Camera-ready version. To be published and presented at IEEE GLOBECOM 2015.

Robust Consensus-based Cooperative Spectrum Sensing under Insistent Spectrum Sensing Data Falsification Attacks

Aida Vosoughi and Joseph R. Cavallaro
Department of Electrical and Computer Engineering
Rice University, Houston, Texas, USA

Alan Marshall
Department of Electrical Engineering and Electronics
University of Liverpool, Liverpool, UK

***Abstract*—In this paper, we introduce Insistent Spectrum Sensing Data Falsification (ISSDF) as a new practical and destructive attack model aimed at distributed cooperative spectrum sensing schemes that are based on iterative average consensus. We compare various linear iteration-based and iterative gossip-based schemes in terms of primary user detection performance and convergence speed under this attack. Moreover, we devise a trust management scheme to mitigate the attack and we propose a practical trust-aware consensus-based scheme for distributed cooperative spectrum sensing which is resilient to ISSDF. Finally, we quantify the performance improvement due to trust management through extensive simulations.**

I. Introduction

Dynamic spectrum access has been introduced to solve the radio frequency spectrum shortage problem which is caused by fixed assignment of the bands to primary users (licensees of the bands or PU's). Dynamic use of spectrum facilitates more flexibility by allowing secondary users (SU's) to use licensed spectrum bands on an opportunistic non-interference basis. That is when a PU is inactive, an SU can use that channel on the condition of leaving the channel whenever the PU returns.

A well-known approach for detecting the activity of PU is *cooperative spectrum sensing* where a set of SU's cooperate by communicating their sensing information with each other and collectively deciding on the presence or absence of the PU [1]. In a centralized network, the final decision is made by a fusion center that aggregates the information from all of the SU's in the network. In contrast, in a decentralized network (e.g. a cognitive radio ad hoc network or CRAHN), the cooperation must be done in a distributed manner. Distributed average consensus algorithms [2] [3] can be used to perform distributed cooperative spectrum sensing (DCSS). These algorithms are based on iterative diffusion and aggregation of data through neighbor nodes and are known to converge fast when all of the nodes in the network follow the predefined protocol [4]; however, if some of the nodes behave maliciously by broadcasting falsified values (Spectrum Sensing Data Falsification Attack or SSDF [5]) or by not following the consensus strategy properly, then these schemes can be compromised significantly. These attacks lead to serious problems in DCSS (See discussion in Section V). As a result, it is necessary to design a low-cost and practical distributed scheme for robust detection of PU activities in a potentially hostile environment. To this end, this paper presents the following contributions:

- We introduce a new practical and extremely destructive attack model against iterative consensus-based DCSS schemes: Insistent Spectrum Sensing Data Falsification (ISSDF), where malicious nodes repeatedly broadcast falsified sensing information in all of the consensus iterations to maximize their effect.
- As a step towards mitigating ISSDF, we propose a trust management scheme that greatly improves the detection performance under this attack.
- We compare and analyze a variety of average-consensus-based algorithms for DCSS both in an honest network and under ISSDF attack. To the best of our knowledge, this paper is the first to present such comparison. Based on our comparisons through extensive simulations, we draw conclusions about the most cost-effective algorithm for DCSS: a practical trust-aware scheme based on equally-weighted linear iterations.

II. Background and Related Work

Distributed consensus algorithms have been studied in applications such as coordination in multi-agent systems, sensor networks, wireless ad hoc networks, and peer-to-peer distributed systems among others [6] [4]. For example, the gossip-based Push-Sum protocol [2] is proposed for average consensus in P2P and wireless ad hoc networks; also, linear iteration-based scheme [3] has been proposed for average consensus in sensor fusion. Recently, consensus-based DCSS schemes have been introduced which utilize linear iterations [7] and gossip-based strategies [8] [9]. However, these update rules are not resilient against malicious nodes that send incorrect data to the other nodes. Sundaram et. al. [10] show that for distributed function calculation using linear iterations, the network graph connectivity is a key factor in resilience to malicious (faulty) nodes. In their attack model, malicious nodes do not follow the linear iterative strategy and instead arbitrarily update their values in each iteration; however, the malicious nodes do not falsify their own initial values to affect the consensus. Therefore their attack model is essentially of different nature from falsifying attacks such as SSDF. Nevertheless, Sundaram et. al. prove that it is possible for a set of conspiring malicious nodes to prevent the network to converge to the correct answer [10]. A trust scheme using trust propagation in the presence of a set of pre-trusted nodes has been used to mitigate the effect of Byzantine adversaries in linear iterative consensus in sensor networks [11].

