Utilizing Misleading Information for Cooperative Spectrum Sensing in Cognitive Radio Networks

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

Kevin Kwiat
Air Force Research Laboratory
Information Directorate
Rome, NY 13441
Email: kevin.kwiat@rl.af.mil

***Abstract*—In cognitive radio networks, the radios continuously scan the radio spectrum and create a spectrum usage report. Due to channel uncertainty, there are inaccuracies in these reports. Oftentimes, the radios share and fuse the observed data in order to increase the accuracy of the spectrum usage. However, malicious nodes tend to send false information (i.e., attack) in order to mislead the construction of the spectrum usage report.**

In this paper, we use a trust model to evaluate the trustworthiness of every node and use the trust values to effectively fuse the information from all nodes. A node compares the information sent by a neighboring node with the predicted information. Based on the ratio of matches (or mismatches), the neighboring node is assigned a trust value. Then, we propose a log-weighted metric utilizing trust values to distinguish malicious nodes from others. Subsequently, we propose threshold based Selective Inversion (SI) fusion and Complete Inversion (CI) fusion to effectively combine not only the information sent by honest nodes but also utilize misleading information sent by malicious nodes. We also propose a combination of the two inversion schemes. We compare the performance of the inversion based fusion schemes with blind and trust-based fusions. Results reveal better performance for inversion based fusion schemes for various intensities of attack. We also conduct simulations to evaluate the optimal thresholds that are used for invoking the inversion based fusion schemes.[1]

I. Introduction

Cognitive radio networks are poised to bring about radical changes in the wireless communications paradigm. Unlike traditional radios, cognitive radios constantly monitor the spectrum and intelligently use the radio spectrum in an opportunistic manner, both in licensed and unlicensed bands. Cognitive radios determine which portions of the spectrum are available and dynamically access the best available bands/channels.

Usually, a stand alone cognitive radio cannot accurately measure the true spectrum occupancy due to typical wireless channel impairments like channel fading, noise, multipath shadowing and fading. Thus, multiple radios engage in cooperative spectrum sensing [3], [4] where locally generated reports by a radio are sent to all its neighboring radios. A node receiving such reports fuses the information to generate a more accurate spectrum usage report. However, dependence on information from other radios introduces a vulnerability known as Spectrum Sensing Data Falsification [2] where a radio participating in cooperative spectrum sensing intentionally alters the observed spectrum occupancy. A malicious radio changes sensed usage on a channel before broadcasting; say for example, if the channel is sensed empty denoted by a value 0, the radio advertises a value 1 on that channel, and vice-versa. Such malicious intent could be to gain unfair share of the spectrum or deny spectrum to legitimate radios. Regardless, of the intent, falsification of data cripples the utility of cooperative sensing and induces wrong spectrum decisions. For example, a radio might be tempted to use a channel that is being used by the licensed user which is a violation of the regulatory aspects. Alternatively, a radio might be denied access to a channel that is usable. As far as combating such falsification attacks are concerned, there is a considerable amount of work that deals with either isolation of malicious nodes [6] or fault tolerant fusion by *disregarding* the reports from malicious nodes [2]. To the best of our knowledge, there is no effort that utilizes misleading information for information fusion.

In this paper, we introduce a new approach that utilizes our previous work [1] on trust computation through anomaly detection. We propose a framework where instead of disregarding falsified information from malicious nodes, we utilize their information to our advantage. First, we propose a trust model to evaluate the trustworthiness of every node. This is done by comparing the information sent by each node with the predicted information. Based on the ratio of matches (or mismatches), every node is assigned a trust value. Once the trust values are known, we use a log-weighted metric to distinguish the malicious nodes from others. Then, we use weighted threshold based *Selective Inversion (SI) fusion* and *Complete Inversion (CI) fusion* schemes to effectively fuse data obtained from *all* nodes. We also propose a combination of the two inversion schemes which provides better performance for any attack intensity. We find the conditions for which the combination works better. The fraction of *mismatches* (false alarms and missed detections) for different intensities of attack and node densities is considered as a performance metric. We show that fraction of mismatches for the proposed fusion techniques is always less than trust based fusion that completely ignores the malicious nodes. In the end, we demonstrate that we could utilize false information sent by malicious nodes to increase the gain in cooperative spectrum sensing.

