SENATE: A Permissionless Byzantine Consensus Protocol in Wireless Networks for Real-Time Internet-of-Things Applications

Zhiyuan Jiang, Zixu Cao, Bhaskar Krishnamachari, Sheng Zhou, Zhisheng Niu, *Fellow, IEEE*

***Abstract*—The blockchain technology has achieved tremendous success in open (permissionless) decentralized consensus by employing proof-of-work (PoW) or its variants, whereby unauthorized nodes cannot gain disproportionate impact on consensus beyond their computational power. However, PoW-based systems incur a high delay and low throughput, making them ineffective in dealing with real-time Internet-of-Things (IoT) applications. On the other hand, byzantine fault-tolerant (BFT) consensus algorithms with better delay and throughput performance cannot be employed in permissionless settings due to vulnerability to Sybil attacks. In this paper, we present Sybil-proof wirelEss Network coordinAte based byzanTine consEnsus (SENATE), which has the merits of both real-time consensus reaching and Sybil-proof, i.e., it is based on the conventional BFT consensus framework yet works in open systems of wireless devices where faulty nodes may launch Sybil attacks. As in a Senate in the legislature where the quota of senators per state (district) is a constant irrespective with the population of the state, "senators" in SENATE are selected from participating distributed nodes based on their wireless network coordinates (WNC) with a fixed number of nodes per district in the WNC space. Elected senators then participate in the subsequent consensus reaching process and broadcast the result. Thereby, SENATE is proof against Sybil attacks since pseudonyms of a faulty node are likely to be adjacent in the WNC space and hence fail to be elected. Simulation results reveal that SENATE can achieve real-time consensus (consensus delay under one second) in a network of hundreds of nodes.**

***Index Terms*—Internet-of-Things, byzantine fault-tolerant, Sybil attack, wireless network, permissionless blockchain, real-time consensus**

I. Introduction

In recent years, digital cryptocurrency has seen an explosive development, both in academia and financial markets. Behind its tremendous success, the key enabling technology of digital cryptocurrency is the *blockchain* [1], [2] which combines several judiciously designed techniques to facilitate trusted distributed ledgers such that intermediary can be eliminated during transactions. In particular, the Bitcoin blockchain ingeniously adopts the proof-of-work (PoW) for mining to, among other purposes, deal with identity attacks (Sybil attacks) in open (permissionless) systems wherein the identities of participating nodes are not assumed to be known a priori. Specifically, each node, whether it is authentic or a pseudonym, must solve a cryptographic puzzle to participate in the block generation process and obtain rewards, hence the so-called mining. Therefore, the impact of a mining node is directly tied to its computational power, irrespective of the number of identities it has. In addition to the successful application in cryptocurrency, the blockchain has also been applied in, e.g., modern business and industry, to achieve reliable and robust consensus [3], [4].

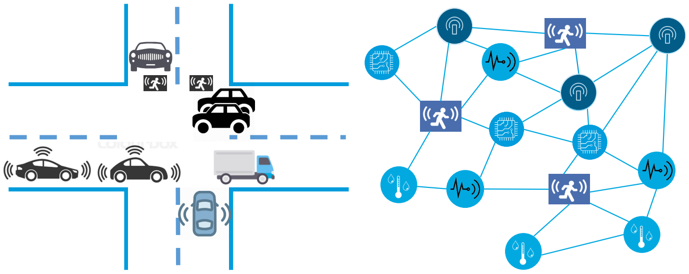


Fig. 1. Time-critical application scenarios, e.g., autonomous driving systems wherein vehicles and pedestrians go through intersections based on distributed consensus (left); Internet-of-Things where terminals (drones, sensors and actuators) act based on coordinated and synchronized behavior (right).

According to the necessity of a prior identity authorization procedure, blockchain technologies can be categorized by *permissionless* and *permissioned* blockchains. Permissionless blockchains, such as Bitcoin [1] and Ethereum [5], are applied in open systems wherein faulty nodes may apply Sybil attacks, the counteraction of which usually involves enforcing a strict coupling between the consensus impact of a node and the computational power (PoW for Bitcoin) or the resources (proof-of-stake for Ethereum Casper) of the node. Despite the robustness against Sybil attacks, a price is payed that permissionless blockchains usually suffers from high processing delay (6 blocks are recommended in Bitcoin which amounts to one hour) and low throughput (at most 7 transactions per second [6] for Bitcoin). Many existing works try to remedy this issue [6], [7], however, the inherent

Z. Jiang and Z. Cao are with Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China. Emails: {jiangzhiyuan,caozixu}@shu.edu.cn.

B. Krishnamachari is with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA. Email: bkrishna@usc.edu.

S. Zhou and Z. Niu are with Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China. Emails: {zhiyuan, sheng.zhou, niuzhs}@tsinghua.edu.cn.

This work was supported by the National Key R&D Program of China (Grant No. 2017YFE0121400), the Nature Science Foundation of China (No. 61871254, No. 91638204, No. 61861136003, the program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, and research funds from the Shanghai Institute for Advanced Communication and Data Science (SICS). The corresponding author is Sheng Zhou.

TABLE I
SENATE IN COMPARISONS WITH OTHER BLOCKCHAINS

	Bitcoin	Bitcoin-NG	PeerCensus	ByzCoin	SENATE
Open system	Yes	Yes	Yes	Yes	Yes
Delay	Large	Medium (10 s-100 s)	Large	Medium (∼60 s)	Small (∼1 s)

mining based probabilistic consensus reaching technique is the key limiting factor. Solidus [8] is a novel blockchain technology that uses PoW to select the leaders and then adopts conventional byzantine fault-tolerant (BFT) protocols to establish consensus. However, since it still needs PoW to achieve permissionless consensus, the delay is inevitably large, and moreover, it is unreasonable to assume the terminals in an IoT system can afford to run the computational-expensive PoW operations. On the other hand, permissioned blockchains, such as Hyperledger Fabric [9], need not be wary of Sybil attacks since all participating nodes have gone through an explicit authentication process such that they can be trusted. Adopting a long line of existing research on BFT protocols [10]–[13], the processing delay and throughput of permissioned blockchains can be dramatically improved; a nice comparison between traditional BFT protocols and permissionless blockchains was presented by Vukolić [14].

Meanwhile, the *Internet-of-Things* (IoT) [15], [16] is envisioned as a key transforming technology in the future. It is identified that [16] low latency and security are, among others, the most critical challenges for IoT, especially for applications that have stringent latency and safety requirements, e.g., real-time sensing and control. However, as mentioned above, neither permissionless nor permissioned blockchains can suffice to achieve open and real-time consensus for IoT systems in the presence of malicious nodes, due to the fact that PoW entails an inevitably high latency and permissioned blockchains require a specific authentication process, i.e., closed. Our work aims to fill up this gap.

In this paper, a large-scale dense wireless network scenario is considered, which is likely to be encountered in the future in IoT deployments as well as in the context of vehicular networks and intelligent transportation (Figure 1). The main contributions include:

- We come up with a solution for Sybil-proof BFT consensus which offers the benefits of both permissioned and permissionless blockchains (cf. Table I for details), i.e., low-delay and high-throughput BFT consensus in permissionless systems. The proposed protocol, namely Sybil-proof wireless network coordinate (WNC) based byzantine consensus (SENATE), consists of three major phases: sortition, senator selection and byzantine agreement. SENATE thwarts the Sybil attack by exploiting the fact that even a faulty node cannot forge its wireless channel to other nodes such that a unique wireless fingerprint can be leveraged to identify nodes; a fully decentralized approach is proposed to cross-check the credibility of nodes.
- Analysis on time duration of SENATE shows that SENATE achieves real-time consensus in permissionless systems, with consensus delay on the order of hundreds of milliseconds for a network of about 100 nodes, based on long-term-evolution (LTE) numerologies.

Notations: Throughout the paper, we use boldface uppercase letters, boldface lowercase letters and lowercase letters to designate matrices, column vectors and scalars, respectively. The transpose of a matrix is denoted by $(\cdot)^{\mathsf{T}}$. $X_{i,j}$ and x_i denotes the (i,j)-th entry and i-th element of matrix $\boldsymbol{X}$ and vector $\boldsymbol{x}$, respectively. The ℓ_p and nuclear norm of a matrix are denoted by $\|\cdot\|_p$ and $\|\cdot\|_*$, respectively. The vector consisting of the diagonal entries of a square matrix is denoted by $\mathsf{diag}[\cdot]$. The trace of a matrix is denoted by $\mathsf{tr}[\cdot]$. The matrix with all entries being one is denoted by $\mathbf{1}$, and likewise zero matrix is denoted by $\mathbf{0}$. A quantity that has the same order of x is denoted by $\mathcal{O}(x)$.

II. Related Work

The idea of only allowing selected nodes to participate in BFT consensus reaching is shared by, e.g., NEO [17] and Algorand [6]. NEO is a delegated BFT consensus based blockchain in which a small number of servers are *statically* configured to run consensus on behalf of a larger open network. Similar with SENATE, Algorand counteracts Sybil attacks by adopting a sortition phase; the difference is that a random verifiable function based solution combining with proof-of-stake (PoS) is leveraged by Algorand, whereas SENATE is based on the underlying wireless channels. Because SENATE is not employing PoS, it is not tied to a digital currency and thus can be applied more broadly to achieve consensus in wireless systems; moreover, it avoids the unfairness introduced by PoS which intentionally favors participants with more resources.

