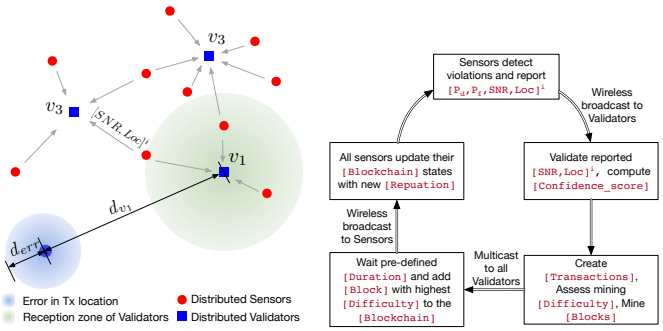
SenseChain: Blockchain based Reputation System for Distributed Spectrum Enforcement

***Abstract*—Distributed enforcement of spectrum policies require fusion of sensing results from a set of spatially scattered sensors to detect anomalous behavior with the highest possible accuracy. Central to this problem is the lack of trust or reputation of the participating sensors, which often leads to incorrect and biased inferences. In SenseChain, we leverage the distributed consensus mechanism employed in Blockchain networks to capture the reputation of the sensors, leading to a highly reliable and accurate enforcement system. Specifically, we define and analyze a detection mechanism to identify falsifying sensors using a distributed anomaly detection system and use the Blockchain to record the individual's behavior. The reputation is then based on the combination of the difficulty level of the consensus method and the degree of falsehood in the reported sensor values. We evaluate SenseChain using an integrated Blockchain and anomaly detection simulator to show that DLTs can be used to track reputation of distributed sensors for enforcement of spectrum policies.**

I. Introduction

Recent interest in applying Distributed Ledger Technology (DLT) like Blockchain, beyond cryptocurrencies [1], [2] has led to creative applications that maintain the integrity of transactions while assuring provenance of information being transacted on. Smart Contracts [3] are often seen as a viable way of deploying these transactional systems on a Blockchain. While such applications benefit from continued proliferation of DLT platforms (e.g. Ethereum, Hyperledger, Hashgraph, etc), these are fundamentally restricted to the features of such platforms that limit the innovation and scope for new applications. Interestingly, the core features of DLTs like immutability and distributed validation of transactions can be found in many applications that rely on data aggregation and dissemination among untrustworthy entities. Central to this, is the definition of *Transaction* and *Consensus*. While transactions are unique entries in an electronic ledger that encapsulate exchange of valuables (money, goods, data, etc.) between multiple parties in a cryptographically secure manner, consensus is responsible for all the constituents in a network to agree on one common version of the ledger without involving globally trusted intermediaries (e.g., trusted web servers or human arbiters like lawyers etc.).

We envision a connected world of Things where instead of any one universal Blockchain for a gamut of applications (Ethereum, Hyperledger, etc.), there are numerous independent islands of DLTs, employing proprietary definitions of transactions and consensus algorithms that are tailored for specific applications. In practice there are no technical barriers to implement such an idea, unless it needs to scale with a global footprint. We believe that many applications need not be scaled beyond a certain geographical area or a small interconnected



(a) Distributed anomaly detection of spectrum Sensors

(b) Reputation assignment and weighted fusion using Blockchain

Fig. 1: SenseChain: Distributed nodes can interchange roles as Sensor or Validator. Each Validator assesses false sensor reports, assigns confidence scores and creates transactions for each peer sensor. Transaction blocks are mined with Proof-of-Work difficulty proportional to number of sensors in a block.

network like a Smart-City, Smart-Home, Micro-grid and Mobile Adhoc Network (MANET) of sensors. SenseChain is such an example, where distributed sensors detecting violation of spectrum access policies (malicious or otherwise) can take advantage of Blockchain technology that is self-contained and operates autonomously, while benefiting from the core properties of DLT. Our work focuses on the post sensing phase, where the sensing results may be incorrectly reported with a malicious intent to disrupt the information fusion. Therefore, SenseChain also includes an anomaly detection system that separates the *good* actions from the *bad* and then assign a reputation metric to each sensor based on individual actions. This reputation can then be utilized in fusing the results of the sensors via a weighted function. The challenge in such reputation based systems is an implicit reliance on a separate trusted infrastructure to detect falsifying sensors and disseminate reputation metric. SenseChain eliminates such restriction by assigning the task of validation to the sensors itself and requiring to compute a Proof-of-Work to include their validation in the Blockchain. Thus, the Blockchain serves as a historical ledger of falsifying behavior since the genesis of the enforcement system that is immutable and is available to all the nodes in the network.

Figure 1a shows a distributed spectrum enforcement scenario, where certain nodes assume the role of Sensor (red circles) while others assume the role of Validator (blue squares). Each sensor broadcasts their sensing results for peer validation, which may include false reports from some of the sensors. Depending on the reception zone (based on transmit power of the sensors) each Validator is tasked with identifying false reports for different number of sensors as highlighted by

the green area in figure 1a. The first step is to estimate the approximate location of the transmitter location based on the reported sensor values. There are many examples of such methods in the literature using variety of methods like multilateration [4], centroid [5], clustering, etc [6]. All of these methods estimate the location of the transmitter with some degree of uncertainty that can be modeled as an error term, d_{err} in figure 1a, around the true location of the transmitter (which is unknown to the validators). Guided by the estimated position of the transmitter and the characteristics of d_{err}, Validators use the Log-distance path loss model to validate the reported SNR of the sensors using its own sensed SNR and distance to transmitter as a reference. This results in a annulus validation zone for each sensor within the broadcast zone. Consequently, if a sensors is outside the annulus, it is concluded as a *bad* action with high certainty, while a score is calculated proportional to the thickness of the annulus, if the sensor lies inside the annulus. In the context of this work, we define this validation step as *anomaly detection*.

The anomaly detection phase is followed by recording the confidence score in the Blockchain for provenance and calculation of historical reputation. Figure 1b shows the protocol that also includes the Blockchain based accumulation of this reputation metric. The confidence score for each sensor along with the sensing values constitute a transaction and a block is mined by the Validator for all the sensors it validates. In `SenseChain` the role of Blockchain is two fold: 1) Provide an immutable record of anomalous behavior by the sensors and 2) By requiring the difficulty for the mining to be proportional to the number of sensors being validated, it acts as a measure of credibility of the validators as well. The mined blocks are multicast within the Validators to converge on the block that is mined with the highest difficulty as it is deemed as the most credible validation for the set of sensors included in that block. We define this as the *Most-Difficult-Chain* consensus. Once each Validator gets an updated Blockchain state, it broadcasts that to its validated sensors to update their local states as well. In that way, in the following round if any sensor chooses to assume the role of a validator it will always have historical confidence scores that is used to calculate the most current reputation of the sensor. Therefore, we make the following contributions:

1) Design and analysis of a fully distributed, peer-based anomaly detection algorithm to assess the degree of falsification by Sensors using a confidence score by the Validators (Section III).
2) Design and analysis of a novel Blockchain design to record the confidence scores where the Difficulty of mining is proportional to the number of the sensors being validated. This provides a measure of credibility of validation (Section IV).
3) Design and analysis of network protocol to disseminate Blockchain and achieve consensus based on the *Most-Difficult-Chain* rule (Section IV-B).
4) Employ a non-linear function to aggregate historical confidence scores and Difficulty in the block to compute a reputation score for each Sensor (Section V).
5) Practical evaluation using a novel simulator that combine the anomaly detection and the Blockchain based reputation management system (Section VI).