[1] Approved for Public Release; Distribution Unlimited: 88ABW-2013-0878, 22Feb13.

978-1-4673-3122-7/13/$31.00 ©2013 IEEE

II. System Model and Assumptions

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 that the number of malicious nodes ($\eta(M)$) is less than the number of regular nodes ($\eta(H)$). Each node i fuses the spectral sensing data it receives from its neighbors. We make the following assumptions.

• 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 (actual) binary 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 (advertise) this information to its neighboring nodes as B^i_{adv}. For a malicious node, $B^i_{adv} \neq B^i_{act}$ and for honest nodes both of them are equal. Similarly, a secondary node would also hear broadcast messages (binary occupancy vectors) from its neighbors. Based on the vectors a node receives, the node employs a fusion technique to obtain a better estimate about the spectrum usage that can significantly improve the performance of spectrum sensing. 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. P_{attack} also denotes the intensity of attack; hence these terms are used alternatively.

• The nodes need not know the 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 allow us to find distance between the two nodes using received signal strength (RSS) through localization or lateration [7].

• 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 to as T_k, and since it is fixed, its coordinates (x_{T_k}, y_{T_k}) are known to the nodes. A comprehensive table for notations used is tabulated in Table I.

III. Trust Model

Consider Fig. 1. Let O be the position of any node i. Let j be a neighbor whose exact location is not known, but its distance from i is known through RSS localization. On any channel k, the bounds on the received power due to the primary transmitter T_k is given by $[\gamma^j_k]_{high}$ and $[\gamma^j_k]_{low}$. Using commonly used model for RSS [5], we get

$$\gamma^i_k = P_k \times \frac{A^2}{s^{\alpha}_{ik}}; \tag{1}$$

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. It has been shown in [1], that the upper and lower bounds are:

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

$$[\gamma^j_k]_{low} = P_k \times \frac{A^2}{s^{k\ \alpha}_{max_j}}; \tag{3}$$

where the minimum and maximum distances are $s^k_{min_j}$ and $s^k_{max_j}$ respectively (also shown in Fig. 1).

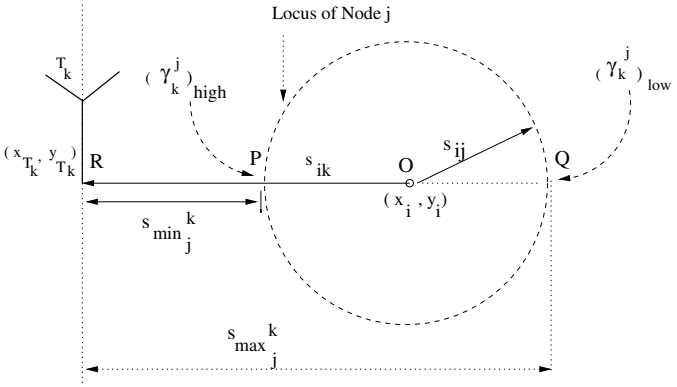

Fig. 1. Bounds of RSS on channel k of neighbor node j

Thus the predicted power vector is given as $P^{ij}_{predict}$= $\big[\ ([\gamma^j_1]_{low}, [\gamma^j_1]_{high}), ([\gamma^j_2]_{low}, [\gamma^j_2]_{high}), \ldots\ldots, ([\gamma^j_n]_{low}, [\gamma^j_n]_{high})\ \big]$.