Recently, the concept of Proof-of-location (PoL) is emerging rapidly which is in line with our work. FOAM [18] adopted crypto spatial coordinate to provide an alternative, more secure location service compared with GPS. In wireless networks, PoL was adopted by Dasu *et al.* [19] to replace PoW for faster transactions. The PoLs were generated by authorities such as wireless network operators and hence some notion of centralization was introduced. SENATE also uses the concept of PoL whereas, on a high level, nodes generate PoLs in a fully-decentralized manner without any trusted authority.

There have been some work where wireless channel fingerprints are utilized to protect against Sybil attacks (cf. [20] for a survey and [21], [22] for a signalprint-based approach). However, most existing work relies on a trusted authority to verify the wireless channels of nodes [21], or pre-distribute encrypted keys [20]. In [22], a wireless ad hoc network of commodity 802.11 devices was considered; a view selection policy based on signalprint observations was proposed. In contrast, our work considers a fully decentralized wireless network and a novel WNC based protocol, i.e., SENATE, is proposed; compared with existing work, SENATE has much lower running time and better understandability and hence more favourable for real-time implementations. A comprehensive survey of existing works on blockchain in IoT can be found in [23]–[25].

TABLE II
DESCRIPTION OF KEY NOTATIONS

N:	Number of nodes.
S:	Number of candidates.
K:	Number of senators.
F:	Number of faulty nodes.
d_{ij}:	Distance between the node-i and node-j.
p_n:	Transmission probability of node-n in the ALOHA game.
c:	Relative cost per transmission.
$\boldsymbol{X}$:	Geographical location coordinates of nodes.
$\hat{\boldsymbol{X}}$:	Reconstructed coordinates of nodes given $\boldsymbol{D}$.
$\boldsymbol{D}$:	Euclidean distance matrix of nodes.
$\boldsymbol{E}$:	Arbitrary error matrix introduced by faulty nodes.
T_{x}:	Time consumed by running procedure x.

III. SYSTEM MODEL

Real-time Internet-of-Things applications require time-critical messages to be transmitted properly. For example, in a vehicular network that supports autonomous driving, safety-related messages need to be conveyed to the related vehicles as soon as possible. At the same time, fake messages and those terminals who transmit them should be identified and avoided.

We consider a wireless network consisting of N geographically distributed nodes with full connectivity, namely any pair of nodes in the network are within each other's radio communication range. The system is open, or permissionless, in the sense that any node can join the system without prior identity authentication. The objective is for the *good* nodes to reach a valid (the definition for validity is addressed later) consensus on a set of values over a certain time period such that deterministic concerted actions can be carried out, in the meantime, subject to malicious behaviors by *faulty* nodes. Note that the considered scenario distinguishes from the state machine replication wherein a log is proposed by a client and different nodes agree on the same log record; here different nodes may have different set of initial values, e.g., sensory data from environment, and hence a reasonably good (valid) consensus needs to be reached. For clarity, key denotations are listed in Table II.

Unlike existing work on byzantine consensus which mainly adopts the Internet as the overlay network, a wireless overlay is considered. In this regard, the behavior of a faulty node should be clarified. Specifically, the following assumptions are made in this paper.

- The objective of a faulty node is to *rig* the consensus reaching process to benefit itself, rather than halting the process.
- To achieve its purpose, possible malicious behaviors include: (1) Byzantine node [10], namely it does not comply with the protocol and can report arbitrary messages; (2) Sybil attack [26]: it can generate pseudonyms to gain inappropriate power in the process of reaching consensus.
- In the overlay wireless network, a faulty node does not block or interfere with other nodes' transmissions and messages.

The first assumption describes the motive of a faulty node and therefore has implications on the other two assumptions. The second assumption simply states that, on a message level, there is no restriction on the behavior of a faulty node, both from the perspectives of the message content and the identity of the message sender. In most existing byzantine agreement protocols with the Internet as overlay [10], the third assumption is also implied which limits the malicious behavior of a node to itself; whereas in wireless networks with the broadcast nature of electromagnetic waves, this assumption has more implications, meaning that a faulty node is assumed to comply with the communication protocol. For instance, a faulty node would not transmit when another node is scheduled (by the consensus protocol). This assumption stems in large part from the first assumption, since messing with the communication protocol, e.g., transmitting with a high power and thus blocking other nodes, leads to retransmissions and hence halting the consensus reaching process. Besides, the following two reasons also justify the assumption: (1) an attack becomes trivially devastating without this assumption, namely a faulty node with sufficient transmit power can block other transmissions all the time to prevent reaching consensus; (2) a node not complying with the communication protocol is obviously malicious and easy to spot.

In this work, we assume nodes can obtain ranging estimations based on others' pilot signals. However, we do not focus on specific methods to obtain the distance estimations; they can be based on receive signal strength (RSS), time of arrival (ToA) or other approaches which have been studied extensively [27]. The net effect of ranging estimations is considered by a statistical model, i.e., $\hat{d}_{ij} = \sigma_{ij} d_{ij} + n_{ij}$, where d_{ij} denotes the geographic distance between node-i and node-j and hence $d_{ji} = d_{ij}$, the distance estimation at node-j from node-i is denoted by $\hat{d}_{ij}$, the estimation error is introduced by multiplicative and additive random coefficients σ_{ij} and n_{ij}, respectively. In the wireless localization literature, it is usually assumed that

- For ToA-based ranging estimations, $\sigma_{ij} = 1$ and n_{ij} is modeled as a Gaussian distributed variable.
- For RSS-based ranging estimations, the shadow fading is modeled as σ_{ij} which is assumed to be log-normal distributed; it is often termed as log-normal shadow fading by taking logarithm on both sides.

Considering the ranging estimation error, it is observed thereby that the RSS-based approach is effective with short distances since there is a multiplicative error component; the ToA-based approach applies to a more wide range of distances although it may require a central node to calibrate the clocks of terminals to ensure synchronization.

IV. SENATE

SENATE consists of three major phases: sortition, senator selection and byzantine agreement.

A. Sortition

In the sortition process, the objective is to prevent faulty nodes to generate *arbitrarily* many pseudonyms; note that this does not mean we eliminate the Sybil attack by sortition completely. The key to achieve this is by developing an

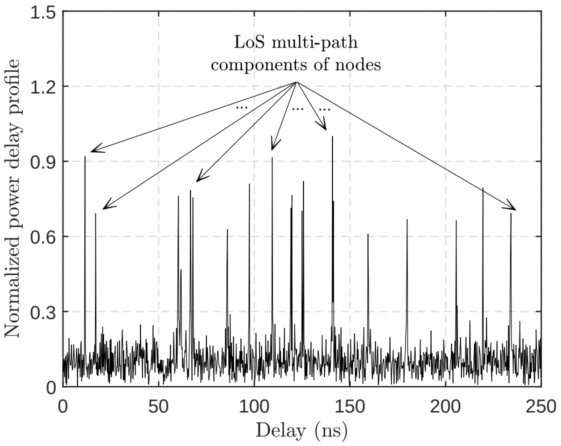


Fig. 2. An exemplary PDP of ultra-wide bandwidth signals (bandwidth of 10 GHz) which is leveraged to estimated the number of nodes in the system by counting the number of significant MPCs.

ALOHA game with selfish users [28] such that no one can cheat based on the Nash equilibrium arguments.

In the ALOHA game, a critical requirement is that every good player (node) knows the total number of users (including faulty nodes) in the system to determine its action; in the considered scenario, this requirement poses a challenge in the presence of faulty nodes, namely faulty nodes can let good nodes believe there are more nodes in the system such that good nodes may behave more conservatively and hence benefit faulty nodes. To prevent this, a *chorus* procedure is proposed which leverages the unique pattern of channel power delay profile (PDP) to estimate the total number of participating nodes. The key observation is that a faulty node cannot forge multi-path components (MPCs) and hence this feature can be utilized, especially with line-of-sight (LoS) transmission environment, to estimate the node quantity. In Fig. 2, an exemplary PDP [29] is depicted which has 20 nodes located in a square area of dimension 100 meters and the signal bandwidth is 10 GHz.

Chorus: In this procedure consisting of $T_{\textsf{Chorus}}$ time slots, each node is supposed to randomly select one time slot to receive and, on the other hand, transmit pilot symbols in the remaining $T_{\textsf{Chorus}} - 1$ time slots. An analysis in Section IV-A2 shows that even in the presence of faulty nodes, this procedure is robust. In the receive time slot of, e.g., node-i, it estimates the PDP of receive signal and calculates the number of MPCs. In an LoS environment, this measurement gives an accurate estimation of the number of transmitters in the system; moreover, even a faulty node cannot generate multiple MPCs given its location.

This is analogous to let all nodes perform chorus first such that every node can have an estimation of the population based on nodes' unique timbre. The procedure lasts $T_{\textsf{Chorus}}$ time slots since we assume that nodes cannot operate in the full-duplex mode; otherwise the procedure can be shortened to one time slot wherein every node transmit and receive simultaneously. The detailed procedure, as well as the analysis for appropriate $T_{\textsf{Chorus}}$, is given in Section IV-A2.

ALOHA game based sortition: Given each node's estimations about the total number of nodes in the system, we let every node be selfish in the ALOHA game to prevent faulty nodes from gaining advantages; this all-be-selfish methodology is essentially identical with the blockchain technology which allows all miners to compete for the opportunity to register a block. In particular, an ALOHA random access game with selfish users is implemented. It is roughly described as follows.