II. Models and Preliminaries

System Model: Let $\mathcal{S}=\{s_1,\ldots,s_N\}$ be the set of N sensors in the area. Let $\mathcal{V}=\{v_1,\ldots,v_M\}$ be the set of M validators in the area. The target that is being sensed is denoted by $\mathcal{T}$. The target may be a primary user, a rogue source or a transmitter to be localized. The sensors and validators are mobile entities with omnidirectional antennas and have a limited broadcast range. The validators have their own overlay network to communicate among themselves. Validators and sensors are mobile crowd devices. Devices with more processing capabilities assume the role of a validator. As such, typically there are much more crowd sensors than validators (i.e., $M \ll N$).
Sensing Model: Each sensor $s_i \in S$ senses the target $\mathcal{T}$ and may report the probability of detection (P_{d_i}), probability of false alarms (P_{f_i}), SNR and location. The validators use the SNR and location for anomaly detection and to assign reputations to the sensors. The reputations of sensors can be used by the validators to fuse the information of sensors for more reliable inference. The weighted aggregation of P_d and P_f as in [7], using the normalized reputations as weights would increase the credibility of detection. e.g., $P_d^{fused} = \sum_i w_i P_{d_i}$, where w_i is the reputation of sensor s_i normalized by the aggregate reputation of all the sensors. Similarly, the weighted localization as in [8], using the normalized reputations leads to more credible and accurate localization.
Blockchain Model: Each validator, v_j receives the sensing reports (referred to as 'transactions') from all the sensors within a finite range (referred to as a 'Validator Range') within a fixed duration (referred to as the 'sensing phase'). The validator detects whether the report is valid or an anomaly, evaluates and appends a confidence score to each transaction from all sensors within its range. These transactions are aggregated by the validator to create a candidate block with a certain difficulty (Proof of Work [9]). Each validator appends its candidate block to the blockchain which is then broadcasted to all the validators. The validators arrive at consensus on the most difficult blockchain. The blockchain is used to extract the reputation of each sensor. This work does not rely on a specific type of financial incentive mechanism, any approach that rewards validators for their effort (difficulty) [10] and sensors according to their reputation is valid.
Threat Model: *1) Malicious Sensors (Sensing threat):* The sensors may falsify either their reported SNR, their reported location or both. *2) Malicious Validators (Validation threat):* The validators may forge information in the block they create by falsifying the confidence scores. Since, the validators may assume the role of a sensor, whose reputation depends on the reputation-weighted fusion of reports from other sensors, the validators are intrinsically discouraged from falsifying.

Protocol: There are three key phases of activity in `SenseChain` that each of the entity in the system adheres to: 1) Sensing Phase: the sensors sense the target and report their findings to the validators in the vicinity, 2) Validation Phase: the validators assess the truthfulness of the reports and creates a block by aggregating the reports, and 3) Blockchain Phase: the validators broadcast mined blocks to arrive at consensuses on the Blockchain before calculating the reputation of sensors.

In the sensing phase, each sensor $s_i \in \mathcal{S}$ senses the target $\mathcal{T}$, and creates a report with its perceived signal-to-noise ratio (SNR), SNR^i and an estimate of its location, Loc^i and broadcasts the report, $[SNR, Loc]^i$ to all the validators. Additionally, the sensors may report other sensing parameters depending on the application of interest. However, for applications in wireless communications, the SNR and the location are identified as fundamental sensing parameters that all sensors must report. A sketch of the protocol steps is shown in figure 1b. It is to be noted that there is not explicit synchrony in the network and all nodes will converge at the same Blockchain state eventually by employing the *most-difficult-chain* rule described in subsequent sections.

III. SenseChain: Anomaly Detection

The goal of anomaly detection is to gauge the truthfulness of a sensor in a distributed manner, using only the fundamental sensing information that is reported by the sensor. In the validation phase, each validator $v_j \in \mathcal{V}$ receives the reports from all the sensors located within the 'Validator Range' (limited to a distance R around the validator). Algorithm 1 describes the anomaly detection performed by each validator.

Algorithm 1: Anomaly Detection Algorithm

1 **Function** *AnomalyDetection(Map,* $[SNR, Loc]$*)*
2 $R = 100$; $d_0 = Diameter(Map)$;
3 n_{v_j} = count($[SNR, Loc]$);
4 $[Loc_{\mathcal{T}}, d_{err}]$=DistributedTargetLocalization($[SNR, Loc]$);
5 $d_{v_j} = EuclideanDistance(Loc_{\mathcal{T}}, Loc_{v_j})$;
6 **for** *i=1:*n_{v_j} **do**
7 $[d_{s_i}^{min}, d_{s_i}^{max}]$ = Estimate Annulus as in §III-A;
8 $\hat{d}_s = EuclideanDistance(Loc_{\mathcal{T}}, Loc^i)$;
9 **if** $(\hat{d}_s \geq d_s^{min}$ & $\hat{d}_s \leq d_s^{max})$ & $(d_s^{max} - d_s^{min} < R)$ **then**
10 $S_{s_i} = 1 - \frac{(d_{s_i}^{max} - d_{s_i}^{min})}{d_0}$
11 **else**
12 $S_{s_i} = 0$;
13 **end**
14 **end**
15 **return** S_{s_i} for all $s_i \in \mathcal{S}$
16 **end**

First, the validator $v_j \in \mathcal{V}$ estimates the region most likely to contain the target $\mathcal{T}$ using the reports from the sensors as in line 4. This is done using a multilateration based localization similar to [11]. The centroid of the estimated region is taken as the location estimate of the target, denoted by $Loc_{\mathcal{T}}$. The estimated location of the target may vary from the true location due to two major reasons: 1) potential falsification in the sensing reports from malicious sensors, and 2) inaccuracies in the path loss model, receiver heterogeneity and other noise sources. The effect of falsified sensing reports on the localization of the target can be mitigated by clustering the location estimates similar to [12]. To account for any errors in the estimated location, we model the error in the location estimate and the true location using the random variable d_{err}. This represents the circular region with a radius equal to d_{err} around the estimated location, that is likely to include the true location of the target. This is shown in figure 1a. The distance from the target to the validator v_j, denoted by d_{v_j} is estimated as the euclidean distance between the estimated location of the target, $Loc_{\mathcal{T}}$ and the location of the validator, Loc_{v_j} (line 5). Due to the uncertainty in the location of the target, the true distance from the target to the validator will be a value in the interval $[(d_{v_j} - d_{err}), (d_{v_j} + d_{err})]$.

