Trust Computation Through Anomaly Monitoring in Distributed Cognitive Radio Networks

Shameek Bhattacharjee, Saptarshi Debroy and Mainak Chatterjee
Department of Electrical Engineering & Computer Science
University of Central Florida
Orlando, FL 32816
Email: {shameek, saptarsh, mainak}@eecs.ucf.edu.

[1] ***Abstract*—The open philosophy of cognitive radio networks makes them vulnerable to various types of attacks which compromises the efficiency of these networks. One such attack is the Spectrum Sensing Data Falsification (SSDF) attack where malicious nodes report false spectrum occupancy data to others which when used leads to inference that is far from the true spectrum occupancy. Thus, there is a need to identify the malicious nodes or at least find the trustworthiness of nodes such that the data sent by malicious nodes could be filtered out.**

This paper proposes a scheme for trust based fusion by monitoring anomalies in advertised spectrum usage reports by secondary nodes. Such monitoring leads to evaluation of *trust* of a node by its neighbors. The calculated trust is then used to determine if a neighbor node's advertised data could be used for fusion or not. We provide a heuristic trust threshold for nodes to disregard malicious nodes while fusing the data, which holds good for any probability of attack. To validate our model we conduct extensive simulation experiments. Our results show that majority of the nodes are able to fuse data with greater accuracy for various probabilities or intensities of attack. We also compare our results with blind fusion scheme and observe improvement in accuracy of fusion from individual nodes' as well as overall network's perspective. We also report a very counter-intuitive observation: at lower probabilities of attack, a malicious node's contribution to the overall gain in cooperation is more than the damage done.

I. Introduction

In cognitive radio networks (CRN), the unlicensed secondary nodes continuously sense for available bands and use them in an opportunistic manner [7]. Due to typical channel impairments like shadowing and multi-path fading, localized spectrum sensing by individual nodes is often erroneous. Thus participation of secondary nodes in cooperative or collaborative spectrum sensing leads to more accurate inference of primary occupancy [3]. The negative side of cooperative sensing is that it may be impaired by Spectrum Sensing Data Falsification (SSDF) or Byzantine attacks, where participating nodes lie about their local sensing result [1], [5]. The intensity of malicious behavior can be an all out attack (binary vector inversion) or a more intelligent strategy of injecting false information on a varying fraction of the total number of channels. Such false information adversely affects the cooperative fusion results compromising system efficiency.

Most of the existing work on SSDF attacks that deals with malicious node detection methods [5] or robust fusion of advertised data [1] consider centralized CRN where there is a central entity to fuse advertised spectrum data. For ad-hoc networks, monitoring packet forwarding anomalies was introduced in [6] to track routing misbehavior. In ad hoc networks, packet forwarding anomalies decide the success and failure counts which are then used to build and manage trust. As far as trust management in CRN is concerned, authors in [2] provide a methodology for building and maintaining trust metrics based on beta distribution model; however, success and failure are assumed to be known. In [8], authors provide yet another trust establishment scheme for centralized CRN based on success and failure counts. Although these works introduce the concept of trust management in cognitive radio networks, they only deal with mapping success and failure counts to trust metrics without providing a framework that actually measures and deciphers the number of successful and failed interactions which is the basis of trust calculation. Identifying such interactions as success or failure is an important issue in cooperative cognitive communication which still remains unanswered. The hurdles discussed above motivates the investigation to understand successful interactions with respect to cooperative spectrum sensing in an ad-hoc CRN where each node must carefully assess the trustworthiness of other nodes before using their advertised data.

In this paper, we propose an intelligent trust based fusion model for robust fusion in the presence of Byzantine attacker nodes. The trust is evaluated by capturing anomalies in spectrum sensing data advertisements without the nodes needing to know the locations of others. All nodes, including the malicious ones, share their local spectrum occupancy reports with the neighbors. In the absence of a centralized fusion center, each node performs its own fusion based on received spectrum information. Each node evaluates a trust coefficient for every neighboring node, and consults the coefficient to decide whether it is advisable to include a neighbor's advertised information for fusion. The cognitive nodes do not require any prior information except the location of fixed *primary transmitters* which are already known. Our method enables a secondary node to predict the bounds on the received power levels of its neighbors. A normalization criterion is then used to predict binary occupancy vectors from those bounds. The results of comparison between the predicted and the advertised vectors of a neighbor is logged as 'matches' and 'mismatches' (a cooperation misbehavior) and these form