- Every node is selfish, in the sense that they all want to transmit as soon as possible in a collision-free time slot. However, after a successful transmission, a good node would stop competing whereas a faulty node might keep on transmitting to launch Sybil attacks.
- Once a node successfully transmits in a time slot (collision-free), it is selected as the s-th candidate where $s \in \{1, ..., S\}$ denotes the s-th successful transmission in the ALOHA game; then it transmits a pilot signal for ranging estimations immediately afterwards.

By this definition, it can be shown (Section IV-A) that a Nash equilibrium exists based on which every node adopts the same mixed-strategy [28]: transmit in each time slot with a probability p_n with $0 < p_n < 1$, and no one can benefit by changing the strategy unilaterally.

After a quorum of S candidates is reached, the sortition phase terminates with S candidates going into the senator selection in the next phase. Note that a Sybil attack is still possible; a faulty node may occupy several seats among the candidates.

The sortition phase is described in Algorithm 1. We elaborate on several details as follows.

1) Nash Equilibrium of ALOHA Game: An ALOHA game is described as follows. Every node participates in the game.

- In a time slot a node successfully transmits (collision-free), the node receives a payoff of $1 - c$ where $c \in [0, 1]$ denotes the one-time transmission cost and leaves the game.[1]
- In a time slot if a collision happens, every transmitting node receives a payoff of $-c$.

A detailed payoff function is described in Table IV. Every node's goal is to maximize its payoff in a single time slot and the game is repeated. Based on the game setting, we can prove the existence of Nash equilibrium.

Theorem 1: There exists a Nash equilibrium that every node adopts the same mixed-strategy: transmit with probability $p(c, N)$ in each time slot, where

$$p(c, N) = 1 - \sqrt[N-1]{c}. \qquad (1)$$

Proof: The proof is based on [28]. See Appendix A for details. ■

Remark 1: Theorem 1 indicates that by allowing every node to be selfish, a symmetric equilibria exists, namely even a faulty node cannot improve its payoff by changing its transmission probability unilaterally. Specifically, if it increases its transmission probability, there would be more collisions and

[1] A faulty node may otherwise stay in the game and keep playing.

Algorithm 1: Sortition

```
Input: Node-$1, ..., N$;
Output: Candidate-$1, ..., S$;
Chorus Procedure
Every node (node-$i$) uniform-randomly selects
  $1 \leq t_i \leq T_{\mathsf{Chorus}}$.
for $t = 1 : T_{\mathsf{Chorus}}$ do
    if $t = t_i$ then
        Node-$i$ receives signals and estimates the
          number of transmitting nodes based on the
          receive PDP; the estimation is denoted by $\sigma_i$.
        The total number of nodes estimated by node-$i$
          is $\hat{N}_i = 1 + \frac{T_{\mathsf{Chorus}}}{T_{\mathsf{Chorus}}-1}\sigma_i$.
    else
        Node-$i$ transmits a pilot symbol in the $t$-th time
          slot.
ALOHA Game based Sortition
Every node can transmit in every time slot.
for $s$=1:$S$ do
    The $s$-th candidate is selected when a new
      successful transmission happens and the
      corresponding transmission node is candidate-$s$;
      afterwards, it transmits a pilot signal.
return Candidate-$1, ..., S$.
```

the payoff decreases due to cost c; if it decreases its chance then its success chance also decreases. The transmission cost per time slot c clearly plays a critical role here, which denotes the relative cost per transmission as compared with one successful transmission. In practice, we propose that, aside from the power and resource cost due to wireless transmissions, an economic approach can be applied whereby a small fee is charged for every sortition transmission to enhance the robustness of the process.

Remark 2: Theorem 1 shows that the access probability is in fact changing mildly with N. Therefore, it can be concluded that SENATE is insensitive to the estimation error of N, which is introduced in the Chorus stage.

2) Analysis on Sortition Duration: **Analysis on Chorus Duration**: When a node, e.g., node-i, receives signals, assuming its estimation on the number of transmitting nodes is correct, the probability of a good node transmitting in this time slot is

$$p_{\mathsf{g},j} = 1 - \frac{1}{T_{\mathsf{Chorus}}}, \text{ node-}j \text{ is a good node and } j \neq i. \quad (2)$$

Therefore, the unbiased estimate should be

$$\hat{N}_i = 1 + \frac{T_{\mathsf{Chorus}}}{T_{\mathsf{Chorus}} - 1} q_i, \quad (3)$$

where q_i denotes the estimation of transmitting node in node-i's receive time slot. The optimal attack a faulty node can launch is to let good nodes believe there are more nodes in the system such that the transmission strategy of the latter would be more conservative. Therefore, the worst effect all faulty nodes can conjure is by always transmitting, and thereby,

$$\hat{N}_i \leq 1 + \frac{T_{\mathsf{Chorus}}}{T_{\mathsf{Chorus}} - 1} F + \frac{T_{\mathsf{Chorus}}}{T_{\mathsf{Chorus}} - 1} \sum_{n=1}^{N-F-1} m_n, \quad (4)$$

where m_n is a Bernoulli random variable with parameter $1 - \frac{1}{T_{\mathsf{Chorus}}}$. By letting T_{Chorus} be sufficiently large compared with N, then

$$\begin{aligned} N < \hat{N}_i &\leq 1 + \frac{T_{\mathsf{Chorus}}}{T_{\mathsf{Chorus}} - 1} F + (N - F - 1) + o\left(\frac{N}{T_{\mathsf{Chorus}}}\right) \\ &= N + o\left(\frac{N}{T_{\mathsf{Chorus}}}\right), \; F \leq N \quad (5) \end{aligned}$$

where $o(\cdot)$ denotes infinitesimal. Therefore, a reasonably good chorus duration is on the order of

$$T_{\mathsf{Chorus}} \approx \mathcal{O}(N) \quad (6)$$

time slots.

Analysis on ALOHA game duration: In the ALOHA game described in the previous subsection, the average time it takes to select S candidates is

$$\begin{aligned} T_{\mathsf{ALOHA}} &= \frac{S}{N p_j (1 - p_j)^{N-1}} = \frac{S}{N(1 - \sqrt[N-1]{c})c} \\ &\xrightarrow{N \to \infty} \frac{S}{-c \log c} \geq eS, \quad (7) \end{aligned}$$

where it can be derived from the last inequality that the optimal selection of c is

$$c_{\mathsf{opt}} = 1/e. \quad (8)$$

Since it may not be practical to choose the cost coefficient c which is related to the relative cost per transmission, we plot the time consumed by the ALOHA game as a function of c in Fig. 3. Without loss of generality, we consider $S = 1$ in the figure. It is observed that the time duration of the ALOHA game is insensitive with c; with a wide range of $c \in (0.1, 0.8)$, the time duration is relatively similar. With this observation, we adopt the results in (7) to represent the ALOHA game duration. The impact of c will be further studied in the simulation results.

Remark 3: Note that the Sortition stage cannot prevent Sybil nodes from entering the next phase of SENATE, neither can it guarantee a majority of non-faulty candidates. It is simply implemented to allow a *finite* number of candidates, among which the faulty and non-faulty nodes are proportional to their true number as shown by the ALOHA game analysis, entering the next phase of SENATE. The elimination of faulty nodes, however, is done by the Senator Selection stage of the scheme as illustrated in what follows.

B. Senator Selection

This phase is dedicated to further removing the pseudonyms generated by faulty nodes, by cross-checking the ranging estimations among nodes in a fully distributed manner.

After S candidates are selected, they no longer follow the ALOHA-based random access protocol. Instead, each candidate is assigned a unique time slot to transmit in a frame of S time slots in this phase.

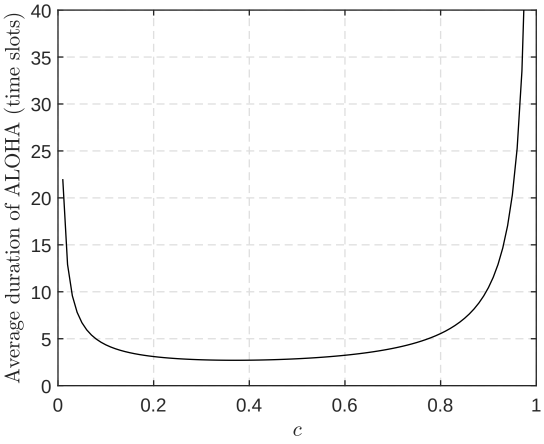


Fig. 3. The average time duration of the ALOHA game to select one candidate with varying cost per transmission c.

Since every candidate has transmitted a pilot signal, every candidate has obtained the distance estimations from other candidates. For candidate-i, its distance estimations are denoted by a vector

$$\hat{\boldsymbol{d}}_i = \left[\hat{d}_{1i}, ..., \hat{d}_{(i-1)i}, 0, \hat{d}_{(i+1)i}, ..., \hat{d}_{Si}\right]^{\mathsf{T}}, i = 1, ..., S. \quad (9)$$

The estimated Euclidean distance matrix (EDM) [30] with squared norm is hence

$$\hat{\boldsymbol{D}} \triangleq \left[\hat{\boldsymbol{d}}_1^2, ..., \hat{\boldsymbol{d}}_S^2\right]. \quad (10)$$

Distance feedback and symmetry verification: Each candidate feeds back its $\hat{\boldsymbol{d}}_i$ in its dedicated time slot. Afterwards, every candidate obtains the distance estimations between any pair of candidates (double-directional) in the network. Note that $d_{ij} = d_{ji}$, $\forall i, j$, and thereby every node can remove suspicious distance feedback based on checking the estimated EDM (here we assume the feedback is perfect)

$$\left|\hat{d}_{ij}^2 - \hat{d}_{ji}^2\right| < \epsilon \quad (11)$$

where ϵ is a constant related to σ_{ij} and n_{ij}. In this case, as long as (11) does not hold, both $\hat{d}_{ij}$ and $\hat{d}_{ij}$ are removed since we cannot tell if node-i or node-j is lying.