In the context of SSDF attacks in DCSS applications, trust has been deployed for mitigating the attack in non-consensus schemes [5] [12]; however, the problem of coping with SSDF

is hardly explored in consensus-based DCSS schemes. In a previous work, we proposed a trust model for consensus DCSS based on single neighbor gossip [9] that proved to be very effective in mitigating SSDF attacks. In this paper, we study and compare a variety of iterative average consensus schemes that are based on broadcast model, which is more realistic due to the nature of wireless networks, and we integrate our proposed trust management scheme to show the significant improvement in PU detection performance in a decentralized network of SU's. Unlike [11], our trust model, does not require pre-trusted nodes and is based on simple local trust evaluations which makes it a practical and low-cost method for CRAHN's. Also, in our network model, we do not make any assumption on the graph connectivity or node density. Moreover, for the first time we evaluate consensus-based DCSS schemes (with or without integrated trust) against ISSDF attacks where malicious SU's repeatedly broadcast falsified data and do not perform consensus iteration updates.

III. System Model

We consider a location area within the range of a single PU transmitting signal with power P_t. A set of N SU nodes are uniformly spread in the location area and move according to a random way point model [13]. The SU's that are located within the communication range of SU node $i \in \{1, ..., N\}$ are called node i's one-hop neighbors and are denoted by the set N_i. When node i broadcasts a value, all of its neighbors will receive that value. We assume perfect communication between the SU's via a common control channel. Note that since the nodes are mobile, the neighbor sets are always changing.

The SU's have power detectors and detection is modeled as a binary hypothesis testing problem: H_0 if PU is absent and H_1 if PU is present. Hata (suburban areas variant) [14] is used as the path loss model for PU transmission. The Hata model has been recommended by the IEEE 802.22 working group for spectrum sensing modeling in WRAN operating in TV whitespace [5]. For small scale path loss variability, log-normal shadowing model is applied. The overall path loss (in dB) is given by $PL(d) = \overline{PL}(d) + X_\sigma \quad [dB]$, where $\overline{PL}(d)$ is the average path loss as a function of transmitter-receiver distance d based on the Hata model, and X_σ is a zero-mean Gaussian random variable in dB with standard deviation σ in dB [14]. At each sensing round, each SU measures the received power: signal power of $P_t - PL(d)$, if PU is present, or the received noise power, if PU is absent. Before sending their values to others, nodes quantize their sensed power with q bits. Each node's initial value is a discrete power level in $\{0, 1, ..., 2^q - 1\}$. After enough number of consensus iterations, where nodes communicate intermediate values with each other, each node compares its final estimate of average power with the detection threshold (γ) and makes its final binary decision. The spectrum sensing performance of an SU is characterized by the probability of false alarm (Pr_{FA}) and misdetection (Pr_{MD}):

$$Pr_{FA} = Pr(\Gamma > \gamma | H_0) \quad \text{and} \quad Pr_{MD} = Pr(\Gamma < \gamma | H_1)$$

where Γ denotes the received power. The received signal by an individual SU can be modeled as follows:

$$y(n) = \begin{cases} w(n) & H_0 \\ s(n) + w(n) & H_1 \end{cases} \quad (1)$$

where $s(n)$ and $w(n)$ are the signal component with power P_s and the zero-mean additive white Gaussian noise with noise power P_n, respectively. If the power detector takes M samples, the test statistic is given by: $\Gamma = \frac{1}{M} \sum_{n=1}^{M} y(n) y(n)^*$. Using the central limit theorem, it can be shown that for large enough M, the test statistic for an independent node follows a normal distribution [15] as shown in (2). False alarms occur when H_0 is true; therefore, in (2) $P_s = 0$. Thus, the power detection threshold (γ) can be derived from (3) for a desired Pr_{FA} (where $Q^{-1}(.)$ is the inverse Gaussian Q-function).

$$\Gamma \sim \mathcal{N}(P_s + P_n, \frac{2(P_s + P_n)^2}{M}) \quad (2)$$

$$\gamma = P_n(1 + \sqrt{\frac{2}{M}} Q^{-1}(Pr_{FA})) \quad (3)$$

The IEEE 802.22 working group for spectrum sensing modeling recommends setting a maximum of 1% or 10% for Pr_{FA} of independent nodes and derive the power detection threshold based on the noise power as described above in order to evaluate cooperative Pr_{FA} and Pr_{MD} [15] [16]. We follow the same framework in this paper.