The inference drawn by node j on channel k is given as

$$d^j_k|_{infer} = \begin{cases} 0 & \text{if } [\gamma^j_k]_{high} \leq \gamma_{th}; \\ 1 & \text{if } [\gamma^j_k]_{low} \geq \gamma_{th}; \\ X & \text{otherwise} \end{cases} \tag{4}$$

TABLE I
Notations

Symbol	Meaning
n	Number of channels
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
TBF^i	Fusion result Based on selective inclusion of j based on trust

where X denotes that no inference could be drawn. Though we discuss with respect to a single node i, the analysis applies to all other nodes as well.

A. Formation of Trust Evidence

The predicted occupancy vector, given the mutual distance between node i and j, is given as

$$D_i^j = [d_1^j|_{infer}, ..., d_n^j|_{infer}]; \quad d_k^j|_{infer} \in 0, 1, X \tag{5}$$

We compare D_i^j with received $B_{adv}^j = [d_1^j, .., d_k^j..d_n^j]$. We define a vector called *Invert Sequence* IS_i^j which records the values of k on which a mismatch (denoted by β) occurs, a match (denoted as α) occurs, and channels with value X in D_i^j are kept as X. If Q^j is the result of the comparison, then

$$Q^j = \begin{cases} \alpha & \text{if } d_k^j|_{infer} = d_k^j; \\ \beta & \text{if } d_k^j|_{infer} \neq d_k^j \text{ and } IS_i^j = k; \\ X & \text{otherwise} \end{cases} \tag{6}$$

The total number of matches, mismatches and undecided for each node j is denoted as $\eta(\alpha^j)$, $\eta(\beta^j)$ and $\eta(X^j)$. We argue that the trust value should be proportional to the number of matches. Also, a similar proportion of the undecided ones must be considered as matches. Thus, the instantaneous value of trust is obtained as:

$$E_{trust_i}^j = \frac{\eta(\alpha^j) * (1 + \frac{\eta(X^j)}{\eta(\alpha^j)+\eta(\beta^j)})}{\eta(\alpha^j) + \eta(\beta^j) + \eta(X^j)} \tag{7}$$

where $0 \leq E_{trust_i}^j \leq 1$. The trust is a value between 0 and 1 which indicates the probability of whether a node is honest or dishonest. A value closer to 1 indicates high trustworthiness and a value closer to 0 indicates low trustworthiness or malicious intent.

B. Trust based Fusion

We use a simple trust based fusion scheme whereby we consider only the neighboring nodes whose $E_{trust_i}^j$ is higher than some trust threshold, Γ_{opt}.

$$\text{If } E_{trust_i}^j \begin{cases} \geq \Gamma_{opt} & \text{Node } j \text{ Trusted ;} \\ < \Gamma_{opt} & \text{Node } j \text{ is not trusted} \end{cases} \tag{8}$$

We define Trust based fusion as $TBF^i = \nabla[TFS_i \oplus B_{act}^i]$ where TFS_i is the trusted fusion set of binary vectors accumulated by node i using equation (8). ∇ is the operator for majority vote decision rule and the $\oplus$ operator is used for combination. The nodes for which $E_{trust_i}^j$ less than the threshold are the non-trusted nodes (NTS_i) in the neighborhood of i. Most approaches as in [1], [2], [6] use an exclusionary policy of disregarding the nodes that are deemed malicious. *We deviate from this notion of filtering out possible outliers*; rather, we take a *more inclusive approach to exploit malicious information* so as to increase the cooperation gain.

IV. Inversion based Fusion Schemas

Our objective is to intelligently invert elements of the occupancy vector that are sent by malicious nodes. That way, we make use of the information sent by malicious nodes.

A. Log Weighted metric based on Trust

First, we need to figure out how much weight do we give to each node. Note, due to the variability in the trust values, we cannot treat all nodes equally. If node $j's$ trust as computed by node i is $E_{trust_i}^j$, we denote its corresponding weight as:

$$W_i^j = log_e[\frac{E_{trust_i}^j}{1 - E_{trust_i}^j}] \tag{9}$$

The above equation ensures negative weights for nodes whose trusts values are below 0.5 and positive otherwise; thus distinguishing the two classes of nodes. Also, the weights monotonically increase for honest nodes and monotonically decrease for malicious nodes as shown in Fig. 2. We use these weights to decide the criterion for inversion.