Lines 6-14 detail the steps involved in the detection of anomalies for each sensor $s_i \in \mathcal{S}$. The core of the anomaly detection algorithm involves two steps: 1. based on the SNR reported by sensor s_i, the validator determines an *annular region* which should contain the sensor, and 2. if the sensor's reported location lies outside the estimated annulus, it is considered an anomaly.

A. Estimation of the annulus validation zone

Using the reported SNR, SNR^i the validator evaluates the received power, (P_{r,s_i}) experienced by the signal transmitted from the target at each sensor s_i. The average noise floor (NF) for a short range communication, similar to 802.11a/g, is approximately $-96dBm$ [13]. Thus, (P_{r,s_i}) is given by,

$$P_{r,s_i}(dBm) = SNR^i(dB) + NF(-96dBm) \tag{1}$$

The validator estimates the distance to the target using the Log-distance path loss model [14],

$$PL_{s_i} = PL_{v_j} + 10\gamma \log_{10} \frac{d_{s_i}}{d_{v_j}} + \chi \tag{2}$$

where PL_{s_i} and PL_{v_j} is the path loss experienced by the transmission from the target at sensor s_i and validator v_j respectively, and d_{s_i} is the true distance to the sensor from the target. γ is the path loss exponent and χ is a zero-mean gaussian random variable to account for the shadowing effect. Since the path loss is equal to the difference between the transmitted and received powers, and the transmitted power (of the target) remains constant during the sensing phase, we can rewrite (2) in terms of the received power as,

$$-P_{r,s_i} = -P_{r,v_j} + 10\gamma \log_{10} \frac{d_{s_i}}{d_{v_j}} + \chi \tag{3}$$

where P_{r,v_j} is the received power from the signal transmitted from the target at the validator v_j. Note, that since the estimate of d_{v_j} is erroneous, the estimate of d_{s_i} from (3) will lie within a range, $[d_{s_i}^{min}, d_{s_i}^{max}]$. The distance from the target to each sensor s_i, denoted by $\hat{d_{s_i}}$ is estimated as the euclidean distance between the estimated location of the target, $Loc_{\mathcal{T}}$ and the reported location of the sensor, Loc^i (line 8).

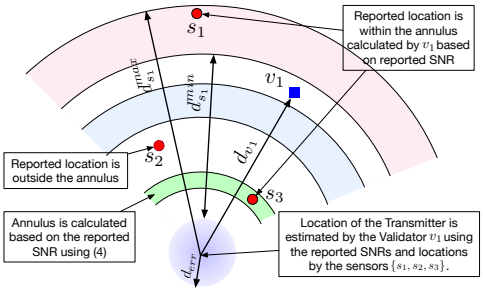


Fig. 2: Anomaly detection: When reported sensor location is outside the annulus it is detected as an anomaly else it is associated with a confidence score to represent its truthfulness.

The annular region estimated for sensor s_i by validator v_j is centered at the location of the target, $Loc_{\mathcal{T}}$ and defined by the inner and outer radii, $d_{s_i}^{min}$ and $d_{s_i}^{max}$ respectively. $d_{s_i}^{min}$ and $d_{s_i}^{max}$ is calculated from (3) as,

$$d_{s_i}^{min} = (d_{v_j} - d_{err}) \times 10^{\left(\frac{P_{r,v_j} - P_{r,s_i} - X_g}{10\gamma}\right)}$$
$$d_{s_i}^{max} = (d_{v_j} + d_{err}) \times 10^{\left(\frac{P_{r,v_j} - P_{r,s_i} - X_g}{10\gamma}\right)} \quad (4)$$

The thickness of the annulus is given by $(d_{s_i}^{max} - d_{s_i}^{min})$. By considering the minimum and maximum values of d_{v_j} in (4), we account for the impact on the annulus, by the error in the estimated and true location of the target. i.e., when the target is located anywhere within the circle as shown in figure 1a, the distance from the target to the sensor is bounded in the interval $[d_{s_i}^{min}, d_{s_i}^{max}]$. This is because $d_{s_i}^{min}, d_{s_i}^{max}$ represent the distance to the sensor from the closest and furthest possible location of the target respectively. The annular regions estimated by a validator v_1 for several sensors is shown in figure 2.

B. Anomalies and confidence score

Any falsification in the reported $[SNR, Loc]^i$ can be identified in two steps:

- If the validator received a report from a sensor s_i, whose reported location, Loc^i is outside the range of the validator, it is flagged as an anomaly. i.e., if $(d_{s_i} - d_{v_j}) > R$, then sensor s_i must have falsified.
- If the reported location of the sensor does not exist in the estimated annulus computed by the validator based on the reported SNR^i, then it is also flagged as an anomaly. i.e., if $\hat{d_{s_i}} < d_{s_i}^{min}$ or $\hat{d_{s_i}} > d_{s_i}^{max}$, then sensor s_i must have falsified. In figure 2, s_2 is detected as a falsifying sensor.

Note that, it is only important to detect falsifications in the sensor report, it is not required to identify the type of falsification (i.e., whether a sensor falsified in its reported SNR or location or both).

If $d_{s_i}^{min} \leq \hat{d_{s_i}} \leq d_{s_i}^{max}$, the sensor report may have been truthful with a confidence level. In figure 2 reports from s_1 and s_3 are not detected as anomalies. Since annulus estimated for s_3 is smaller, and the reported location of s_3 is within

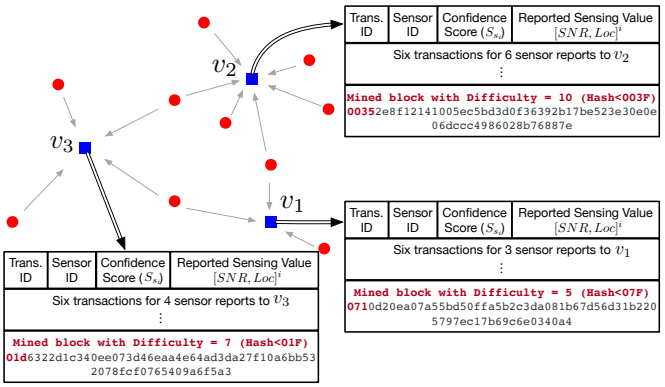


Fig. 3: The validators create a valid block by aggregating all transactions from sensors and calculating a hash less than the target based on the difficulty.

the annulus, s_3 must have reported the truth or falsified by an insignificant amount. Thus, s_3 is more likely to be truthful than s_1. The larger the thickness of the annulus, the higher uncertainty in the truthfulness of the sensor. Thus, the thickness of the annulus serves as a measure of the confidence on the truthfulness of the sensor. The confidence score of a sensor s_i, denoted by S_{s_i} is defined as, the normalized thickness of the annulus,

$$S_{s_i} = \begin{cases} 1 - \frac{(d_{s_i}^{max} - d_{s_i}^{min})}{d_0}, \text{if } (d_{s_i}^{min} \leq \hat{d_{s_i}} \leq d_{s_i}^{max}) \quad \& \\ \qquad\qquad\qquad\qquad (d_{s_i}^{max} - d_{s_i}^{min} < R) \\ 0 \qquad\qquad\qquad\qquad \text{, otherwise} \end{cases} \quad (5)$$

where, d_0 is a reference distance (the diameter of the region of interest) used for normalization. When a sensor falsifies information and it is flagged as an anomaly it would be assigned a confidence score of 0. When most of the sensors in the validator range are truthful, the thickness of the estimated annulus would be very small (since d_{err} is small), and the confidence in the truthfulness of the sensors increases (i.e., S_{s_i} is close to 1). If most of the sensors falsify, the thickness of the annulus is large (since d_{err} is large) and the uncertainity in the truthfulness increases. i.e., a falsifying sensor may not be detected as an anomaly. However, in this case, the score assigned to the falsifying sensor would be smaller than 1.