IV. Average Consensus-based Schemes for Cooperative Spectrum Sensing

In average consensus-based DCSS schemes, the goal of the SU's is to estimate the average of the received power by all of the SU's. Each sensing round includes a number of consensus iterations. First, each SU measures its own received power as its initial value; then in each subsequent iteration, the SU's broadcast their values and update their own values (average estimates) based on what they receive from the other nodes. After enough number of iterations, the estimates of all of the nodes converge close to the global average. At this time, each node independently compares its own final estimate of the average power with the detection threshold and makes its final decision about the PU presence. A consensus-based DCSS scheme can only be practical if it converges in a sufficiently small number of iterations, otherwise, the scheme will not be cost-effective due to the communication and computation overhead. In this section we look at two categories of average-consensus algorithms: gossip-based and weighted linear iteration-based, and we propose an equally-weighted linear iteration-based scheme for DCSS.

A. Gossip-based

The Push-Sum protocol has been proposed as a gossip-based solution for average consensus problem [2]. Each sensing round includes I iterations. Each node $i \in \{1, ..., N\}$ maintains a sum, $s_i(c)$, and a gossip weight, $w_i(c)$, at all iterations $c \in \{0, 1, ..., I\}$. These are initialized as: $s_i(0) =$ *received power at node* i, $w_i(0) = 1$. Initially, each node i sends the pair $(s_i(0), w_i(0))$ to itself and in all subsequent iterations, it sends a fraction of its sum and gossip weight to one or more randomly chosen neighbors; the remaining of the sum and the weight are sent to itself (kept at i). Each node i updates its sum and weight at each iteration as follows [2]:

$$s_i(c+1) = \sum_{j \in N_i \cup i} \hat{s}_j(c), \qquad w_i(c+1) = \sum_{j \in N_i \cup i} \hat{w}_j(c) \quad (4)$$

where $\hat{s}_j(c)$ is the sum fraction and $\hat{w}_j(c)$ is the weight fraction received from j at iteration c. At iteration c, node i's estimate of the average is $\frac{s_i(c)}{w_i(c)}$ and as the iterations progress, the estimate approaches the global average. The following are two variants of this scheme:

- One-neighbor gossip: At each iteration c, each node i sends $\frac{1}{2}s_i(c)$ and $\frac{1}{2}w_i(c)$ to one randomly chosen neighbor and the remaining half is kept at i [9].
- Flooding gossip: At each iteration c, each node i broadcasts $\frac{1}{1+|N_i|}s_i(c)$ and $\frac{1}{1+|N_i|}w_i(c)$ to all of its neighbors and the remaining share is kept. Note that the nodes must know the number of their active neighbors at the current sensing round in order to calculate the share that they must broadcast.

B. Weighted linear iteration-based

The weighted linear iterative scheme follows a weighted averaging update strategy at each iteration in order to reach consensus on the average [3]. We denote the value of node i at consensus iteration c by $v_i(c)$. At each sensing round, each node i starts with $v_i(0) =$ *received power at node* i, and after I iterations, its final updated value is $v_i(I)$. The goal is that for enough number of iterations, $v_i(I), i = 1, ..., N$ converges to $\frac{1}{N}\sum_{i=1}^{N} v_i(0)$. At each consensus iteration c, each node i updates its value with a weighted linear combination of its own value and the received values from its neighbors [3]:

$$v_i(c+1) = W_{ii}(c)v_i(c) + \sum_{j \in N_i} W_{ij}(c)v_j(c) \qquad (5)$$

where W_{ij} denotes the weight for $v_j(c)$ at node i ($W_{ii}(c)$ is self-weight). Two heuristics have been proposed for choosing weights [3]:

- Metropolis: Weights are based on the larger number of neighbors in each pair of nodes: $W_{ij}(c) = \frac{1}{1+max\{|N_i|,|N_j|\}}, \ \ j \in N_i$
- Maximum-degree: Weights are based on the largest number of neighbors in the whole network [7]: $W_{ij}(c) = \frac{1}{1+\text{max degree}}, \ \ j \in N_i$

In both of the above schemes, self-weight is set such that the sum of weights is 1: $W_{ii}(c) = 1 - \sum_{j \in N_i} W_{ij}(c)$.