Robust WNC generation: Despite the fact that the symmetry verification can, to some extent depending on distance estimation error, eliminate untruthful distance feedback, a faulty node can still launch what we refer to as a "shout attack"[2].

Definition 1 (Shout Attack): A shout attack is that a faulty node pretends to be further away to other nodes, by synchronously adding to the distance estimations to other nodes. In particular, for ToA-based ranging estimations, a faulty node can purposely transmit pilot signals later than supposed, and, accordingly, feed back tampered (larger) distance estimations; for RSS-based ranging estimations, a faulty node can purposely amplify its pilot signal power and, accordingly, feed back tampered (larger) distance estimations. □

By definition, a shout attack cannot be detected by symmetry verification and gives a faulty node arbitrarily many fake geographical locations that are arbitrarily far from its real one. The purpose of a shout attack is hence to create pseudonyms and facilitate the Sybil attack, which causes a severe challenge to SENATE since SENATE uses the location information for Sybil protection.

To thwart the shout attack, we introduce the **seesaw test** based on the following intuition. In the real world with (at most) 3-dimensional space, it is increasingly unlikely that a faulty node, which launches the shout attack, is further away to other nodes proportionally, as the number of nodes grows. This is analogous to placing elastic sticks between each pair of nodes in the system, with the lengths of sticks given by the distance estimations. The circumstance for a faulty node launching shout attack in the 2-dimensional space is illustrated in Figure 4; its related sticks are bent dramatically and hence the elastic force levers it out (screened out by the seesaw test), like being on the smaller-weight side of seesaws. This argument is mathematically formalized in Theorem 2 which states the local error is proportional to the number of good nodes.

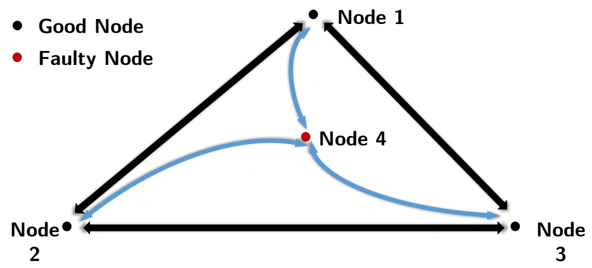


Fig. 4. A seesaw test. A faulty node launching the shout attack would be "levered out".

An iterative WNC calculation with seesaw tests is proposed, the essence of which can be illustrated as follows. In each round of the iteration, every node, e.g., node-i, uses the distance estimation to another node-j to push (resp. pull) node-j if the distance estimation is larger (resp. smaller) than the predicted distance between the two nodes by the current WNC; then node-j moves accordingly. At the end of each round, the node with the largest local error, i.e., its related seesaws are bent the most, is identified as a faulty node and therefore removed; note that a termination criterion (specified later) is added such that the WNC calculation terminates when the system has small prediction error.

K**-means clustering**. After obtaining the coordinates of all candidates, a K-means clustering algorithm [31] is applied to the coordinates such that all candidates are divided into K clusters. Then K representatives, each from one cluster, are selected as senators; this prevents Sybil attacks because pseudonyms of one faulty candidate are likely to fall into the same cluster.

In this phase, the S candidates transmit in a round-robin fashion. The detailed algorithm description of the phase is

[2] Likewise, a "whisper attack" can be defined by which the faulty node pretends to be nearer to other nodes. For ease of exposition, we use the shout attack for illustration henceforth.

Algorithm 2: Senator Selection

```
   Input: Candidate-1, ..., $S$;
   Output: Senator-1,...,$K$; validSenate.
 1 Distance estimation and feedback
 2 for $i$=$1$:$S$ do
 3     Based on the pilot signals received in the sortition
         phase, candidate-$i$ estimates its distance to other
         candidates and feeds back its distance vector $\hat{\boldsymbol{d}}_i$.
 4 EDM symmetry verification
 5 $\boldsymbol{\Sigma} \triangleq |\hat{\boldsymbol{D}} - \hat{\boldsymbol{D}}^{\mathsf{T}}|$.
 6 for Every element $D_{ij}$ in $\hat{\boldsymbol{D}}$ do
 7     if $\Sigma_{ij} > \Delta d$ then
 8         $\hat{D}_{ij} \leftarrow$ invalidValue.
 9 Robust WNC generation
10 At every terminal, generate the WNC simultaneously
     based on the same following procedure:
11 terminate $\leftarrow$ false; $\boldsymbol{X} \triangleq [\boldsymbol{x}_1, ..., \boldsymbol{x}_S]^{\mathsf{T}} \leftarrow \mathbf{0}_{S\times 2}$;
     $\boldsymbol{e} \leftarrow \mathbf{0}_S$.
12 while terminate = false do
13     for $\{i, j\} \in \{1, ..., S\} \times \{1, ..., S\}$ do
14         if $D_{ij} \neq$ invalidValue or $0$ then
15             $w \leftarrow \frac{e_i}{e_i + e_j}$;
               $e_i \leftarrow \frac{|\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 - \hat{d}_{ij}|}{\hat{d}_{ij}} \delta w + (1 - \delta w) e_i$;
               $\boldsymbol{f} \leftarrow w \left( \frac{|\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 - \hat{d}_{ij}|}{\hat{d}_{ij}} \right) (\boldsymbol{x}_i - \boldsymbol{x}_j)$;
16             $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \gamma \boldsymbol{f}$.
17     maxError $\leftarrow \max_i\{e_i\}$; errIndex $\leftarrow \arg\max_i\{e_i\}$.
18     if maxError $> \beta \frac{\sum e_i}{S}$ then
19         Remove candidate-errIndex, and its
             corresponding entries in $\boldsymbol{X}$, $\boldsymbol{e}$ and $\hat{\boldsymbol{D}}$;
           $S \leftarrow S - 1$.
20     else
21         terminate $\leftarrow$ true
22 $K$-means clustering
23 if $S < K$ then
24     validSenate $\leftarrow$ false.
25 else
26     $\boldsymbol{S} \leftarrow$ Kmeans($\boldsymbol{X}, K$); validSenate $\leftarrow$ true
27 return $\boldsymbol{S}$; validSenate.
```

presented in Algorithm 2, and some explanations follow.

1) Robust WNC Generation: The rationale for robust WNC generation is as follows. In the face of EDM estimation error introduced by faulty nodes, i.e., denote

$$\hat{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{E}, \tag{12}$$

where the entries of $\boldsymbol{E}$ can be arbitrarily large considering malicious behavior, our goal is to recover $\boldsymbol{D}$. Towards this end, two structures can be exploited: (a) although the faulty nodes can cause arbitrarily large error, the error is sparse in terms of entries of $\boldsymbol{E}$, i.e., majority is still good; (b) the EDM stems from space of limited dimensionality[3] and hence there are mature tools in distance geometry [30] that can be utilized to verify its authenticity. Thereby, considering 2-dimensional space, the EDM can be written as

$$\boldsymbol{D} = -2\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}} + \mathbf{1}\mathsf{diag}\left(\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}\right)^{\mathsf{T}} + \mathsf{diag}\left(\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}\right)\mathbf{1}^{\mathsf{T}}, \tag{13}$$

where $\boldsymbol{X} \in \mathbb{R}^{S\times 2}$ is the geographical location coordinates of candidates, i.e., $\boldsymbol{X} \triangleq [\boldsymbol{x}_1, ..., \boldsymbol{x}_S]^{\mathsf{T}}$. We then formulate the WNC generation problem as follows, exploiting the sparse error property.

$$\begin{aligned} \textbf{P1:} \quad & \underset{\boldsymbol{X},\boldsymbol{E}}{\text{minimize}} \ \|\boldsymbol{E}\|_0, \\ & \text{s.t.}, (12), (13), \\ & \mathsf{rank}(\boldsymbol{X}) = 2. \end{aligned} \tag{14}$$

The ℓ^0 norm based formulation in **P1** is notoriously non-convex and in fact NP-complete based on compressive sensing theory. Therefore, the ℓ^0 norm in **P1** is relaxed to ℓ^1 norm which often exhibits near optimal performance [32], i.e.,

$$\begin{aligned} \textbf{P2:} \quad & \underset{\boldsymbol{X},\boldsymbol{E}}{\text{minimize}} \ \|\boldsymbol{E}\|_1, \\ & \text{s.t.}, (12), (13), \\ & \mathsf{rank}(\boldsymbol{X}) = 2. \end{aligned} \tag{15}$$

We adopt a data-driven gradient-descend-based method to solve **P2**. Based on an estimation $\hat{d}_{ij}$, we can update $\boldsymbol{x}_i$ (or $\boldsymbol{x}_j$) based on the gradient of the objective function in **P2**:

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \mu \frac{\partial \left| \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 - \hat{d}_{ij}^2 \right|}{\partial \boldsymbol{x}_i}, \tag{16}$$

which corresponds to the 15-step in Algorithm 2. Also note that in the algorithm, we keep track of the local error array $\boldsymbol{e}$ whose element e_i represents the squared distance error related to candidate-i; that is how much candidate-i is levered in the seesaw test (Figure 4). Therefore in the 15-step, we take into account the fact that a candidate with small error should not be updated based on the location of a candidate with large error; the latter is likely to be a faulty node. Based on this argument, we remove the candidate with the largest error at the end of each round, until the error is evenly distributed among candidates which means the error is introduced by ranging estimation instead of faulty nodes.