[1]This research was partially supported by the National Science Foundation, under award no. CCF-0950342.

the basis of the trust that secondary nodes constructs for all its neighbors. Mismatches indicate presence of anomalies. Higher trust value signifies more trustworthiness. To test and validate the proposed models, we conduct extensive simulation experiments. We find that our proposed model ensures that malicious nodes have significantly less trust coefficients than honest nodes irrespective of their attack probabilities. We identify an heuristic threshold range for the trust coefficients (0.45 to 0.50) below which a neighbor's advertised occupancy vector can be disregarded. We also compare our proposed trusted fusion method with blind fusion technique for validating the benefits of our fusion scheme. We also find that at lower probabilities of attack the contribution of a malicious node to the entire network in cooperative sensing is more than the damage it does. Thus, including them for fusion does more good than harm to the network overall.

TABLE I
NOTATIONS

Symbol	Meaning
N_i	Neighbor set of node i
H	Set of honest nodes
M	Set of malicious nodes
γ_{th}	Common threshold used to normalize power vectors
$s_{T_{ik}}$	Distance between node i and primary tower T_k for channel k
d_k	Binary Decision on a channel k, $d_k \in 0, 1$
j	Set of all neighbors of i, $j \in N_i$
P^i	Measured power vector on n channels at node i
B^i_{act}	Actual binary occupancy vector formed at i
B^i_{adv}	Advertised binary occupancy vector by node i
$P^{ij}_{predict}$	Vector of power ranges for neighbor j predicted by i
D^j_i	Binary occupancy of node j, predicted by i
$d^j_k\|_{predict}$	Predicted decision on any channel k, for D^j_p
$(\alpha, \beta, X)^j$	Three tuple trust evidence
$E^j_{trust_i}$	Reputation or trust of neighbor j calculated by node i
BF^i_{blind}	Fusion result at node i, when all j are included in fusion.
TBF^i	Fusion result Based on selective inclusion of j based on trust

II. System Model

We assume all secondary nodes continuously undergo spectrum sensing to determine whether a channel is occupied or not. Let us assume secondary node i constructs its observed occupancy vector as: $B^i_{act} = [d_1, d_2, \cdots, d_n]$, where d_k is 1 or 0 depending on whether the channel is occupied or unoccupied, and n is the number of channels being monitored. Once this binary vector is created, a secondary node would broadcast this information to its neighboring nodes. Similarly, a secondary node would also hear broadcast messages (binary occupancy vectors) from its neighbors. Based on the vectors a node receives, the node will employ a fusion technique to obtain a better estimate about the spectrum usage that can significantly improve the performance of spectrum sensing [1], [3]. Such cooperative sensing has other benefits such as mitigating shadowing and multi-path effects.

We consider that the malicious nodes do not report their occupancy vectors truthfully; rather they inject errors in their occupancy vectors by flipping the bits in the vector. Flipping 0 to 1 implies that the channel is occupied when in reality it is unoccupied. Flipping 1 to 0 implies that an occupied channel is reported as unoccupied. We denote probability of attack P_{attack}, as the percentage of channels that a malicious node changes from its actual observed vector.

A. Assumptions

1. We consider an ad-hoc secondary network with N nodes; H is the set of honest nodes and M malicious/dishonest nodes. The malicious nodes launch independent attacks without collaboration. We assume $\eta(M) < \eta(H)$, since in a realistic network, the number of malicious nodes is less than number of regular honest nodes. The secondary network has no dedicated central fusion center or allocation authority, and each individual node fuses the spectral sensing data it hears from other nodes. A node fuses the information it gathers and forms its opinion on the availability of spectrum.
2. The nodes need not know geographical coordinates of other nodes involved in cooperation. We assume the transmit power level of all secondary nodes are same. Knowledge of the transmitter output power, channel losses, and antenna gains with the appropriate path loss model allows us to find distance between the two nodes using R.S.S. (Received Signal Strength) through localization or lateration [10].
3. Unlike in [5], which discusses a more restrictive fusion model (AND fusion rule), we use majority voting fusion rule, which gives more flexibility towards errors committed by nodes and/or malfunctioning nodes.
4. Each primary transmitter *whether it chooses to transmit or not*, transmits only on one channel; so the channel associated with a primary transmitter is known. The primary transmitter that transmits on channel k, is referred as T_k, and since it is fixed, its coordinates (x_{T_k}, y_{T_k}) are known to the nodes.
5. We assume nodes use some interference aware channel access framework, so that the secondary nodes do not interfere with others who might use the same channels. Interference awareness is outside the scope of this paper.
6. We assume that the data sent by a node is not corrupted by the channel itself.