IV. SenseChain: Blockchain based reputation

Algorithm 2 describes the blockchain related functionality of each validator. For each sensing report, the validators prepare a transaction, by including a transaction id, a sensor id, the sensing report $[SNR, Loc]^i$ and the confidence score for that report, as shown in line 4 and figure 3. The transactions for all the sensors are aggregated (line 6) and are ready to be inserted in to a block. The process of creating a block is called *mining* and is outlined in lines 7 and 8. Figure 4 depicts the structure of the Blockchain and its key features. Each block is composed of a block header and a block body which contains the list of transactions from the sensors. The block header includes the hash of the block, the hash of the previous block, the difficulty target for that block, a nonce and a timestamp.

Algorithm 2: Blockchain based Reputation Algorithm

Function *Reputation*($S_{s_i} \forall s_i \in \mathcal{S}$, $[SNR, Loc]$, N, $Blockchain$)
  n_{v_j} = count($[SNR, Loc]$);
  for *i=1:*n_{v_j} **do**
    $transaction$ = $< transID, sensorID, [SNR, Loc]^i, S_{s_i} >$;
  end
  $transactions$ = Aggragate all transactions;
  Evaluate Difficulty of v_j, D_{v_j} from (6);
  $Block$ = CreateBlockwithDifficulty($transactions$,D_{v_j});
  $CandidateBlockchain$ = Add $Block$ to $Blockchain$;
  Broadcast $CandidateBlockchain$ to all validators;
  `// Consensus on Most-Difficult-Chain`
  Wait for Block-wait-time ($\tau_{\mathcal{B}}$) = 7s;
  Receive all Candidate Blockchains;
  $Blockchain$ = Select Most Difficult Blockchain;
  for *i=1:*N **do**
    Evaluate Reputation of s_i, $\mathcal{R}_{s_i}$ from (7);
  end
  return $\mathcal{R}$, $Blockchain$;
end

The merkle root field represents the hash value of the current block. Merkle tree hashing is commonly used in distributed systems and P2P networks for efficient data verification [15]. The nonce field is used for the proof-of-work algorithm, and it is the trial counter value that produced the hash with leading zeros. The difficulty target specifies the number of leading zero bits that the hash should contain to be considered valid. The implementation details are outlined in Section VI.

A. Difficulty of mining

When a new validator joins a network of validators, it gets a copy of the current blockchain. In addition to anomaly detection, validators also perform the functions of a blockchain miner [10]. Each validator generates a block $\mathcal{B}$ by aggregating the transactions, iterating over a nonce value and calculating the hash of a block with the nonce value included [9]. For the block $\mathcal{B}$ to be considered *valid*, a value of a hash function has to be less than a target T, i.e., $\mathrm{hash}(\mathcal{B}) < T$, where hash is a cryptographic hash function. The process of creating a valid block, typically requires a large amount of effort, which serves as a Proof-of-Work for the validators. The difficulty is a measure of how hard it is to find a hash below a given target T [16]. Unlike in many blockchain implementations we use a heterogeneous difficulty assignment mechanism, where each validator is assigned a different difficulty target. We define the difficulty of each validator $v_j \in \mathcal{V}$ as,

$$D_{v_j} = \left\lceil D_{max} \times \frac{n_{v_j}}{N} \right\rceil \qquad \forall v_j \in \mathcal{V} \tag{6}$$

where n_{v_j} is the number of sensors within the Validator Range R for validator v_j, and D_{max} is the maximum difficulty (designer's choice). The difficulty assumes an integer value in the interval $[1, D_{max}]$, and represents the number of leading zero bits in the target T. An example of valid blocks (with valid hashes) created by validators with different difficulty targets is shown in figure 3.

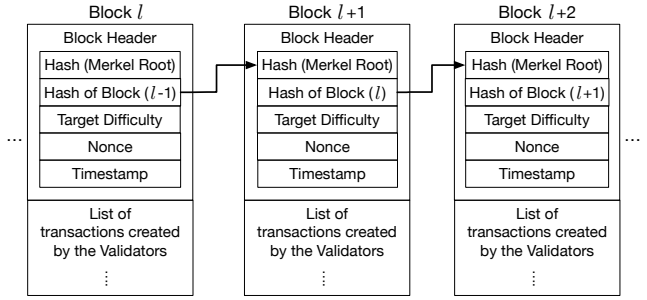


Fig. 4: Blockchain structure showing chained blocks with header fields and body. The target difficulty changes from block to block depending on the most difficult block that was successfully mined.

Blockchain Structure: The average time for a validator to create a valid block is directly proportional to the difficulty of the target and inversely poroportional to the average hash rate [17]. All the validators are assumed to have the same mining power (hash rate). Thus, a validator assigned with a higher difficulty target would on average, take a longer time to mine a block compared to a validator with a lower difficulty target. Validators are rewarded according to their effort (the difficulty). Each validator contends with all the other validators in the entire area in creating a valid block.

Lines 7-10 describe the validators' functions in creating a blockchain with a valid block. Each validator first determines it's mandated difficulty level using (6). Then the validator creates a valid block with the assigned difficulty level as discussed above, by incorporating the pool of transactions. Once a valid block is created it is added to the current blockchain and broadcasted. To provide sufficient time for all validators (with different difficulty levels) to generate a valid block, the validators wait for a fixed amount of time (referred to as the 'Block-wait time' denoted by $\tau_{\mathcal{B}}$) to receive all the broadcasted candidate blockchains. The block time is set by design to account for the block mining time, the propagation time of blocks to reach all validators, and for all validators to reach a consensus. Since, the propogation time is much less than the block mining time, the value of $\tau_{\mathcal{B}}$ is determined empirically, as the average time required to mine a block with a difficulty of D_{max}. All the validators in the network receive the candidate blockchains. The validators wait till the end of the 'Block-wait time', $\tau_{\mathcal{B}}$, and select the candidate blockchain with the most difficulty as the valid blockchain as discussed in the following section.