C. Proposed equally-weighted linear iteration-based

Inspired by the linear iteration-based scheme with two heuristic weight choices described above, we propose an approximate scheme based on equal weights. This scheme is practical for DCSS because unlike the previous schemes, it does not require any topology knowledge of the decentralized network such as the maximum degree or the degree of neighbor nodes. At each iteration, each node simply calculates an equally-weighted average of its own value and all of the received values from its neighbors. Therefore, all of the weights (including self-weight) in (5) will be essentially equal: $W_{ij}(c) = W_{ii}(c) = \frac{1}{1+|N_i|}, \ \ j \in N_i$.

This approximate heuristic does not necessarily satisfy the conditions for asymptotic convergence to the exact global average (see Equation (6) of [3]), however the convergence is faster compared to the Metropolis and maximum-degree heuristics. The reason is that for any graph, the neighbor weights of the equal-weighting scheme are greater than or equal to neighbor weights of the other two schemes (self-weights are smaller or equal accordingly). We have verified this faster convergence through extensive experiments for various graph topologies. Fast convergence is vital for an efficient DCSS scheme and since the estimated average at each node is only used for a binary decision about the PU presence, the precision of the average itself is of less importance. Therefore, as our performance results in Section VII also confirm, our proposed approximate scheme is the most efficient choice; in addition, as discussed before, it is the only practical choice among the schemes described in this section.

V. Insistent Spectrum Sensing Data Falsification

As discussed in Section IV, average consensus-based DCSS schemes are iterative and in all of the consensus iterations, all of the nodes must follow a predefined update strategy. We introduce Insistent Spectrum Sensing Data Falsification (ISSDF) as an attack where a malicious node falsifies its sensing data and broadcasts the same falsified value in every iteration of consensus. Therefore, ISSDF is different from traditional SSDF attack [5] in that here in addition to falsifying its initial value, the attacker also disregards the received values from the other nodes and does not update its estimate at all. Intuitively ISSDF attack is significantly more destructive than simple SSDF because the falsified data is repeatedly fed into the consensus process in every iteration which causes divergence from the correct average. We consider the following two types of ISSDF attacks:

1) "Always-No" ISSDF: When PU is present, an Always-No ISSDF attacker constantly broadcasts the lowest possible power level in all of the consensus iterations in order to lessen the other nodes' estimate of global average and make them decide that the PU is absent: $v_{Always-No}(c) = 0, \ \ c = 1, .., I$. These attackers try to deceive honest nodes into interfering with the PU by increasing the probability of misdetection (Pr_{MD}) in the network. This attack is very harmful because it directly conflicts with the fundamental requirement of non-interference in cognitive radio networks and can seriously disrupt the spectrum sharing mechanism.

2) "Always-Yes" ISSDF: When PU is absent, an Always-Yes ISSDF attacker repeatedly broadcasts the highest possible power level in all of the consensus iterations in order to raise the other nodes' estimate of global average and mislead them to decide that the PU is present: $v_{Always-Yes}(c) = 2^q - 1, \ \ c = 1, .., I$, where q is the number of quantization bits. These attackers try to misguide honest nodes to back-off and leave the channel. One potential motivation behind this attack is selfishness: By increasing the probability of false alarm (Pr_{FA}), the attacker aims at eliminating some of its contenders for using the free channel. This attack can significantly degrades the spectrum utilization efficiency which is the main goal of dynamic spectrum sharing.

As the naming reveals, in both of the above attack models, a malicious node will *always* be malicious and follow the same malicious strategy. Also, in our model we assume an

honest node remains honest, therefore the honesty state of the nodes does not change from one sensing round to the other. In Section VI, we describe in detail our proposed trust management scheme to mitigate ISSDF attacks.