In the case that the selected senators do not reach a quorum of K, Algorithm 2 returns validSenate = false.

Intriguingly, this method is similar with the spring network based method where any pair of nodes are connected by a spring in, e.g., [27], [33]; the objective in those works is to minimize the elastic potential energy of the system (equivalent with the total square error (TSE) of distance prediction) given the current lengths of springs (distance estimations) by placing the nodes (the distances among nodes are the rest lengths of the springs) on a plane. Although the objective in **P2** is not minimizing the TSE, the presented data-driven gradient-descend-based method turns out to be similar with the Vivaldi algorithm [33], except for the faulty detection.

[3] We use 2-dimensional space for ease of exposition in this paper. However, the generalization to 3-dimensional is considered straightforward.

2) Analysis on Seesaw Test: The seesaw test is based on the rationale that a faulty node implementing the shout attack can be detected because its resultant location would be out of the 2D space. A question arises accordingly: how *out-of-space* the faulty node is given a certain strength of its shout attack, and moreover the effect of the number of good nodes. This question is important because its answer can quantitatively characterize the effectiveness of the seesaw test against forged locations.

In seeking for a concise and illustrative answer, we consider a simplified scenario where there is one faulty node, without loss of generality located at $\boldsymbol{x}_0 = (0, 0)$, who is trying to launch a shout attack to M good nodes located at $\boldsymbol{x}_m = (x_m, y_m)$, $\forall m \in 1, ..., M$. Concretely, we consider that the faulty node adds an arbitrary (independent with real node-locations) error vector to the entries in the EDM that are related to it; note that this is more general than the shout attack whereby the error is added synchronously. The arbitrary error is written based on (12) as

$$\boldsymbol{E} = \begin{bmatrix} 0 & \boldsymbol{e}^{\mathsf{T}} \\ \boldsymbol{e} & \boldsymbol{0}_{M \times M} \end{bmatrix}. \tag{17}$$

Note that no ranging estimation error is considered in this subsection to focus on the synthetic error by the faulty node. The level of out-of-space of the faulty node is measured by

$$h(\varsigma^2) \triangleq \mathbb{E}_{\boldsymbol{X}} \left[\min_{\boldsymbol{Z},\, \mathsf{rank}(\boldsymbol{Z})=2} \left[\min_{\boldsymbol{e},\, \|\boldsymbol{e}\|_1 = M\varsigma^2} \left\| \hat{\boldsymbol{X}} - \boldsymbol{Z} \right\|_2^2 \right] \right], \tag{18}$$

where $\hat{\boldsymbol{X}}$ denotes the reconstructed coordinates of nodes given the tempered EDM in (12) and (17). In other words, the level is quantified by the minimum squared Euclidean distance from the reconstructed coordinate space to its projection into any 2D space, given that the faulty node implements the attack that minimizes this distance. It is essential to note the sequence of minimization, meaning that the faulty node first chooses the error then the closest 2D space is selected. Since this quantity is affected by the locations of good nodes by noting that closer good nodes produce stronger lever force in the seesaw test given the same strength of shout attack, the expectation in (18) is taken over a given location distribution. In the following theorem, we adopt the 2D Gaussian distribution for ease of exposition.

Theorem 2: Assume that the faulty node is at $(0, 0)$ with the attack strength of $\|\boldsymbol{e}\|_1 = M\varsigma^2$, and the good nodes' coordinates are i.i.d. generated based on a Gaussian distribution with zero mean and variance of σ^2, i.e., $(x_m, y_m) \sim (\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, $\forall m \in 1, ..., M$. When the error $\boldsymbol{e}$ is independent with (x_m, y_m), $\forall m \in 1, ..., M$, then

$$h(\varsigma^2) = \min \left\{ (M-1)\sigma^2,\, (M-2)\varsigma^2 \right\}. \tag{19}$$

Proof: See Appendix B. ■

Remark 4: It is shown that a faulty node cannot conceal its lie by noting that $h(\varsigma^2)$ scales with the attack strength ς^2, until ς^2 is comparable with the squared distance measurement (σ^2) whereby the attack becomes quite obvious. In addition, the effect is amplified by approximately M times; this is intuitive since it becomes increasingly more difficult to lie to more good nodes when they form a concrete 2D space. Another note is that $h(\varsigma^2) > 0$ as long as $M \geq 3$, because at least 3 nodes can determine a 2D space.

Remark 5: Theorem 2 assumes that the error matrix $\boldsymbol{E}$ is independent with the coordinates of good nodes, which requires that a faulty node is unaware of the coordinates of other nodes (assumed mostly good nodes) in the sortition phase; this is reasonable because the coordinate information is not accessible in the sortition phase before any distance feedback occurs. Also note that since distance feedback is only performed once at the beginning of Algorithm 2, the faulty nodes cannot gain knowledge of coordinates of other nodes gradually.

Corollary 1: In L-dimensional space,

$$h(\varsigma^2) = \min \left\{ (M-L+1)\sigma^2,\, (M-L)\varsigma^2 \right\}. \tag{20}$$

Remark 6: This corollary generates the effectiveness of the seesaw test to higher dimensions, e.g., 3D scenarios with applications for, e.g., drone swarms.

3) Analysis on Senator Selection Duration: In the senator selection (Algorithm 2), the distance estimation and feedback phase takes $\mathcal{O}(S)$ time slots depending on the distance vector feedback transmission time. The rest of the algorithm can be performed as computational tasks at each node, without any interaction among nodes. The time duration of the computation is difficult to quantify, depending on individual computation capability. However, it is reasonable to neglect the computation delay since, in this case, it does not scale with the number of nodes.

C. Byzantine Agreement

The K senators run a byzantine agreement protocol to reach consensus. We primarily consider the median validity for consensus, which is defined as follows [34].[4] Assume a single consensus value is to be reached upon. Denote by G the sorted array of the initial values of good nodes. Among N nodes, F nodes are faulty and it is assumed that $F \leq t$ and hence $G = [G[0], ..., G[N-F-1]]$.

Definition 2 (Median Validity): We call a value x median-valid, if it holds that

$$G\left[\left\lceil \frac{N-F}{2} \right\rceil - 1 - t\right] \leq x \leq G\left[\left\lceil \frac{N-F}{2} \right\rceil - 1 + t\right]. \square \tag{21}$$

Thereby, we adopt the Jack algorithm [34] which ensures the following properties, as long as the number of faulty senators, i.e., F, satisfies $N \geq 3F + 1$.

- **Agreement**: For every selection of input values and every selection of faulty senators, all good senators can decide on the same value.
- **Termination**: Every good senator can decide on a value in finite time.
- **Median Validity**: The decision is median-valid.

Upon agreement, every senator broadcasts its consensus value, and every good node in the system adopts the majority

[4] Nevertheless, basically any BFT protocol can be plugged into SENATE in this phase, and works well in an open system since we have achieved Sybil-proof in the previous phases.

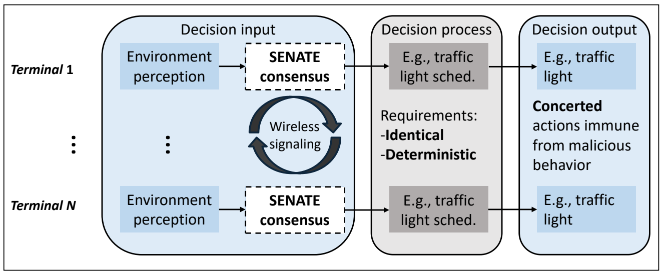


Fig. 5. A decentralized decision framework for IoT, adopting SENATE as a consensus module.

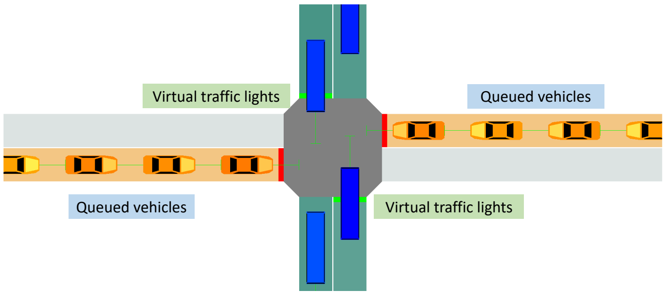


Fig. 6. The simulated intersection setting in SUMO.

TABLE III
TEST ENVIRONMENT

	Item	Value / Version
Test Environments	CPU	Intel(R) Core(TM) i5-7500 CPU @3.40 GHz
	Memory	16 GB
	SUMO	v1.0.1

value. Since consensus is reached in the senate, the majority value can reflect the consensus value and ensures safety and agreement among all nodes.

In this phase, since we have removed Sybil nodes from the selected senators to a great extent, basically any byzantine agreement protocol can be implemented among the K senators. In particular, we adopt the Jack scheme proposed in [34] which consists of the following two major stages:

- **Setup stage**: Each senator broadcasts its initial value and receives other senators' initial values. Thereby, each senator broadcasts its acceptable values and a proposed value, jointly considering its and other senators' initial values.
- **Search stage**: Rotating among $(t+1)$ pre-determined leaders, in each round a leader receives proposals from other senators and accordingly proposes a value based on its acceptable values; if an agreement is reached based on proposals, the leader would propose the agreed value. It is proved that as long as one leader among the $(t+1)$ leaders is a good node, a valid agreement would be reached. Therefore, at most t faulty nodes are allowed in this phase.