III. Monitoring Observations

In the context of cooperative spectrum sensing in cognitive radio networks, if we need to monitor for SSDF attack, we need to monitor anomalies in spectrum sensing data reported among the nodes participating in cooperation. In this section, we deduce successful and failed interactions (anomalies) based on which trust is calculated. We do this by predicting bounds on received power of a neighbor and apply a normalization criterion to obtain predicted binary vector, which is compared with advertised binary vector to unravel possible anomalies. Subsequently, we propose a trust based filtered fusion scheme which uses evidence based trust to decide whether to consider the node's advertised vector in fusion or not.

A. Predicting Power Vector

Suppose node i measures the power vector $P^i = \{\gamma^i_1, \gamma^i_2, ...\gamma^i_n\}$, where γ^i_k is the power received on channel k and n is the number of channels. Each node i forms its binary vector $B^i_{act} = [\ d^i_1, d^i_2, ...d^i_n\ ]$ from it's power vector P^i by comparing γ^i_k with threshold γ_{th}, where

$$d^i_k \begin{cases} = 1 & \text{when } \gamma^i_k \geq \gamma_{th} \\ = 0 & \text{when } \gamma^i_k < \gamma_{th} \end{cases} \quad (1)$$

Each node i, advertises a public binary vector B^i_{adv}.

$$B^i_{adv} \begin{cases} = B^i_{act} & \text{if node } i \in H \\ \neq B^i_{act} & \text{if node } i \in M \end{cases} \quad (2)$$

Just the way node i advertises its binary vector, it also hears similar advertisement from its neighbors. For a neighboring node $j \in N_i$, node i estimates its possible power vector using their mutual distance deciphered through received signal strength (RSS) localization [10]. Though it is difficult for node i to accurately predict the power vector of node j, nevertheless it can always estimate the lower and upper bounds using RSS models. Let us describe how node i estimates the upper and lower bounds of power vector as $P^{ij}_{predict}$.

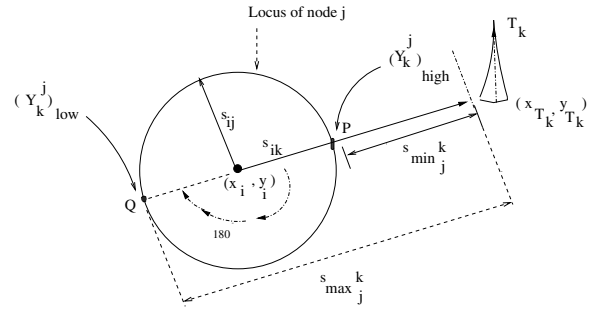


Fig. 1. Max. and Min. RSS range on channel k of neighbor node j

Assuming transmit power of all nodes are same, node i calculates the distance between the node j and itself whenever it receives a signal (vector) from it as s_{ij}. Based on the distance s_{ij}, node j may be anywhere on the circle with node i at the center. We draw a straight line from the center of the circle to the primary transmitter T_k located at (x_{T_k}, y_{T_k}) as shown in Fig 1. Under ideal conditions, the RSS due to T_k will be maximum on the circle that is closest to T_k, i.e., on point P and minimum at a point Q that is farthest from T_k. We denote the power levels at these two locations as $[\gamma^j_k]_{high}$ and $[\gamma^j_k]_{low}$ at distances $s_{min^k_j}$ and $s_{max^k_j}$, respectively. For all locations on the circle, the RSS on channel k varies between $[\gamma^j_k]_{high}$ and $[\gamma^j_k]_{low}$.