VI. Proposed Trust Management Scheme

Our trust management scheme is based on trust scores that the nodes assign to each other. The trust score that node i assigns to node j at time step (sensing round) t, denoted by $\theta_{ij}(t)$, is a real number in the interval $[0, 1]$ that can be interpreted as the probability of j being trustworthy in the view of node i. In order to make the DCSS schemes resilient to malicious nodes, we propose that the significance of the neighbors' reports in a node's estimate update must be determined by these trust scores. Instead of simply accepting neighbors' reports, each node should gradually determine the level of trust it associates to its neighbors through interaction observations. The trust score calculation method that we use in this work is inspired by [5] and [12]. At sensing round t, after sufficient number of consensus iterations, node i makes a final binary decision about the presence of the PU and then compares the decision with the initial value that it has received from each of its neighbors j and makes the following binary observation:

$$o_{ij}(t) = \begin{cases} 1 & if \;\; g_{ij}(t) > \gamma \text{ AND } f_i(t) = \text{PU present} \\ & \text{OR } \; g_{ij}(t) < \gamma \text{ AND } f_i(t) = \text{PU absent} \\ 0 & \text{Otherwise} \end{cases} \tag{6}$$

where $g_{ij}(t)$ denotes the initial sensing report that node i received from neighbor node j in the first consensus iteration and $f_i(t)$ is node i's final decision in sensing round t. If j's report is in agreement with i's final decision, the observation is 1 and otherwise it is 0. In the ISSDF attack model, an attacker repeats the same falsified value in every iteration; therefore, it is sufficient to evaluate only the values that are received in the very first iteration of consensus. Note that since there is essentially no "ground truth" about the presence or absence of the PU, the best an honest SU can do is to rely on its own final decision when evaluating a neighbor's trustworthiness.

We propose that each node i maintains a binary observation list per neighbor j denoted by the string O_{ij}. When node i makes a new observation from node j, it records it in O_{ij}. The observation list grows to a maximum length, after which the oldest observations are replaced with new ones. The trust score can be calculated as follows: $\theta_{ij}(t) = \frac{H(O_{ij})}{|O_{ij}|}$, where $H(.)$ denotes the Hamming weight of the binary string O_{ij} and $|O_{ij}|$ is the current length. Note that at each sensing round the trust scores are updated once the final decisions are made (after the final consensus iteration) and not in between consensus iterations. A node needs to make a minimum number of observations (O_{min}) from a neighbor before it can assign a non-zero trust score to it, that is if $|O_{ij}| < O_{min}$ then $\theta_{ij} = 0$. This means when updating values, nodes ignore received power levels from new neighbors with whom they have not had enough number of interactions; however, the observations from a new neighbor are recorded until enough number of those are stored. In addition, we propose that a node should assign a trust score of zero to neighbors with whom the agreement is below a predefined minimum (θ_{min}), that is if node j is in conflict with node i in more than θ_{min} fraction of their interactions then $\theta_{ij}(t)$ will be set to zero. As will be shown by the simulations in Section VII, these proprieties of the trust scheme enable honest nodes to detect malicious ISSDF nodes with a high accuracy and almost exclude them in their updates by assigning zero or low trust scores to them.

A. Incorporating trust management into average consensus-based DCSS schemes

To incorporate the trust management into the gossip-based schemes that we discussed earlier in Section IV, we alter the iteration update algorithm (4) by scaling the sum and weight using trust score as follows:

$$s_i(c+1) = \sum_{j \in N_i \cup i} \theta_{ij}(t) \hat{s}_j(c), \quad w_i(c+1) = \sum_{j \in N_i \cup i} \theta_{ij}(t) \hat{w}_j(c) \tag{7}$$

For linear iterative schemes, the trust scores are used as weights in the linear combinations (or as second-level weights in case the linear combination is already weighted). Thus, we incorporate the trust scores by modifying (5) as follows:

$$v_i(c+1) = W_{ii}(c) v_i(c) + \sum_{j \in N_i} \theta_{ij}(t) W_{ij}(c) v_j(c) \tag{8}$$

where $W_{ii}(c) = 1 - \sum_{j \in N_i} \theta_{ij}(t) W_{ij}(c)$. Note that our proposed trust system does not introduce any communication overhead. For computational overhead, incorporating the trust scores requires $I \times |N_i|$ extra multiplications (I is the number of iterations). Also, $|N_i|$ comparisons are needed at each sensing round. This overhead is reasonably tolerable for realistic scenarios.