Choice of Maximum Difficulty (D_{max}): A higher maxmimum difficulty (D_{max}), sets a lower target value for the calculated hash [9] and determines the value of $\tau_{\mathcal{B}}$. For a lower D_{max} (a high target value), validators can generate a valid block faster, with low computation cost, so the delay in disseminating the information (reputation) is less. However, blockchain forks may occur more frequently and the security (immutability) of the blockchain itself is lowered. This is because the amount of computation required by validators to

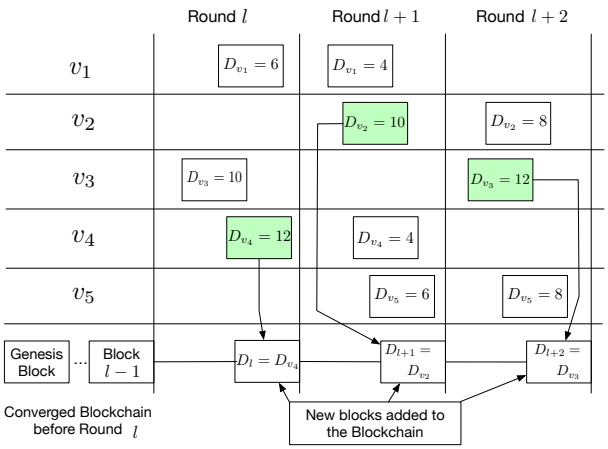


Fig. 5: Consensus using the *Most-Difficult-Chain* rule: At the end of each round, the most difficult block that is successfully mined by the validators is added to the blockchain.

generate a valid block is also less. Thus, there is a tradeoff between the computational power (or the delay in settlements) and the level of security.

B. Most-Difficult-Chain consensus

The validators select the candidate Blockchain with the highest difficulty, which is termed as the *Most-Difficult-Chain rule* (unlike the Longest-chain rule in Bitcoin [10]). In the event that multiple validators succeed in creating a block, the blockchain may fork. Even if a blockchain fork occurs, the blockchain would converge, because each validator selects the chain with the highest aggregate difficulty and generate a new block following the most difficult blockchain. Even though blocks are mined and may arrive at the validators at different times, the chain can be synchronized by exchanging status messages between the validators similar to [18].

The difficulty defined in (6) is proportional to the number of sensors within the range of a validator. Thus the most difficult block corresponds to the validator with the most number of sensors, which makes the most credible inference by harnessing the power of the crowd. Since the validator creates one transaction for each sensor within it's range, the difficulty of a block is proportional to the number of transactions in the block. Hence, the most difficult chain is the blockchain with the most number of transactions (scores and sensing data). The reputation of sensors' are assigned based on their transactions. Thus, the most difficult blockchain would contribute to the most credible assignment of reputation for sensors and would contribute to reliable inference using sensing data. Figure 5 shows an example of the consensus on the most-difficult blockchain among 5 validators. Before round l, a majority of the validators arrive at consensus on a blockchain. Each round involves a sensing, validation and a blockchain exchange phase. In round l, a valid block is created by validators v_1, v_3 and v_4. Note that validators v_2 and v_5 were unable to create a valid block within time $\tau_{\mathcal{B}}$. Each validator adds its own block to the blockchain and multicasts it to all other validators. Due to the various delays in mining and propogation time, the validator's may have different local views of the blockchain state. To guarantee the consensus properties and thus convergence to one canonical blockchain state, the `SenseChain` protocol relies on the assumption that the majority of the consensus validators follow the most-difficult chain. Thereby, in round l the blockchain with the highest aggregate difficulty, i.e., the blockchain from v_4 is agreed upon as the canonical blockchain state by the majority. Similarly, in rounds $l+1$ and $l+2$ the blockchains from v_2 and v_3 respectively, represent the canonical blockchain states.

V. Historical reputation and provenance

The reputation of a sensor s_i is calculated from the information stored in the blockchain. Let $l = 1, \ldots, L$ represent the L blocks in the blockchain. For a blockchain of length L, the reputation is defined by the non-linear sigmoid function [19], whose exponent is the weighted average of the difficulty of the block, D_l and the confidence scores of the sensor recorded in that block, $S_{s_i,l}$ (from the genesis block) and is defined as,

$$\mathcal{R}_{s_i} = \frac{1}{1+e^{-exp_{s_i}}} \tag{7}$$

$$\text{where, } exp_{s_i} = \frac{\sum_{l=1}^{L} a_{i,l} D_l S_{s_i,l}}{L.D_{max} \sum_{l=1}^{L} a_{i,l}}$$

where $a_{i,l}$ represents the association of sensor s_i with block l. $a_{i,l}$=1 if block l contains information about s_i, and $a_{i,l}$=0 otherwise. Note that the reputation of each sensor is calculated by the validator, by using the most-difficult-chain, scanning through all the blocks, extracting the the confidence scores and difficulties of each block. The exponent represents a historical average of the confidence in the sensors' reports and the difficulty with which the corresponding block was mined. Even though a malicious validator may forge information in its current block, since the reputation calculation relies on all the records from the genesis block and due to the immutability property of the blockchain, the impact on the calculated reputation will be very small as the blockchain grows.

The reputation $\mathcal{R}_{s_i}$ assumes a value in between 0 and 1. When a sensor continuously acts truthfully (or maliciously), the confidence scores recorded for that sensor in each block would be close to 1 (or 0), asymptotically driving the reputation R_{s_i} to a value of 1 (or 0) respectively. The nonlinear nature of the reputation function ensures that a sensor which engages vastly in either truthful or false behaviour would have a reputation of 1 or 0 respectively. Over time, this allows the validators to identify sensors that are always truthful or always malicious. Reputation of sensors that frequently alter their behaviour is subject to more pronounced variation. Thus, the reputation of a sensor serves as a measure of the credibility of it's reports. The validators use the reputation of sensors assimilated over time to perform weighted fusion of the sensing data in order to accurately detect and localize the target (see Section II).

VI. Evaluation and Results

A. Simulation Framework

`SenseChain` is evaluated on a mobile sensing and Blockchain simulator built on Matlab. The simulation parameters are shown in table I. We analyze the various facets of `SenseChain` using practical simulations.

1) Sensing Environment: We consider a random network topology with several Sensors, Validators and a single mobile target ($\mathcal{T}$). Random Waypoint mobility is chosen for movement of the various entities in the area of interest and each node is equipped with omnidirectional antenna. The target continuously transmits a signal at a fixed transmit power (40 mW). During the sensing phase, all the sensors receive the signal and compute their SNR from the received power (using (1) and NF = -96 dBm [13]). The sensors broadcast their reports ($[SNR, Loc]$) to the validators. Malicious behaviour of a sensor is emulated as a random variation (referred to as the *degree of falsification*) about the true SNR and true location of that sensor. Sensors exhibit malicious behaviour with varying probabilities and varying degrees of falsification. The validators receive reports only from sensors within the broadcast range (100 m). The diameter of the network, d_0 is 424 meters. The validators assign a confidence score to the sensing reports as in Algorithm 1.