Using commonly used model for RSS [4], we get

$$\gamma^i_k = P_k \times \frac{A^2}{s^\alpha_{ik}}; \quad (3)$$

where A = frequency constant, α is path loss factor, s_{ik} is the distance between T_k and node i, and P_k is the transmit power of T_k.

$$[\gamma^j_k]_{high} = P_k \times \frac{A^2}{s^{k\ \alpha}_{min_j}}; \quad (4)$$

$$[\gamma^j_k]_{low} = P_k \times \frac{A^2}{s^{k\ \alpha}_{max_j}}; \quad (5)$$

We divide Eqn. 3 with Eqn. 4 and Eqn. 5. Since s_{ik}, $s_{min^k_j}$ and γ^i_k are known to node i, it is easy to find $[\gamma^j_k]_{high}$ and $[\gamma^j_k]_{low}$, respectively. Node j may be anywhere on the circular locus. Now the predicted power vector of node j is a 2-tuple vector $P^{ij}_{predict}$= $\big[([\gamma^j_1]_{low}, [\gamma^j_1]_{high}), ([\gamma^j_2]_{low}, [\gamma^j_2]_{high}), ..., ([\gamma^j_n]_{low}, [\gamma^j_n]_{high})\big]$. With the estimated power vector being known, the inference drawn by a node j on channel k is,

$$d^j_k|_{infer} = \begin{cases} 0 & \text{if both } [\gamma^j_k]_{low} \text{ and } [\gamma^j_k]_{high} \leq \gamma_{th}; \\ 1 & \text{if both } [\gamma^j_k]_{low} \text{ and } [\gamma^j_k]_{high} \geq \gamma_{th}; \\ X & \text{otherwise} \end{cases} \quad (6)$$

Eqn. 6 is the normalization criterion. When both the lower and higher predicted power levels on a channel are less than γ_{th}, it implies that channel i is not being used by any primary transmitter, i.e., channel is unoccupied. So in this case $d^j_k|_{infer}$ is inferred as 0. Similarly, if both the lower and higher predicted power levels are greater than γ_{th}, $d^j_k|_{infer}$ is inferred as 1. Such inference can be drawn for the above two scenarios. However, no inference can be drawn when one of $[\gamma^j_k]_{low}$ and $[\gamma^j_k]_{high}$ is above γ_{th} and the other is below γ_{th}. We denote such cases as X. Now node i compares $D^j_i = [d^j_1|_{infer}, ..., d^j_n|_{infer}]$ with received $B^j_{adv} = [d^j_1, ...d^j_n]$ on corresponding channels k for matches and mismatches.

A match occurs when on channel k, $d^j_k|_{infer} = d^j_k$. A mismatch occurs if $d^j_k|_{infer} \neq d^j_k$. If $d^j_k|_{infer} = X$ no inference can be drawn. For each neighbor j, node i computes the number of matches, mismatches, and no inference with B^j_{adv} as $(\alpha, \beta, X)^j$ respectively.

B. Trust Model

The logic behind the trust evidence is to evaluate how many have achieved trustworthiness (α), how many inferences have *not* succeeded to achieve trust (β), and how many inferences are uncertain. To account for the X channels where no inference could be drawn, we consider them in the ratio of $\alpha : \beta$. Thus the number of matches is updated as $\alpha^j + \frac{X_j}{\alpha^j+\beta^j} \times \alpha^j$. Based on the proportion of matches, the instantaneous trust coefficient that node i computes for node j is given by

$$E^j_{trust_i} = \frac{\alpha^j \times (1 + \frac{X^j}{\alpha^j+\beta^j})}{\alpha^j + \beta^j + X^j} \quad (7)$$

where $0 < E^j_{trust_i} < 1$.

C. Trust based Selective Fusion

Using the computed trust coefficients, we study the performance of two fusion schemes: blind fusion and proposed trust-based fusion.

1) Blind Fusion: For blind fusion, node i considers all its neighbors to be honest and includes B^j_{adv} from all its neighbors along with its own B^i_{act}. We formally define Blind Fusion as $BF^i_{blind} = \nabla[B^j_{adv} \oplus B^i_{act}]$, $j \in N_i$ where ∇ is the operator for majority voting rule. Majority voting is a popular fusion rule where final fused inference on a channel is based on what at least half the neighboring nodes advertise with all the nodes treated equally. $\oplus$ is the operator for combination.