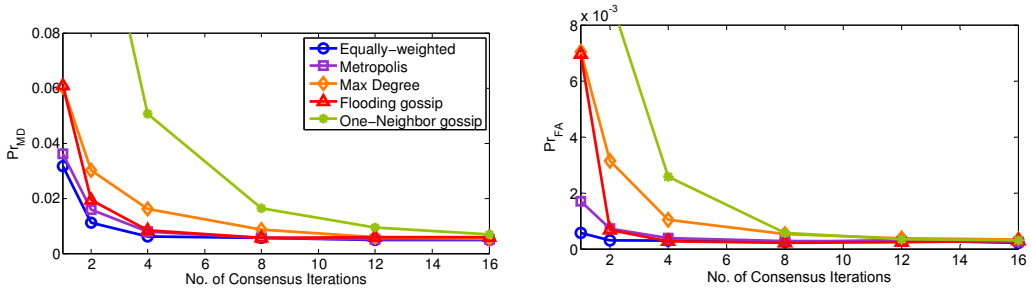
VII. Experimental Results

A. Simulation setup

For our simulations we consider a cognitive radio ad hoc network operating in TV whitespace (615 MHz center frequency) with 40 SU's that are spread and moving in a square 250 m×250 m area. The PU is located 20 km away from the center of the square area and when transmitting, its transmit power is 54 dBm. $q = 6$ bit is used for quantization. As discussed in Section III, for evaluation purpose we derive the detection threshold by setting a maximum false alarm rate for independent (non-cooperative) SU's. Here we set maximum $P_{FA} = 0.1$ [15] [16] and derive the required threshold based on the noise power as in (3). The simulation parameters are listed in Table I. Through Monte Carlo simulations, we derive Pr_{FA} and Pr_{MD} in the network as the average portion of the honest nodes in the network that make a false alarm and misdetection error in a sensing round, respectively. Each of our Monte Carlo runs, starts with a new random network and continues for 1600 (virtual) seconds during which the nodes are moving. The nodes perform sensing every 2 second, therefore each run includes 800 consecutive sensing rounds; after which the experiment is repeated with a new random graph. We run our experiment for different DCSS schemes in the presence of a variable number of ISSDF Always-No or Always-Yes attackers. In addition, for comparison, we also run the experiment for centralized and non-cooperative scenarios. In the centralized scenario, a fusion center calculates the

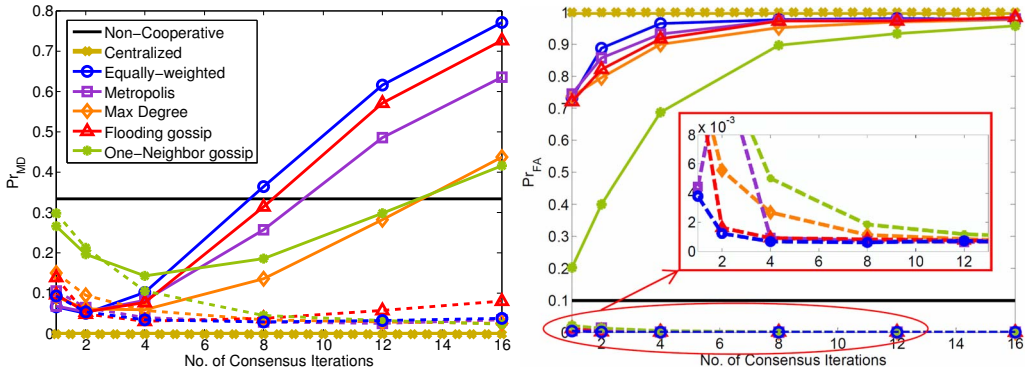
TABLE I. SIMULATION PARAMETERS

Path Loss Model		Way Point Mobility Model		Noise and Threshold		Monte Carlo Sim. and Trust	
PU Distance from CRAHN	20 km	CRAHN Area	250 m×250 m	Noise Figure	11 dB	# SU Nodes	40
PU Antenna Height	20 m	Min Velocity	1 m/s	Channel Bandwidth	6 MHz	SU Node Range	60 m
SU Antenna Height	1 m	Max Velocity	3 m/s	Ultimate Noise Floor	-174 dBm	Simulation Time	1600 s
Center Frequency	615 MHz	Min Pause	120 s	Max Pr_{FA} (non-cooperative)	0.1	Sense Interval	2 s
Log-normal Shadowing Standard Deviation	5.5 dB	Max Pause	360 s	Noise Power	-95.22 dBm	Min # of Observations for trust (O_{min})	6
Transmit Power (P_t)	54 dBm			Detection Threshold	-94.23 dBm	Min trust (θ_{min})	0.5