The validity is assured by the median validity specified in Section III. The termination and agreement properties are also proved in [34]. In fact, this protocol achieves the optimum safety and 2-approximate of the optimum median validity.

1) Analysis on Byzantine Agreement Duration: A simple calculation on the byzantine agreement protocol duration reveals that it takes approximately $\mathcal{O}(K^2)$ time slots. Combining with the analyses in Section IV-A2 and Section IV-B3, SENATE takes approximately

$$T_{\text{SENATE}} = \mathcal{O}(N) + \mathcal{O}(S) + \mathcal{O}(K^2) \tag{22}$$

time slots. In our implementation, the number of candidates S and the number of senators K are usually much smaller than the number of nodes N, and hence the total time of running SENATE is dominated by the first term in (22).

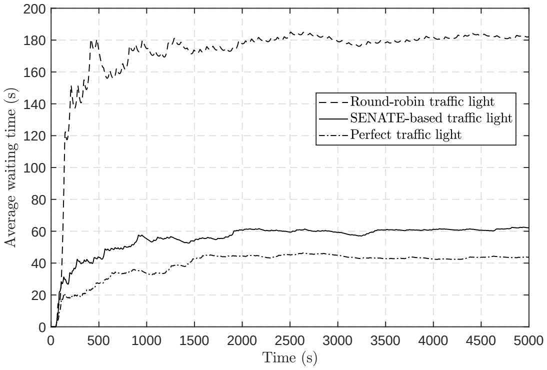


Fig. 7. The average waiting time of one lane in the decentralized virtual traffic light setting.

V. SUMO-BASED TRAFFIC LIGHTS SIMULATIONS

In this section, to test the performance of SENATE, we adopt a real-world application to illustrate an IoT system control framework where SENATE is applied to achieve consensus in decentralized control. In Fig. 5, the high-level structural decentralized decision process is depicted. In order to achieve concerted actions among distributed terminals with possible malicious behavior, the following procedures are adopted:

- A SENATE-based consensus module is applied to make sure that all terminals reach consensus upon the decision input, i.e., environment parameters which determine the decision output.
- The decision process in each terminal should be identical and deterministic, i.e., given the same input, the output should be the same to ensure concerted actions.

We implement this framework on the Simulation of Urban MObility (SUMO) [35] platform, and the test environment is shown in Table III. Autonomous driving with distributed vehicle control is a challenging task in the future since the consensus delay plays a significant role in this scenario to ensure safety. The delay of our framework is low enough to be applied to the scenario. The virtual traffic light control among autonomous vehicles at an intersection is studied (see Fig. 6); the traffic light control is imaginary in the sense that there is no actual traffic lights—they represent a passing law that decentralized autonomous vehicles reach consensus upon. The input environment parameters include the numbers

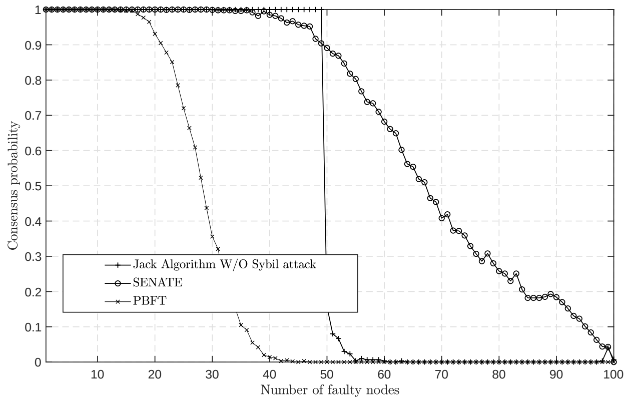


Fig. 8. The probability a valid consensus is reached with 100 nodes, with (aided by SENATE or PBFT) and without Sybil attacks.

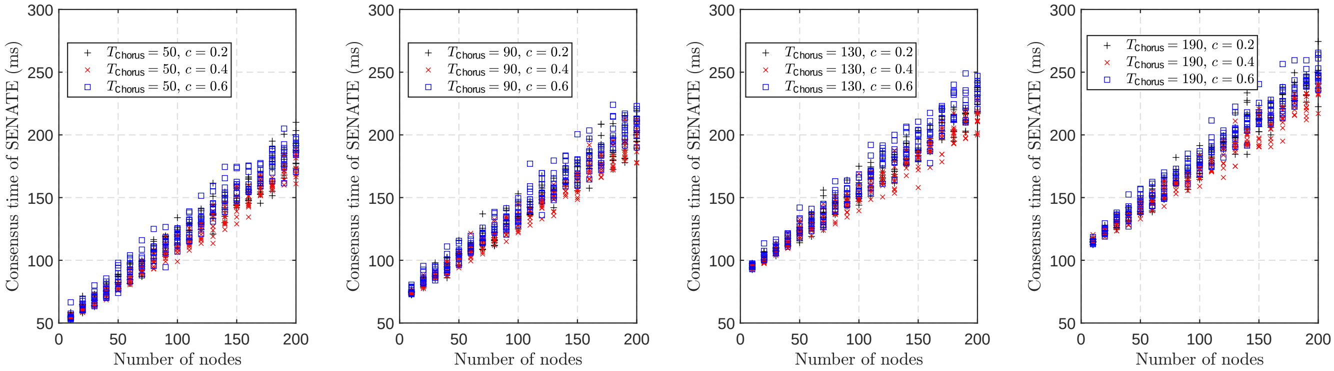
of queuing vehicles in every direction. The decision process in each terminal is quite straightforward in this case—the direction with the largest number of queuing vehicles will get green lights and vice versa; after the queued vehicles in one direction have passed by the intersection, the other direction is allowed to pass; a four-way red light period is enforced to let pedestrians pass. At each time slot (one second in the simulation), the SENATE consensus module is ran once (based on Fig. 9, one second is sufficient) to update the decision input parameters; then the decision is made whether, for each vehicle in each direction, to pass or stop based on the virtual traffic lights. We simulate the average waiting time in each direction based on 1) perfect scheduling, i.e., assuming correct consensus is always reached and thus the direction with the largest number of waiting vehicles is scheduled ; 2) SENATE-based consensus; 3) A conventional round-robin traffic lights, with 5 malicious vehicles (in the same direction) which always apply Sybil attacks and try to rig the consensus to benefit themselves. The ranging estimations at nodes are assumed to be perfect, and the information exchange is simulated by direct modifications of data arrays for nodes. The results are shown in Fig. 7. It is observed that SENATE-based consensus can effectively eliminate the malicious terminals and achieve similar performance as perfect control; the gap to the optimum is significantly smaller compared with the gap with the conventional round-robin traffic lights.

To further investigate the impact of the number of malicious nodes, we run a simulation in Figure 8, where there are $N = 100$ nodes and SENATE selects $S = 50$ candidates and $K = 7$ senators, the number of faulty nodes is shown by x-axis and the performance of consensus probability is shown which is obtained by running the algorithms for 1000 episodes. The node locations are randomly generated in a square with a side length of 200 meter in each episode. The faulty nodes are assumed to always launch Sybil attacks and propose randomly generated values which are deviating from the true values; specifically, the good and faulty nodes' initial values are uniformly randomly generated from interval $[-1, 1]$ and $[99, 101]$, respectively. In the figure, we also simulate the Jack algorithm [34] without faulty nodes launching Sybil attacks for comparisons; the Jack algorithm under this scenario and attack assumptions can ensure consensus when the number of faulty nodes does not reach majority (50%), and cannot otherwise; this is direct consequences of the design of the Jack algorithm and the definition of median validity. The PBFT scheme with Sybil attacks is simulated as well. For PBFT, the sortition phase is added for fair comparisons. In the PBFT simulation, the primary node [10] is always set as a normal node. All faulty nodes do not respond to any requests from other nodes in the "prepare" and "commit" processes to obstruct the consensus to be reached. Based on the design of PBFT, at most 33% faulty nodes can be tolerated. In Fig. 8, it is observed that the performance of PBFT (combined with the sortition phase) is approximately 5% worse than theoretically suggested (33%), due to the sybil attack. It is observed that SENATE perform close-to the conventional BFT protocol as if there was no Sybil attack; this verifies that SENATE is Sybil-proof. We also observe that SENATE perform better when faulty nodes reach majority, which is because the SENATE randomly selects the senators such that there is a probability that the selected senators are dominated by good nodes; the Jack algorithm can also perform, at least, as well as SENATE if a sortition phase is added, but the effect is not shown in the figure to be in line with the original Jack algorithm. It is also clear that SENATE outperforms PBFT significantly because SENATE can mitigate the Sybil attack by cross-checking the wireless coordinates of nodes.