2) Blockchain Simulator: The blockchain environment in this work is different from typical implementations in two ways: 1) Heterogeneous difficulty assignment: The difficulty varies for each validator and in each mined block in the blockchain. 2) Consensus: The validators arrive at consensus on the *Most-Difficult-Chain* to avoid forking. The simulation works as follows. For each sensing report, the validator creates a transaction by inserting the sensor id, the sensing report ($[SNR, Loc]$) and the confidence score. The sensor id is an integer index in the interval $[1, N]$, to represent the N sensors (e.g., sensor id of s_i is i). The transaction id is created by hashing transaction data through SHA-256 [20] twice, similar to typical blockchain implementations. The timestamps refers to the time at which the block was created, encoded as a Unix Epoch timestamp. The hash function used to generate the block id is SHA-256. The genesis block is created without any transactions by including a timestamp and creating a block with a hash corresponding to difficulty D_{max}.

At each round, each validator generates a block by aggregating the transactions, iterating over a nonce value (a random integer) and calculating the hash of the block. Each block is mined with a different difficulty (see Section IV-B). The block is considered *valid*, when its hash is less than a target T that depends on the difficulty. This is verified by checking whether the hash of the block has at least as many leading zero bits as T, for that validator. Once a valid block is created it is added to the blockchain and multicast to the Validators. Note that the size of the block (the number of transactions) is not fixed as it depends on the number of sensor being validated. All the validators receive the candidate blockchains within $\tau_{\mathcal{B}}$ time (7 seconds). Each validator calculates the total difficulty of each candidate blockchain, and selects the one with the highest aggregate difficulty. Thus, the validators arrive at consensus on the *Most-Difficult-Chain*. A total of 1000 blocks were mined for each analysis presented below. For each sensor $s_i \in \mathcal{S}$, the validators scans through each block in the blockchain to extract the entries corresponding to its sensor id i. The validators then compute the reputation of each sensor using (7).

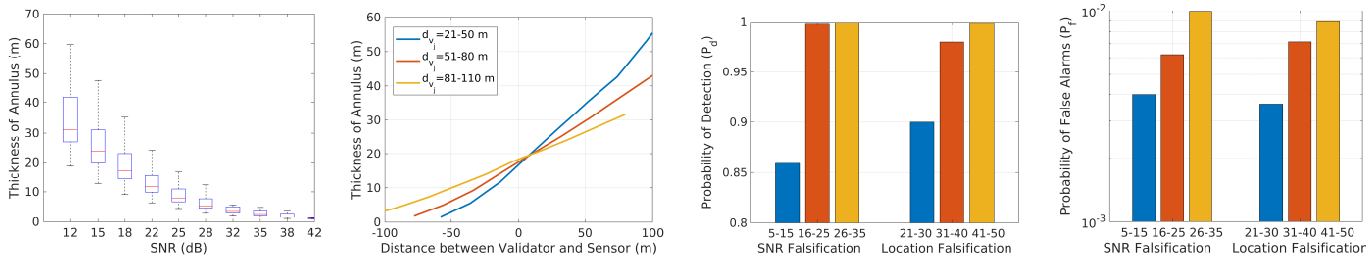
TABLE I: Simulation Parameters

Parameters	Value/Model
Area	300m × 300m
Node Distribution	Uniform Distribution
Mobility Model	Random Waypoint
Propagation Model	Log-distance propagation model [14]
Path-loss exponent (γ)	3 (urban area)
Carrier Frequency (f)	600 MHz
Number of Validators	5
Number of Sensors	20
Antenna Type	Omnidirectional
Broadcast Range	100
Maximum Difficulty (D_{max})	16
Block-wait Time ($\tau_{\mathcal{B}}$)	7 s
Target location error (d_{err})	Uniformly distributed in [20,30] m

B. Performance of anomaly detection

A falsifying sensor is detected as an anomaly, when its reported location lies outside the annulus (see Section III), else a confidence score is associated with the sensor report, to reflect the confidence in the truthfulness of the sensor. The smaller the thickness of the annulus the more confident the validator will be on the truthfulness of the sensor. The thickness of the annulus is defined in Section III-A and is a measure of the confidence score. Figure 6a shows the dependence of the thickness of the annulus on the SNR reported by the sensors. When the reported SNR is low, the annulus is wider, and the confidence in the truthfulness will be low. Since the thickness is high, a falsifying sensor may go undetected (as shown in figure 2), but would only be validated with a low confidence score. When the sensor reports a high SNR, the thickness of the annulus estimated by the validator is small. A sensor that falsifies by reporting a high SNR value is more likely to be detected as an anomaly since the the thickness of the annulus is very small (and the reported location is likely to lie outside the annulus). For a sensor that falsely reports a low SNR value, the thickness of the estimated annulus would be large. Hence, it is possible for the reported location to lie within the annulus. But the confidence score for such sensors would be very low due to the large thickness.

Figure 6b shows the dependence of the thickness of the annulus on the distance between the validators and the reported location of the sensor Loc^i. Consider two validators v_1 and v_2, with $d_{v_1} < d_{v_2}$. When the sensor is located closer to $\mathcal{T}$ than either validator, the thickness of the annulus estimated by v_1 would be less than v_2. That is v_1 would be able to,



(a) Variation of annulus width with reported SNR

(b) Annulus width with Sensor to Validator distance

(c) P_d with varying falsification in SNR (dB) and Location (m)

(d) P_f with varying falsification in SNR (dB) and Location (m)

Fig. 6: Performance of Anomaly Detection. (a) and (b) show the dependence of the thickness of the annulus on the SNR of the sensor and the distances from the target to the validator and the sensor. (c) and (d) show that when the degree of falsification is high, a validator is more likely to detect an anomaly, however the false alarms are also high.

more accurately assess the truthfulness of the sensor. When the sensor is located further away from $\mathcal{T}$ than either validator, the thickness of the annulus estimated by v_2 would be less than v_1. In this case v_2 would be able to, more accurately assess the truthfulness of the sensor.

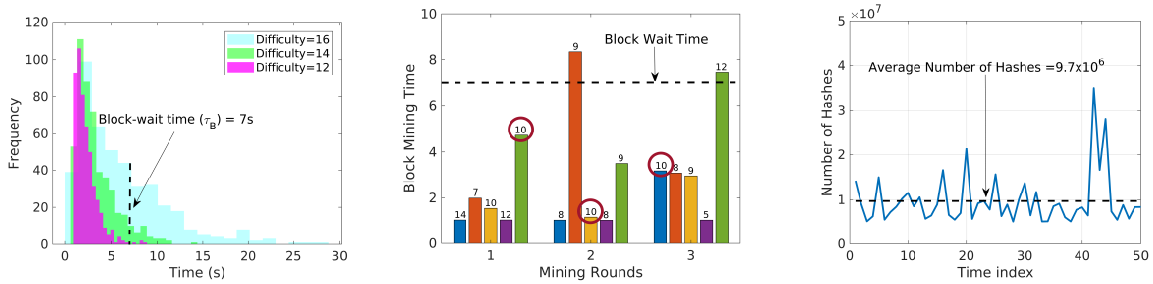
The performance of the anomaly detector with varying degree of falsification is shown in figures 6c and 6d. Figure 6c shows the probability of detection (P_d) and figure 6d shows the probability of false alarms (P_f) in detecting anomalies in the sensing reports. Recall that the annulus for a sensor is estimated using the reported SNR. If the sensor reports its true location and SNR, it will exist within the annulus. First, consider the effect on P_d and P_f with the degree of falsification in the reported SNR (i.e., the difference in the reported SNR and the true SNR) but reported location is true. When the degree of falsification in SNR increases, the reported location of the sensor is more likely to be outside the annulus. Hence, P_d increases with the degree of falsification. Even for low degrees of falsification in SNR (5-15 dB), P_d is relatively large (≈ 0.86). However, when a sensor falsifies to a higher degree, the possibility of flagging truthful sensors as anomalies increases. i.e, P_f also increases. Since the falsifying sensor is included in the distributed localization of $\mathcal{T}$ (as explained in §III), a higher degree of falsification leads to a higher possibility of error in the location of $\mathcal{T}$. Consequently, this leads to errors in the estimated annulus and truthful sensors may be detected as anomalies. However, P_f is very low (less than 10^{-2}) even for higher degrees of falsification in SNR.