(a) PU is present (b) PU is absent

Fig. 1. Comparing different consensus-based DCSS schemes in an honest network.



(a) PU is present, 20% Always-No ISSDF attackers (b) PU is absent, 20% Always-Yes ISSDF attackers. Inset sub-figure zooms in on the dashed curves.

Fig. 2. Comparing different consensus-based DCSS schemes under 20% ISSDF nodes. Solid lines: w/o trust, Dashed lines: with trust.

average of the values from all of the nodes and makes one final decision for the whole network. On the other hand, in non-cooperative case, each node independently makes its final decision only based on its own sensing. In non-cooperative scenario, as mentioned above, $Pr_{FA} = 0.1$ and from simulations based on our specific setup, we measure $Pr_{MD} = 0.34$. These will serve as our comparison references.

B. Simulation results

Figure 1 contrasts the performance of different DCSS schemes discussed in Section IV. We compare them in an honest network case (i.e. no attackers) in terms of misdetection and false alarm rate and we show the results for different number of consensus iterations. Obviously, when there are no malicious nodes, the performance improves with more consensus iterations. Also as expected, all of these cooperative schemes perform much better than non-cooperative scheme (note that $Pr_{FA} = 0.1$ and $Pr_{MD} = 0.34$ for non-cooperative). The equally-weighted linear iterative scheme performs the best among all in both Pr_{MD} and Pr_{FA}.

Figure 2 shows the error rates for different schemes, if 20% of total nodes are ISSDF attackers. In addition, the error rates for non-cooperative and centralized scenarios are shown for comparison. In this figure, solid and dashed curves represent without and with trust management, respectively. Without trust management, error rates increase quickly with more consensus iterations. This is because as discussed previously, ISSDF attackers repeatedly broadcast their falsified data in every consensus iteration and therefore, with more iterations their destructive effect only becomes stronger. In fact, if unattended, the ISSDF attack can make the cooperation useless and even worse than the non-cooperative scenario. Our proposed trust management scheme significantly improves the performance in both cases of Always-Yes and Always-No ISSDF attacks. The trust-aware schemes converge to much lower error rates compared to oblivious schemes within only a small number of consensus iterations. Obviously, less number of iterations is more efficient and thus more desirable.

We can see from Figure 2 that the effect of Always-Yes ISSDF attackers are more significant than that of Always-No attackers. With 20% Always-Yes nodes, cooperation becomes disadvantageous: the centralized scheme results in $Pr_{FA} = 1$ and for the consensus schemes also Pr_{FA} quickly grows to

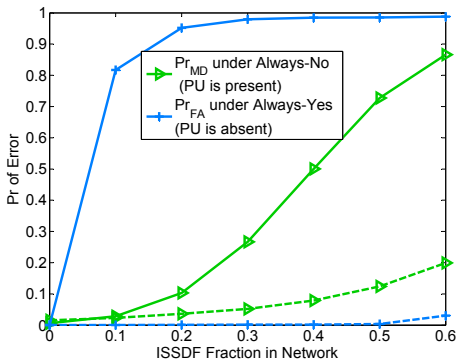


Fig. 3. Equally-weighted linear iterative DCSS scheme under varying fraction of ISSDF attackers. Number of consensus iterations is 4. Solid lines: w/o trust, Dashed lines: with trust.

1 as we increase the number of consensus iterations. As described previously, the detection threshold is calculated based on the noise power and maximum Pr_{FA}; thus in our setup the threshold is close to the minimum power level and much lower than the maximum level. As a result, it is much easier for a malicious node to cause a false alarm by broadcasting high power values (Always-Yes: 2^q-1) than to cause a misdetection by broadcasting low values (Always-No: 0), where q is the number of quantization bits.