In Fig. 9, we investigate the time consumed by SENATE before reaching a consensus, representing the viability of SENATE for real-time IoT applications. We run SENATE and calculate the interaction time slots that are needed to reach consensus; note that the computational delay is neglected due to unknown computation capabilities of nodes, as well as the fact that the computation is executed in a distributed manner and hence its delay does not scale with the network scale. The number of candidates is set to be $S = \max[10, \lfloor N/3 \rfloor]$, the number of senators is $K = 7$, and we vary the values of the chorus time length $T_{\textsf{Chorus}}$ and relative cost per transmission c. We adopt the LTE numerology that each time slot lasts 0.5 ms. Based on the simulation results, the sortition process, which constitutes the main part of the consumed time of SENATE, is quite robust in terms of values of key parameters, e.g., $T_{\textsf{Chorus}}$ and c, in the sense that the sortition delay does not vary significantly with different parameter values. In general, the consensus delay of SENATE scales with the network scale, as predicted in Section IV-A2. However, it is found that in a network with a scale of hundreds of nodes (10 to 200 nodes are considered in the figure), SENATE can reach a consensus within hundreds of milliseconds. Compared with PoW-based blockchains which generate a block every several minutes, depending on specific technologies involved, SENATE is much more suitable for real-time applications in IoT systems.

VI. Conclusions and Future Work

SENATE is a real-time distributed BFT protocol which is applicable to fully-connected wireless-networked systems without prior identity authentication. In order to prevent malicious nodes to generate an arbitrary number of pseudonyms,



(a)

(b)

(c)

(d)

Fig. 9. The time consumed by SENATE before reaching a consensus with different chorus time duration and relative cost per transmission c. The time slot length is 0.5 ms, as in LTE systems.

SENATE leverages the wireless signals transmitted by nodes to cross-verify their identities in a fully-decentralized (no-trust) manner, based on the fact that pseudonyms are likely to be adjacent geographically. Thereby, only selected nodes, i.e., senators, participate in the final consensus reaching process. Computer simulations show that SENATE is Sybil-proof, by comparing the consensus probabilities between systems wherein faulty nodes launch and do not launch Sybil attacks, respectively. It is also shown that SENATE achieves real-time consensus, with delay on the order of hundreds of milliseconds in a network of approximately 100 nodes based on LTE numerologies, making it more attractive for delay-sensitive applications in IoT systems compared with conventional PoW-based blockchain technologies.

In this paper, we neglect the computation burden of running the consensus protocol in a network that may consist of low-cost IoT devices. Future work should consider to incorporate edge computing to cope with such a computation-intensive task (the most intensive task is identified to be running the K-means clustering algorithm in Algorithm 2 which usually has a complexity of $\mathcal{O}(S^2)$ [36]), while still maintaining low consensus delay and decentralized open-access property.

Appendix A
Proof of Theorem 1

First, we invoke the following lemma which ensures the existence of a Nash equilibrium.

Lemma 1 ([37]): A finite symmetric game has a symmetric mixed-strategy equilibrium.

The finite symmetric game in Lemma 1 denotes a game wherein every player has the same finite action set, and the payoff received by each player given the same action (and other players' actions) is identical, irrespective with the specific player. A mixed strategy is in contrast with a pure strategy; the latter employs a fixed action each time whereas the former can be viewed as a mix (randomized strategy) of the latter. The proof of Lemma 1 is based on the Brouwer fixed point theorem and is omitted for brevity.

Provided the existence of symmetric equilibria, we are ready to derive the transmission probability p for every node. Due to symmetry, it suffices to consider a single node, whose game choice is shown in Table IV.

TABLE IV
Payoff Functions

	All other nodes are silent (p_1)	Some other node transmits (p_2)	Expected payoff
Transmit	$1-c$	$-c$	$(1-c)p_1 - cp_2$
Silent	0	0	0

The probability of each event is

$$\begin{aligned} p_1 &= \Pr\{\mathsf{success}\} = (1-p)^{N-1}; \\ p_2 &= \Pr\{\mathsf{collision}\} = 1-(1-p)^{N-1}. \end{aligned} \tag{23}$$

Based on the principle of indifference [38], both expected payoffs should be zero, which yields (1) and concludes the proof.

Appendix B
Proof of Theorem 2

Based on (13), the EDM without error can be written as

$$\begin{aligned} \boldsymbol{D} &= -2\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}} + \mathbf{1}\mathsf{diag}\left(\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}\right)^{\mathsf{T}} + \mathsf{diag}\left(\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}\right)\mathbf{1}^{\mathsf{T}} \\ &= -2\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}} + \mathbf{1}\left[0,\, \boldsymbol{\beta}_0^{\mathsf{T}}\right] + \left[0,\, \boldsymbol{\beta}_0^{\mathsf{T}}\right]^{\mathsf{T}}\mathbf{1}^{\mathsf{T}}, \end{aligned} \tag{24}$$

where $\boldsymbol{\beta}_0 \triangleq \left[d_{10}^2, ..., d_{M0}^2\right]^{\mathsf{T}}$, and

$$\boldsymbol{X} = \begin{bmatrix} 0 & x_1 & \cdots & x_M \\ 0 & y_1 & \cdots & y_M \end{bmatrix}^{\mathsf{T}} \triangleq [\mathbf{0},\, \bar{\boldsymbol{X}}^{\mathsf{T}}]^{\mathsf{T}}. \tag{25}$$

The attack is implemented on EDM and we can derive the resultant coordinate covariance, which is tampered by the attack, based on [30]:

$$\begin{aligned} &\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^{\mathsf{T}} \\ &= -\frac{1}{2}\left(\boldsymbol{D} + \boldsymbol{E} - \mathbf{1}[0,\, \boldsymbol{\beta}_0^{\mathsf{T}} + \boldsymbol{e}^{\mathsf{T}}] - [0,\, \boldsymbol{\beta}_0^{\mathsf{T}} + \boldsymbol{e}^{\mathsf{T}}]^{\mathsf{T}}\mathbf{1}^{\mathsf{T}}\right) \\ &= \boldsymbol{X}\boldsymbol{X}^{\mathsf{T}} + \frac{1}{2}\left(\begin{bmatrix} 0 & \mathbf{0}^{\mathsf{T}} \\ \mathbf{0} & \mathbf{1}\boldsymbol{e}^{\mathsf{T}} \end{bmatrix} + \begin{bmatrix} 0 & \mathbf{0}^{\mathsf{T}} \\ \mathbf{0} & \boldsymbol{e}\mathbf{1}^{\mathsf{T}} \end{bmatrix}\right) \\ &= \begin{bmatrix} 0 & \mathbf{0}^{\mathsf{T}} \\ \mathbf{0} & \bar{\boldsymbol{X}}\bar{\boldsymbol{X}}^{\mathsf{T}} + \frac{1}{2}\left(\mathbf{1}\boldsymbol{e}^{\mathsf{T}} + \boldsymbol{e}\mathbf{1}^{\mathsf{T}}\right) \end{bmatrix}. \end{aligned} \tag{26}$$

First, we notice that the eigenspace of $\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^{\mathsf{T}}$ in (26) is at most 4-dimensional (the other eigenvalues are zeros) by the rank inequality

$$\begin{aligned}\mathsf{rank}(\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^{\mathsf{T}}) &\leq \mathsf{rank}(\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}^{\mathsf{T}}) + \mathsf{rank}(\mathbf{1}\boldsymbol{e}^{\mathsf{T}}) + \mathsf{rank}(\boldsymbol{e}\mathbf{1}^{\mathsf{T}}) \\ &= 4. \end{aligned} \tag{27}$$

In addition, the eigenspace is spanned by

$$\mathsf{eigenspace}(\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^{\mathsf{T}}) = \mathsf{span}\{\boldsymbol{x}, \boldsymbol{y}, \mathbf{1}, \boldsymbol{e}\}, \tag{28}$$

where $\boldsymbol{x} \triangleq [x_1, ..., x_K]^{\mathsf{T}}$ and $\boldsymbol{y} \triangleq [y_1, ..., y_K]^{\mathsf{T}}$; this can be seen from the following equation.

$$\begin{aligned}&\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}^{\mathsf{T}} + \frac{1}{2}\left(\mathbf{1}\boldsymbol{e}^{\mathsf{T}} + \boldsymbol{e}\mathbf{1}^{\mathsf{T}}\right) \\ &= [\boldsymbol{x}, \boldsymbol{y}, \mathbf{1}, \boldsymbol{e}] \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}\boldsymbol{I} \end{bmatrix} [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{e}, \mathbf{1}]^{\mathsf{T}}. \end{aligned} \tag{29}$$

Note that we assume $\boldsymbol{E}$ is independent with $\bar{\boldsymbol{X}}$, and that the power of the error is ς^2, i.e.,

$$\left\| \frac{\mathbf{1}\boldsymbol{e}^{\mathsf{T}} + \boldsymbol{e}\mathbf{1}^{\mathsf{T}}}{2} \right\|_* = \varsigma^2 M. \tag{30}$$

Since the error is assumed to be independent with the coordinates and that i.i.d. Gaussian coordinate vectors are uniformly directed in space, a constant share of the error power is leaked out of the coordinate eigenspace spanned by $[\boldsymbol{x}, \boldsymbol{y}]$. Considering the objective of the faulty node is to minimize the leakage power beyond any 2D space, which in this case is equivalent to maximizing the power in the 2D space given that the total power is fixed, the best attack the faulty node can implement is to concentrate its error power to the same linear space spanned by $\mathbf{1}$, i.e.,

$$\boldsymbol{e}_{\text{opt}} = \varsigma^2 \mathbf{1}. \tag{31}$$

Note that the solution $\boldsymbol{e} = -\boldsymbol{e}_{\text{opt}}$ also satisfies the conditions, however, it results in a negative eigenvalue of $\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^{\mathsf{T}}$. This violates with [39, Theorem 2] which proves an important property of EDM that if $\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^{\mathsf{T}}$ is not positive-semi-definite, then there exists at least three distance measurements violating the triangle inequality. In other words, a shout attack may put the faulty node in a higher-dimensional space whereas a whisper attack would lead to violation of the triangle inequality which is much easier to spot.