2) Trust-Based Fusion (TBF): We propose a fusion scheme whereby we only consider neighboring nodes whose $E^j_{trust_i}$ is higher than some trust threshold, Γ_{opt}. (Later in Section IV, we show how to find the heuristic threshold). Thus, for trust-based fusion, node i only considers those neighbors whose $E^j_{trust_i} \geq \Gamma_{opt}$. In effect, the fusion is done with information from trusted nodes only.

$$If E^{j}_{trust_i} \begin{cases} \geq \Gamma_{opt} & \text{Node } j \text{ trusted ;} \\ < \Gamma_{opt} & \text{Node } j \text{ is not trusted} \end{cases} \quad (8)$$

We define Trust based selective fusion result as $TBF^i = \nabla[TFS_i \oplus B^i_{act}]$, where TFS_i is the trusted fusion set of binary vectors accumulated by node i using equation 8, which includes B^j_{adv} of trusted nodes only.
Although the nodes are not aware of the ideal scenario, we are aware of what would have been the ideal fusion result, which is the case when for all node $j \in N_i, B^j_{act} = B^j_{adv}$, so we define Ideal Fusion result for node i, $BF^i_{ideal} = \nabla[B^j_{act} \oplus B^j_{act}]$.

IV. Simulation Model and Performance Analysis

In this section, we study the performance of our proposed technique and its effectiveness in capturing cooperation misbehavior. First, we provide results for the average trust values of different nodes, which reflect that our scheme is successful in capturing cooperation misbehavior through trust coefficients on a customized simulator. Results show that the malicious nodes have a trust distribution significantly lower than honest nodes. Secondly, we show how we use the *TBF* scheme to filter out advertised data from malicious nodes. We also show *TBF* is robust than blind fusion. In the simulation, we evaluate performance metrics, by a parameter 'Mismatch'. For a particular case of fusion scheme, mismatch is the difference between BF^i_{ideal} and the corresponding TBF scheme. Mismatch reflects deviation of *Cooperative* sensing accuracy of nodes in coexistence with malicious nodes. We compare the 'Mismatch' for each fusion scheme to show that *TBF* works better than blind fusion. We compare average mismatch of blind fusion and *TBF* from overall network's perspective.

A. Simulation set up

We simulate a network of square 60x60m grid, with 30 randomly scattered nodes out of which 9 nodes are programmed to be malicious. All nodes continuously scan 50 channels, record the signal power on each of them, and create the binary occupancy vector which they then advertise. The malicious nodes attack (i.e., change the bits in the channel occupancy vector) with a probability between 0.2 to 0.8. It is to be noted that high probability of attack facilitates easy detection and at the same time very low attack probability do not significantly effect the network. Transmission range of all nodes is considered to be 20m, thus a node hears broadcasts from all the nodes that are in a 20m radius.

B. Trust Measurement

In Fig. 2, we see the difference in trust distribution between honest and malicious nodes when malicious nodes attack with probability 0.5. We measure average trust of each node as evaluated by all other neighboring nodes. We observe malicious nodes have significantly low trust values than the honest nodes. In Fig 3(a), we plot a particular malicious node 14's trust against various probabilities of attack and observe that the node's trust is lowered as it becomes more aggressive in attacks, i.e., the more it attacks the more damage it does to its trustworthiness. This is in general true for all malicious nodes which is shown in Fig 3(b). We observe that the average trust for set of honest nodes does not change, but average trust for malicious nodes decreases with increased attack probability.

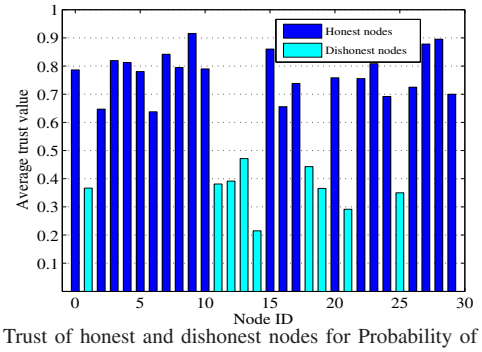
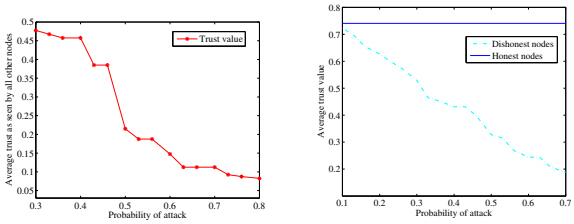


Fig. 2. Trust of honest and dishonest nodes for Probability of attack=0.5



(a) (b)

