michele Carminati Michele Carminati received his Ph.D. degree cum laude in Information Technology from Politecnico di Milano in Italy, where he is currently a Contract Professor and a Postdoctoral Researcher working at NECST laboratory as part of the System Security group inside the Dipartimento di Elettronica, Informazione e Bioingegneria. His research revolves around the application of machine learning methods in various cybersecurity-related fields, ranging from cyber-physical and automotive systems to binary and malware analysis, going through anomaly and intrusion detection. He is actively involved in research projects funded by the European Union, and he is also co-founder of Banksealer, a Fintech spinoff of Politecnico di Milano.

Mara Tanelli Mara Tanelli (M'05, SM'12) is currently a Full Professor of Automatic Control at the Politecnico di Milano, where she earner her Ph.D. in Information Engineering with honors in 2007. Her main research interests are in Automotive Control, Smart Mobility and Industrial Analytics. She is co-author of more than 170 peer-reviewed publications in these research areas. Prof. Tanelli is a member of the Conference Editorial Board of the IEEE Control Systems Society (CSS). She is AE for the IEEE Transactions on Control Systems Technology and for the IEEE Transactions on Human-Machine Systems. Since 2021 she is Chair of the Technical Committee (TC) on Automotive Controls of the IEEE CSS and vice-chair for publications of the corresponding IFAC TC.

Stefano Zanero Stefano Zanero (M'03, SM'10) received a Ph.D. degree in Computer Engineering from Politecnico di Milano, where he is currently an Associate Professor with the Dipartimento di Elettronica, Informazione e Bioingegneria. His research focuses on cybersecurity, in particular cyber-physical systems, malware analysis, and fraud and anomaly detection through the use of machine learning. Prof. Zanero is a Distinguished Contributor of the IEEE Computer Society, where he currently serves on the Board of Governors.