Let us consider the minimization inside the expectation in (18), this is an optimal low-rank approximation problem whose solution is well known to be the dominant 2-dimensional singular space of $\hat{\boldsymbol{X}}$, i.e., the eigenspace of $\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^{\mathsf{T}}$. Denoting the singular value decomposition (SVD) of $\hat{\boldsymbol{X}}$ as $\hat{\boldsymbol{X}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^{\mathsf{T}}$ (singular values are always arranged in non-increasing order), then

$$\boldsymbol{Z}_{\text{opt}} = \hat{\boldsymbol{U}}_2\hat{\boldsymbol{\Sigma}}_2\hat{\boldsymbol{V}}_2^{\mathsf{T}}, \tag{32}$$

wherein $\hat{\boldsymbol{U}}_2$ and $\hat{\boldsymbol{V}}_2$ denote the first two columns of $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{U}}$, respectively, and $\hat{\boldsymbol{\Sigma}}_2$ contains the two dominant singular values. It follows that the minimum projection error is

$$\min_{\boldsymbol{Z} \in \mathbb{R}^{M \times M},\, \mathsf{rank}(\boldsymbol{Z}) = 2} \left\| \hat{\boldsymbol{X}} - \boldsymbol{Z} \right\|_2^2 = \left\| \hat{\boldsymbol{X}} - \boldsymbol{Z}_{\text{opt}} \right\|_2^2 = \sum_{i=3}^{M} \Sigma_{i,i}^2, \tag{33}$$

which is the coordinate power leakage beyond 2D space due to tampered EDM. To solve for this quantity, we adopt the Gram–Schmidt orthogonalization process on the set $\{\boldsymbol{x}, \boldsymbol{y}, |\varsigma|\mathbf{1}\}$ (given the optimal attack derived in (31)). Based on symmetry that $\boldsymbol{x}$ and $\boldsymbol{y}$ are both i.i.d. Gaussian distributed, it is clear that the direction of the third vector is not relevant and hence it is replaced by $\boldsymbol{w}$ where $\boldsymbol{w} \triangleq \sqrt{\varsigma^2 M}[1, ..., 0]^{\mathsf{T}}$ with the same power. For brevity, the detailed process is omitted as the orthogonal basis vectors and the expected leakage power $h(\varsigma^2)$ are given as below:

$$\begin{aligned} \boldsymbol{u}_1 &= \boldsymbol{x}; \\ \boldsymbol{u}_2 &= \boldsymbol{y} - \frac{\boldsymbol{u}_1^{\mathsf{T}}\boldsymbol{y}}{\|\boldsymbol{u}_1\|_2^2}\boldsymbol{u}_1; \\ \boldsymbol{u}_3 &= \boldsymbol{w} - \frac{\boldsymbol{u}_1^{\mathsf{T}}\boldsymbol{w}}{\|\boldsymbol{u}_1\|_2^2}\boldsymbol{u}_1 - \frac{\boldsymbol{u}_2^{\mathsf{T}}\boldsymbol{w}}{\|\boldsymbol{u}_2\|_2^2}\boldsymbol{u}_2; \\ h(\varsigma^2) &= \min_{i=\{1,2,3\}} \{\mathbb{E}[\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i]\} \\ &= \min_{i=\{1,2,3\}} \{M\sigma^2, (M-1)\sigma^2, (M-2)\varsigma^2\}, \end{aligned} \tag{34}$$

and the conclusion follows immediately.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2084–2123, thirdquarter 2016.

[3] W. Viriyasitavat and D. Hoonsopon, "Blockchain characteristics and consensus in modern business processes," *Journal of Industrial Information Integration*, 2018.

[4] W. Viryasitavat, L. Da Xu, Z. Bi, and A. Sapsomboon, "Blockchain-based business process management (bpm) framework for service composition in industry 4.0," *Journal of Intelligent Manufacturing*, May 2018.

[5] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending Bitcoin's proof of work via proof of stake," *SIGMETRICS Performance Evaluation Review*, vol. 42, no. 3, pp. 34–37, 2014.

[6] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*. New York, NY, USA: ACM, 2017, pp. 51–68.

[7] I. Eyal, A. E. Gencer, E. G. Sirer, and R. V. Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. Santa Clara, CA: USENIX Association, 2016, pp. 45–59.

[8] I. Abraham, D. Malkhi, K. Nayak, L. Ren, and A. Spiegelman, "Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus." *Leibniz International Proceedings in Informatics, LIPIcs*, vol. 95, 2018.

[9] C. Cachin, "Architecture of the Hyperledger blockchain fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.

[10] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *OSDI*. Berkeley, CA, USA: USENIX Association, 1999, pp. 173–186.

[11] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 USENIX Annual Technical Conference (USENIX ATC 14)*. Philadelphia, PA: USENIX Association, 2014, pp. 305–319.

[12] R. Guerraoui, N. Knežević, V. Quéma, and M. Vukolić, "The next 700 BFT protocols," in *Proceedings of the 5th European Conference on Computer Systems*, ser. EuroSys '10. New York, NY, USA: ACM, 2010, pp. 363–376.

[13] L. Lamport *et al.*, "Paxos made simple," *ACM Sigact News*, vol. 32, no. 4, pp. 18–25, 2001.

[14] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication," in *Open Problems in Network Security*, J. Camenisch and D. Kesdoğan, Eds. Cham: Springer International Publishing, 2016, pp. 112–125.

[15] S. Li, L. D. Xu, and S. Zhao, "5g internet of things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1 – 9, 2018.

[16] L. D. Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Trans. Ind. Inf.*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[17] Online; accessed 20-Mar-2018; https://neo.org/.

[18] FOAM, online; accessed 20-Sep-2018; https://foam.space/publicAssets/FOAM_Whitepaper.pdf.

[19] T. Dasu, Y. Kanza, and D. Srivastava, “Unchain your blockchain,” in *Proc. Symposium on Foundations and Applications of Blockchain*, vol. 1, 2018, pp. 16–23.

[20] J. Newsome, E. Shi, D. Song, and A. Perrig, “The Sybil attack in sensor networks: Analysis & defenses,” in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, ser. IPSN ’04. New York, NY, USA: ACM, 2004, pp. 259–268.

[21] D. B. Faria and D. R. Cheriton, “Detecting identity-based attacks in wireless networks using signalprints,” in *Proceedings of the 5th ACM Workshop on Wireless Security*. New York, NY, USA: ACM, 2006, pp. 43–52.

[22] M. Demirbas and Y. Song, “An RSSI-based scheme for sybil attack detection in wireless sensor networks,” in *Proceedings of the International Symposium on on World of Wireless, Mobile and Multimedia Networks*, Washington, DC, USA, 2006, pp. 564–570.

[23] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, “Applications of blockchains in the internet of things: A comprehensive survey,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1676–1717, Secondquarter 2019.

[24] C. Cachin and M. Vukolić, “Blockchain consensus protocols in the wild,” *arXiv preprint arXiv:1707.01873*, 2017.

[25] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, “Blockchain technologies for the internet of things: Research issues and challenges,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2188–2204, Apr 2019.

[26] J. R. Douceur, “The Sybil attack,” in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 251–260.

[27] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, “Locating the nodes: cooperative localization in wireless sensor networks,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul 2005.

[28] A. B. MacKenzie and S. B. Wicker, “Selfish users in Aloha: a game-theoretic approach,” in *IEEE Veh. Tech. Conf.*, vol. 3, 2001, pp. 1354–1357.

[29] A. Alvarez, G. Valera, M. Lobeira, R. Torres, and J. L. Garcia, “New channel impulse response model for UWB indoor system simulations,” in *IEEE Vehicular Technology Conference*, vol. 1, Apr 2003, pp. 1–5 vol.1.

[30] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, “Euclidean distance matrices: Essential theory, algorithms, and applications,” *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 12–30, Nov 2015.

[31] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proc. the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

[32] E. J. Candès *et al.*, “Compressive sampling,” in *Proc. international congress of mathematicians*, vol. 3. Madrid, Spain, 2006, pp. 1433–1452.

[33] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, “Vivaldi: A decentralized network coordinate system,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 15–26, Aug. 2004.

[34] D. Stolz and R. Wattenhofer, “Byzantine agreement with median validity,” in *International Conference on Principles of Distributed Systems (OPODIS)*, 2016, pp. 1–14.

[35] K. Daniel, E. Jakob, B. Michael, and B. Laura, “Recent development and applications of sumo - simulation of urban mobility.” *International Journal On Advances in Systems and Measurements*, vol. 5, pp. 128–138, Dec. 2012.

[36] P. Berkhin, *A Survey of Clustering Data Mining Techniques*. Springer Berlin Heidelberg, 2006, pp. 25–71.

[37] J. F. Nash, “Equilibrium points in n-person games,” *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[38] S. fen Cheng, D. M. Reeves, Y. Vorobeychik, and M. P. Wellman, “Notes on equilibria in symmetric games,” in *International Workshop On Game Theoretic And Decision Theoretic Agents (GTDT)*, 2004, pp. 71–78.

[39] J. C. Gower, “Euclidean distance geometry,” *Math. Sci*, vol. 7, no. 1, pp. 1–14, 1982.