Fig. 3. Scenario:(a) Node 14 (malicious node) trust decrease with increased probability of attack (b) All malicious nodes trust decrease

C. Threshold for trust-based fusion

From Fig 4, we obtain the heuristic value of trust threshold Γ_{opt} for trust based fusion. We perform a *TBF* with Γ ranging from 0.2 to 0.8 and compare how 'mismatch' varies from ideal results. Fig. 4 shows that for very low values of threshold there are more mismatches since most of the malicious nodes are included for the purpose of fusion. However, as we increase the threshold, malicious nodes start getting discarded and mismatch decreases. When the threshold is very high (above 0.6), the mismatch again increases as high threshold means that we are also discarding the honest nodes along with the malicious nodes. Since our goal is to minimize the total average mismatch from ideal scenario, we notice a range of threshold values from 0.45 to 0.51, where the average mismatch is least for different probabilities of attack. From individual node's perspective, the idea is to exclude maximum number of malicious nodes from fusion. We choose Γ_{opt} as 0.51, because we can exclude malicious nodes even at lower probabilities of attack. This is evident from Table IV-C containing average trusts for malicious nodes for different probabilities of attack. Now if Γ_{opt} was 0.45, the nodes are successful in excluding majority of malicious nodes from fusion for higher probabilities of attack but not for lower probabilities of attack because malicious nodes have lower trust values when they attack more aggressively. If we take 0.51 as threshold, then more malicious nodes are excluded for lower probability of attack. However if we take $\Gamma_{opt} = 0.45$, we allow a few malicious nodes in fusion for lower probability of attack.

D. Performance of trust based selective fusion

Using $\Gamma_{opt} = 0.50$, we compare the trust-based fusion with blind fusion fusion for 'mismatch' shown in Fig 5. We see that the total mismatches are far more in Blind Fusion, than the trusted and selective fusion. We also provide the performance of TBF from individual node's perspective in Fig 6(a) and

Malicious Node ID	Trust Value for $P_{attack} = 0.3$	Trust Value for $P_{attack} = 0.5$	Trust Value for $P_{attack} = 0.7$
1	0.56667	0.36667	0.24167
11	0.55625	0.38125	0.26094
12	0.48333	0.39167	0.15000
13	0.56786	0.47143	0.30714
14	0.47750	0.21500	0.11250
18	0.59643	0.44286	0.24687
19	0.52812	0.36563	0.23214
21	0.46667	0.29167	0.17500
25	0.51667	0.35000	0.16111

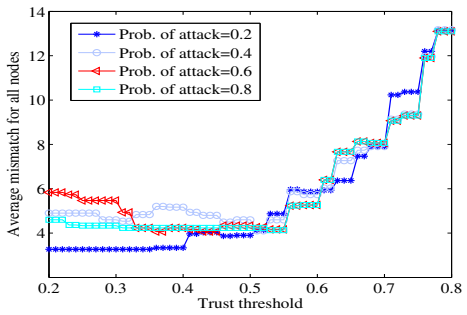


Fig. 4. Choosing threshold for trusted fusion

Fig 6(b). To measure what percentage of nodes are benefited using our framework, even at high probabilities of attack of 0.8, 90% of the nodes have average mismatches less than 8 as shown in Fig. 7(a). The results reflect the effectiveness of proposed *TBF* over blind fusion.

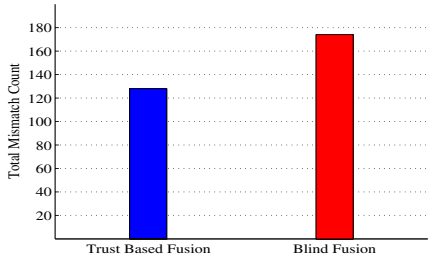
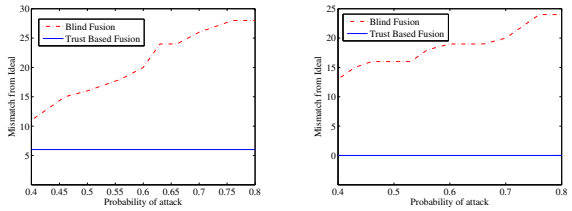