Consider the effect on P_d and P_f by degree of falsification in the reported location while the reported SNR is true. When the degree of falsification in location increases, the more likely is the reported location of the sensor to be outside the annulus, and it is more likely to be detected as an anomaly. Hence, P_d increases with degree of falsification. Since the degree of falsification affects the distributed localization of $\mathcal{T}$, the possibility of flagging truthful sensors as anomalies also increases. i.e, P_f also increases. Overall, we see that Algorithm 1, achieves high probability of detection (≥ 0.86) even for low degrees of falsification and a low probability of false alarms (≤ 0.01) even for high degrees of falsification.

C. Performance of Blockchain based reputation

Blockchain performance: The performance of mining is shown in figure 7. Figure 7a shows the variation in the block mining time with varying difficulty of validators. The dotted line shows $\tau_{\mathcal{B}}$, i.e., the block-wait time which is equal to the average block time to mine a block with maximum difficulty ($D_{max} = 16$). When the difficulty level is high, the average time required to mine a block is more, since more amount of hashes are required on average, to find a hash value less than the target. The validators with a less difficulty target have a higher probability of mining a block within $\tau_{\mathcal{B}}$. The probability that a block is mined by a validator v_j within $\tau_{\mathcal{B}}$, when $D_{v_j} = 12$, $D_{v_j} = 14$ and $D_{v_j} = 16$, is 92%, 78% and 50% respectively. Even though validators with a lower difficulty have a higher probability of mining a block within $\tau_{\mathcal{B}}$, only the most-difficult block mined within $\tau_{\mathcal{B}}$ is added to the chain. This gives all the validators a chance to contribute to the blockchain and get rewarded.

The average time required to mine a block is proportional to the difficulty level and inversely proportional to the mining power of the validators [17]. Figure 7b shows the amount of time required by each validator to generate a valid block in each mining round. The block added to the blockchain at each mining round is determined by the most difficult block mined within the wait time of $\tau_{\mathcal{B}} = 7s$. The validators contend with all other validators in the area. As shown in the figure, in the first mining round v_5 is assigned the highest difficulty ($D_{v_5} = 10$) and generates a valid block within $\tau_{\mathcal{B}}$. Thus, the block from v_5 leads to the most difficult blockchain (as detailed in §IV-B), and is agreed upon as the canonical blockchain. In the second mining round, a block is mined with the highest difficulty by v_3 ($D_{v_3} = 13$) and is added to the blockchain. In the third mining round, even though v_2 has a higher difficulty ($D_{v_2} = 13$) it is unable to mine a block within $\tau_{\mathcal{B}}$. The block mined within $\tau_{\mathcal{B}}$ with the highest difficulty is from v_1 and it is added to the blockchain. Note that in the second and third mining rounds, v_2 and v_5 respectively are unable to create a valid block within $\tau_{\mathcal{B}}$. Figure 7c shows the number of hashes generated by the winning validators (whose blocks are added to the blockchain) in each mining round. This serves as a

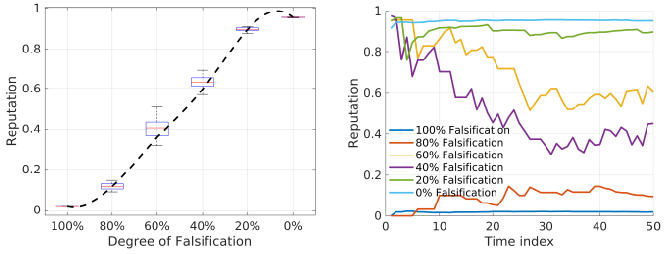


(a) Block mining times of validators with varying difficulty targets

(b) Block mining time per validator and winning block in each round

(c) The number of hashes generated by the winning validator

Fig. 7: Impact of the difficulty of mining on the block mining time and the winning block. In (b) each color bar represents the blocks mined by each validator in order from v_1 to v_5. The numbers on the top of the color bar indicates the difficulty with which the block was mined. The winning block in each mining round are annotated by the red circles.



(a) Reputation with degree of falsification by the Sensors

(b) Reputation of falsifying Sensors over time

Fig. 8: Reputation assignment with varying degree of malicious activity over time

measure of the amount of computation performed, time spent and power consumed by the validators in each round. Even though there are occasional spikes in the computational power (high number of hashes), most of the time the computational power is consistent. On average about 9.7 million hashes are calculated in each mining round.

Reputation Assignment: Figure 8 shows the reputation assigned to nodes, with varying degrees of malicious activity. Figure 8a shows the nonlinearity of the reputation function. Sensors that predominantly exhibit good behaviour ($falsification$<10%) by truthfully reporting sensing information asymptotically accumulate a reputation of 1. Sensors that continuously falsifies sensing information ($falsification$>90%) will accumulate a reputation close to 0. The reputation of sensors that arbitrarily alter their (truthful or malicious) behaviour by falsifying reports with a certain probability, are more susceptible to change based on their relatively dominant behaviour. The reputation of sensors which exhibit $falsification$ between 20% to 80% fall within the linear range of the sigmoid function. The reputation of these sensors are more likely to change depending on their dominant (truthful or malicious) behaviour.

The variation in the reputation of sensors over time is shown in figure 8b. Sensors create malicious reports with a probability equal to their percentage of falsification. Over time, the reputation values of sensors that exhibit consistently, either truthful or malicious behaviour settle much quicker, compared to sensors that exhibit alternating behaviour. It is clear that after about 30 blocks the reputation of nodes settle to within 10% of their steady state reputation. It is important to note that even in the presence of a malicious validator the impact on the reputation is minimal. This is because, the reputation of any node is assimilated from the entire blockchain. Even if a malicious validator creates valid blocks with forged information, the impact on the reputation by these forged blocks, decreases significantly with the number of validators and the length of the blockchain.