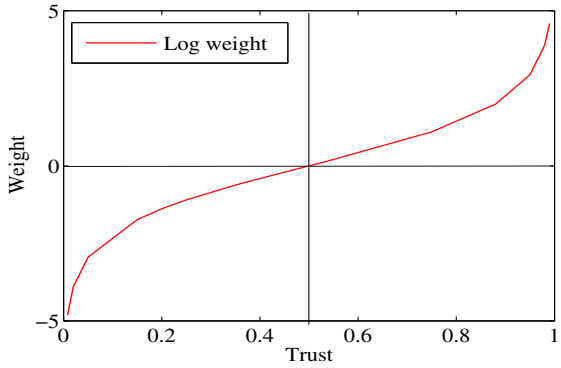

Fig. 2. Relation between Trust and Weight

B. Criterion for Inversion

We define two thresholds: minimum (W_{min}) and optimal (W_{opt}). Nodes for which the weights are more than W_{opt} are considered honest and no change is made to their advertised vectors. Nodes for which the weights are less than W_{min} are deemed malicious; thus every element in their advertised vectors is inverted. The elements in the advertised vectors are selectively inverted for the nodes whose weights lie between W_{min} and W_{opt}. That is,

$$\text{If } W_i^j \begin{cases} > W_{opt} & \text{No Inversion} \\ \leq W_{min} & \text{Complete Inversion} \\ \leq W_{opt} \text{ and } \geq W_{min} & \text{Selective Inversion} \end{cases} \tag{10}$$

The challenge lies in determination of these thresholds– which will be determined through simulations.

1) Complete Inversion (CI): For all malicious neighbors of i (identified by Eqn. (8)), we invert all elements of the advertised vector. In this case, the expected number of channels on which we get the correct opinion is proportional to P_{attack}. For example, if a malicious nodes modifies 80% of the observed data, we get back the actual sensed opinion on 80% of the channels after inversion. Of course, the hind side of complete inversion is that the correct information becomes incorrect. Thus it works better for higher P_{attack}. Hence it is used when weights are very low (i.e., lower than W_{min}). We prove this claim in the simulation section.

2) Selective Inversion (SI): For all neighbors i, we get the invert sequence IS_i^j from the Eqn. (6). IS_i^j indicates all channels with mismatches. For neighbors whose weights lie between W_{opt} and W_{min}, we seek to selectively invert the advertised values in B_{adv}^j for channels comprising the set IS_i^j. Such inversion forms a new vector for all non-trusted neighbors. This scheme is applicable for lower P_{attack}.

The reason for using a combination of two inversion based fusion is because the network does not know P_{attack}. However, the trust weights depend on P_{attack} and as trust changes, the type of inversion scheme that works better is employed.

C. Ideal Fusion and Blind Fusion: For comparisons

Ideal fusion refers to the case when all nodes know and advertise the actual spectrum usage. We measure the deviation (fraction of mismatches) from the ideal result. The lesser the deviation, the better is the performance of the proposed scheme. We use fraction of mismatches from the ideal result as a performance metric We also compare the proposed schemes when there is no defense mechanism and the fusion is done in a blind manner. Later, we show huge improvements of the proposed schemes compared to blind fusion.

V. Simulation Model and Results

We simulate an area of 100x100 units with 100 randomly scattered nodes with 30% of them 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 0.5 to 1.0. It is to be noted that lower attack probability does not significantly affect the network. Transmission range of all nodes is considered to be 20 units.

A. Log Weight Measurement

In Fig. 3, we see a significant difference between the average weights for all honest nodes and the average weight for all malicious nodes. As shown earlier in Fig. 2, the malicious nodes' weights lie on the negative y-axis and the weights of honest node lie on the positive y-axis. As expected, the average weight for the malicious nodes decreases with increasing P_{attack}. We also show how the weight of a single malicious node (Node no. 18) varies with P_{attack}. Obviously, the actions of the honest nodes have nothing to do with P_{attack}; hence a flat line.