Fig. 5. Total mismatch count at P_{attack} 0.5: Overall Network Perspective

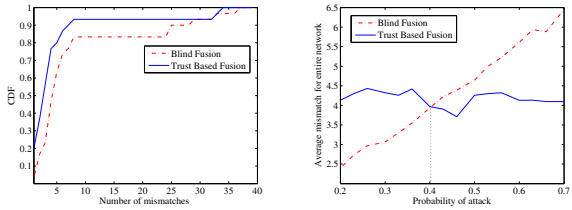


(a) (b)

Fig. 6. Individual node perspective: Blind Fusion vs. Trust Based Fusion: (a)Performance of Node 10 (b) Performance of Node 23

E. Blind and Trust based filtered fusion: Overall network perspective

An interesting comparison for average mismatch for all nodes in the network, reveal how overall cooperative sensing accuracies are effected. We see that for very lower probabilities of attack, although nodes are aware of the possibility of a node being malicious, if all nodes still allow advertisement from those nodes in the fusion the average mismatch is slightly lesser than trust based filtered fusion although the difference



(a) (b)

Fig. 7. (a) The CDF for probability of attack = 0.8; (b) Average mismatch for the entire network vs. probability of attack

is not much as shown in Fig 7(b). This is because lower attack probability means malicious nodes chooses to attack on lesser and cooperate on more channels. That is why the damage done to the network is less if we consider those nodes in fusion. However as the malicious nodes increases their attack probability, we see marked increase in mismatch for blindly fused data, while the mismatch for Trust Based Filter Scheme is much less. From simulation results (Fig 7(b)), we see that for $P_{attack} > 0.40$, *TBF* has very low overall mismatch from ideal, while for blind fusion the mismatches increases. So from malicious nodes perspective, they will have to intelligently choose a probability of attack to cause significant harm to the entire network.

V. Conclusions

We proposed a framework for evaluating trust through monitoring false advertisements of neighboring nodes, and a trust based filtered fusion scheme for cognitive radio networks. We provided a thrust threshold that may be used by each node to disregard advertisements of possible malicious node from the fusion. We are able to identify malicious nodes even at lower probabilities of attack. We also observed that for lower probability of attack, malicious nodes may also contribute to cooperation. So from malicious nodes perspective in order to significantly harm the network, it has to employ a probability of attack higher than critical probability of attack 0.40.

References

[1] R. Chen, Jung-Min Park, K. Bian ,"Robust Distributed Spectrum Sensing in Cognitive Radio Networks", *Proc. of IEEE INFOCOM*, pp. 1876-1884, Apr. 2008.

[2] K. Chen, P. Chen, N. Prasad, Y. Liang and S. Sun,"Trusted Cognitive Radio Networking", *J. of Wirel. Commun. and Mob. Comput.*, Vol. 10, Issue 4, pp. 467-485, Apr. 2010.

[3] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments", in *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*, pp. 131-136, Nov. 2005.

[4] S.-W. Jeon, N. Devroye, M. Vu, S.-Y. Chung and V. Tarokh, "Cognitive networks achieve throughput scaling of a homogeneous network", *Intl. Symposium of Modeling and Optimization in Mobile, Adhoc, and Wireless Networks (WiOPT)*, pp. 1-5, June 2009

[5] H. Li and Z. Han, "Catching attacker(s) for Collaborative Spectrum Sensing in Cognitive Radio Systems: An Abnormality Detection Approach", *In Proc. of IEEE DySpan*, pp. 1-12, Apr. 2010

[6] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks", in *Proc. of ACM MOBICOM*, pp. 255-265, 2000.

[7] J. Mitola, "Cognitive Radio: An integrated Agent Architecture for Software Defined Radio", *Ph.D. Thesis, KTH, Stockholm*, 2000.

[8] S. Parvin, S. Han, L.Gao, F.Hussain and E.Chang, "Towards Trust Establishment for Spectrum Selection in Cognitive Radio Networks", in *Proc. of IEEE International Conference on Advanced Information Networking and Applications*, pp. 579-583, Apr. 2010.

[9] Y. Sun, Z. Han, and K. J. Ray Liu, "Defense of Trust Management Vulnerabilities in Distributed Networks", in *IEEE Communications Magazine, Special Issue, Security in Mobile Ad Hoc and Sensor Networks*, Vol. 46, Issue 2, pp.112-119, Feb 2008.

[10] http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility /wifich2.html#wp1049544.