It is easy to see from Figure 2 that the equally-weighted scheme is the most sensitive to the ISSDF attack. This is essentially due to the fact that this scheme offers a relatively better diffusion speed and therefore the falsified data is diffused more quickly, causing more errors. Nevertheless, by incorporating trust, the equally-weighted scheme performs better than, or as good as, the other schemes. As discussed in Section IV, the equally-weighted scheme is the most practical scheme among all for a decentralized network, where the nodes do not have any knowledge about the network topology. We conclude that our proposed trust-aware equally-weighted linear iteration-based scheme is the best choice for DCSS.

Figure 3 presents the error performance of the equally-weighted scheme under varying number of malicious nodes. Here, the number of consensus iterations is fixed to 4 since it is a cost-effective choice based on the results shown in Figure 2. Figure 3 reveals the significant improvement in performance by integrating trust management. The improvement is greater in the case of Always-Yes attack because independent nodes make false alarm errors with lower rate ($Pr_{FA} = 0.1$ vs. $Pr_{MD} = 0.34$ for non-cooperative) and therefore the trust scheme is more effective in this case. As can be seen from the figure, our proposed trust scheme greatly suppresses the ISSDF attacks even when a large portion of the network is malicious. For example, in the presence of 40% Always-No attackers, Pr_{MD} is improved by trust from about 0.5 to 0.07 and for 40% Always-Yes, the trust-aware scheme achieves Pr_{FA} as low as 0.002, while the error rate is 0.98 in the oblivious case.

VIII. Conclusions

Insistent spectrum sensing data falsification (ISSDF) is introduced as a new attack that significantly degrades the performance of consensus-based distributed spectrum sensing schemes. Furthermore, this is the first work that compares different categories of average consensus-based spectrum sensing schemes in an honest network scenario or under the ISSDF attack. Through simulations for a cognitive radio ad hoc network operating in TV whitespace, we analyze the effect of the introduced attack on the false alarm and misdetection error rates. We propose a trust management method that greatly improves the performance by alleviating attackers' disruptive effect on honest nodes' decision makings. We also analyze the number of required consensus iterations and show through simulations that a small number of trust-aware iterations is sufficient for converging to the lowest error rates. Finally, through comparisons between different schemes, we propose the trust-aware equally-weighted linear iteration-based method as the best and most practical choice for consensus-based cooperative spectrum sensing.

References

[1] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *IEEE International Conference on Communications (ICC)*, vol. 4, 2006, pp. 1658–1663.

[2] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *44th Annual IEEE Symposium on Foundations of Computer Science Proceedings*, Oct 2003, pp. 482–491.

[3] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*. IEEE Press, 2005.

[4] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[5] R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM. The 27th Conference on Computer Communications*, April 2008, pp. 31–35.

[6] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference*, vol. 3, June 2005, pp. 1859–1864.

[7] Z. Li, F. R. Yu, and M. Huang, "A distributed consensus-based cooperative spectrum-sensing scheme in cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 383–393, 2010.

[8] N. Ahmed, D. Hadaller, and S. Keshav, "Guess: Gossiping updates for efficient spectrum sensing," in *Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking*. New York, NY, USA: ACM, 2006, pp. 12–17.

[9] A. Vosoughi, J. Cavallaro, and A. Marshall, "A cooperative spectrum sensing scheme for cognitive radio ad hoc networks based on gossip and trust," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2014, pp. 1175–1179.

[10] S. Sundaram and C. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents - part I: Attacking the network," in *American Control Conference*, June 2008, pp. 1350–1355.

[11] X. Liu and J. Baras, "Using trust in distributed consensus with adversaries in sensor and other networks," in *17th International Conference on Information Fusion (FUSION)*, July 2014, pp. 1–7.

[12] T. Qin, H. Yu, C. Leung, Z. Shen, and C. Miao, "Towards a trust aware cognitive radio architecture," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, no. 2, pp. 86–95, Sep. 2009.

[13] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 257–269, July 2003.

[14] T. S. Rappaport, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.

[15] S. J. Shellhammer, "Spectrum sensing in IEEE 802.22," *IAPR Wksp. Cognitive Info. Processing*, pp. 9–10, 2008.

[16] "Spectrum sensing simulation model, IEEE 802.22-06/0028r10."