VII. Related Work

We categorize the related literature into two groups: **Anomolous behaviour Detection:** Trust and reputation based models for malicious sensor identification have been widely studied in the context of wireless sensor networks [21]. [22] uses a neighbor weight trust algorithm, in the problem of malicious node detection. [23] proposed a new trust management scheme based on D-S (Dempster-Shafer) evidence theory, by considering the spatio-temporal correlation of data collected by neighbouring sensors. These models rely on local inferences from neighbours, which needs to be disseminated throughout the network of trustless entities. In our work we achieve distributed consensus among nodes by sharing information on a blockchain. [24] proposed a malicious node recognition model to resist malicious behavior of high-reputation nodes in existing WSNs. [25] proposes an abnormal sensor identification using the pairwise similarity of sensing results of helpers. Trust has been investigated in the context of crowdsensing and collaborative spectrum sensing [7], [26]. Most of these approaches rely on centralized fusion of information, which is both vulnerable and does not scale well. In contrast, we propose anomaly detection in a purely distributed manner using only the SNR and the location of sensors, and the dissemination of information using the blockchain to assign reputation of sensors.

Blockchains for sensor networks: DLTs like blockchain have gained immense interest in various application domains in wireless sensor networks [2]. Blockchains have been em-

ployed for dynamic spectrum access [27] and to achieve secure routing among malicious nodes [28]. Blockchains have been used to establish a trust model and for the detection of malicious nodes in [21]. [29] proposed a smart contract based framework to solve the problems of trusted access control and distributed in the IoT. [30] addresses the problem of distributing trust and reputation among trustless nodes, by employing collaboration among miners. [31] proposes spectrum sensing as a service using a smart contract to describe the sensing service parameters and helpers are rewarded only if they perform sensing accurately. In contrast to these approaches we employ peer-based anomaly detection algorithm and a heterogeneous difficulty assignment and *Most-Difficult-Chain* rule to diversify the efforts and rewards of miners.

VIII. Conclusion

In this paper, we proposed an anomaly detection and reputation assignment scheme called `SenseChain`, based on the reputation information disseminated via a blockchain. Through simulation and analysis we draw the following conclusions: 1) anomalies in sensing reports can be detected with high accuracy in a distributed manner, 2) the *Most-Difficult-Chain* rule enables distributed consensus among spatially distributed nodes, 3) the non-linear function to aggregate historical confidence scores and corresponding Difficulty, enables the reputation assignment based on a sensors' degree of truthful (or malicious) behaviour. Thus, the distributed anomaly detection by validators and the use of the *Most-Difficult-Chain* to capture and disseminate the behaviour of sensors, provides a fast and tamper-proof means to arrive at distributed consensus on the reputation of sensors, among trustless entities.

References

[1] D. Wood, "Ethereum: A secure decentralised generalised transaction ledger," 2014.

[2] M. A. Khan and K. Salah, "Iot security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395 – 411, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17315765

[3] V. Buterin, "A next generation smart contract & decentralized application platform," 2015.

[4] R. Zekavat and R. M. Buehrer, *Handbook of Position Location: Theory, Practice and Advances*, 1st ed. Wiley-IEEE Press, 2011.

[5] J. Wang, P. Urriza, Y. Han, and D. Cabric, "Weighted centroid localization algorithm: Theoretical analysis and distributed implementation," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3403–3413, October 2011.

[6] J. Yang, Y. Chen, V. B. Lawrence, and V. Swaminathan, "Robust wireless localization to attacks on access points," in *2009 IEEE Sarnoff Symposium*, March 2009, pp. 1–5.

[7] A. Dutta and M. Chiang, "see something, say something crowdsourced enforcement of spectrum policies," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 67–80, Jan 2016.

[8] L. Altoaimy, I. Mahgoub, and M. Rathod, "Weighted localization in vehicular ad hoc networks using vehicle-to-vehicle communication," in *2014 Global Information Infrastructure and Networking Symposium (GIIS)*, Sep. 2014, pp. 1–5.

[9] D. Meshkov, A. Chepurnoy, and M. Jansen, "Revisiting difficulty control for blockchain systems," *IACR Cryptology ePrint Archive*, vol. 2017, p. 731, 2017.

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf."

[11] M. A. A. Careem, A. Dutta, and W. Wang, "Multi-agent planning with cardinality: Towards autonomous enforcement of spectrum policies," 10 2018, pp. 1–10.

[12] S. Chawla and A. Gionis, "k-means-: A unified approach to clustering and outlier detection," in *SDM*, 2013.

[13] "IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, March 2012.

[14] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[15] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17315765

[16] I. Eyal, "The miner's dilemma," in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 89–103.

[17] 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)*, June 2014, pp. 280–285.

[18] Ethereum, "Ethereum wire protocol (eth)," cited July 05. [Online]. Available: https://github.com/ethereum/devp2p/blob/master/caps/eth.md

[19] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, p. 311, Nov 2018. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2017.12.012

[20] U. D. of Commerce, N. I. of Standards, and Technology, *Secure Hash Standard - SHS: Federal Information Processing Standards Publication 180-4*. USA: CreateSpace Independent Publishing Platform, 2012.

[21] W. She, Q. Liu, Z. Tian, J. Chen, B. Wang, and W. Liu, "Blockchain trust model for malicious node detection in wireless sensor networks," *IEEE Access*, vol. 7, pp. 38 947–38 956, 2019.

[22] F. Zawaideh, M. Salamah, and H. Al-Bahadili, "A fair trust-based malicious node detection and isolation scheme for wsns," in *2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes Systems (IT-DREPS)*, Dec 2017, pp. 1–6.

[23] W. Zhang, S. Zhu, J. Tang, and N. Xiong, "A novel trust management scheme based on dempster—shafer evidence theory for malicious nodes detection in wireless sensor networks," *J. Supercomput.*, vol. 74, no. 4, pp. 1779–1801, Apr. 2018. [Online]. Available: https://doi.org/10.1007/s11227-017-2150-3

[24] G. Yin, G. Yang, Y. Wu, X. Yu, and D. Zuo, "A novel reputation model for malicious node detection in wireless sensor network," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, Oct 2008, pp. 1–4.

[25] H. Li and Z. Han, "Catch me if you can: An abnormality detection approach for collaborative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3554–3565, November 2010.

[26] S. Jana, K. Zeng, W. Cheng, and P. Mohapatra, "Trusted collaborative spectrum sensing for mobile cognitive radio networks," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1497–1507, Sep. 2013.

[27] K. Kotobi and S. G. Bilen, "Secure blockchains for dynamic spectrum access: A decentralized database in moving cognitive radio networks enhances security and user access," *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 32–39, March 2018.

[28] J. Yang, S. He, Y. Xu, L. Chen, and J. Ren, "A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks," *Sensors*, vol. 19, no. 4, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/4/970

[29] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, April 2019.

[30] S. Goka and H. Shigeno, "Distributed management system for trust and reward in mobile ad hoc networks," in *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2018, pp. 1–6.

[31] S. Bayhan, A. Zubow, and A. Wolisz, "Spass: Spectrum sensing as a service via smart contracts," in *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Oct 2018, pp. 1–10.