B. Optimal threshold for Inversion based Fusion

In trust based fusion, we discarded nodes whose trust was lower than the threshold of 0.50. From Fig. 4, it is evident that the minimum possible mismatch is achieved at 0.0 for all probabilities of attack. As we lower the threshold, more malicious nodes will be included for inversion; hence the higher number of mismatches. We show the results for mismatch fraction over different P_{attack} considering different weight thresholds in Fig. 5. We observe that the lowest mismatch is the one that corresponds to weight threshold of 0. From the above two observations, we can infer that for all P_{attack}, W_{opt} is 0.

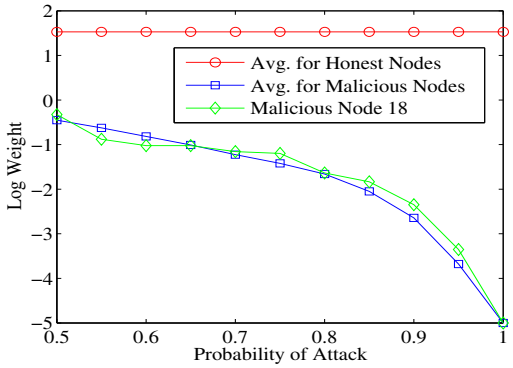

Fig. 3. Weights for malicious and honest nodes

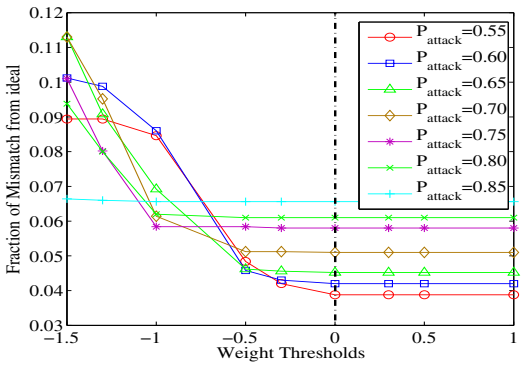

Fig. 4. Fraction of mismatches from ideal vs. weight thresholds

C. Selective Inversion and Complete Inversion

The nodes whose weights are below or equal to 0 are considered potentially malicious node and they are candidates for the inversion schemes. Using the entire range for P_{attack}, we found that for lower values of P_{attack}, selective inversion performs better as evident from Fig. 6. As P_{attack} takes higher values, the undecided channels (i.e., neither matches nor mismatches) are not taken into account in selective inversion and hence the mismatches increase. However, for the complete inversion the reverse happens. As P_{attack} increases, more channels are inverted and the inverted vector from a malicious node is closer to the actual occupancy. This leads to a gain in cooperation even from the malicious nodes. The point where the two inversion schemes SI and CI cross is termed as 'crossover point'. Before the crossover point, selective inversion works better and after the crossover point complete inversion works better. From simulations, we find that the crossover point occurs at $P_{attack} = 0.65$. In the next subsection we justify the choice of crossover point from Fig. 7. With the crossover point known, we are able to back-calculate the threshold W_{min}.

D. Threshold selection for Complete and Selective Inversion

In order to examine the nature of the crossover point, we consider different densities of malicious nodes for both the fusion schemes and see whether there is a consensus on

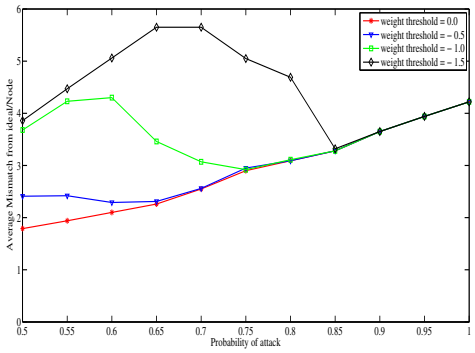

Fig. 5. Fraction of mismatches from ideal vs. probability of attack

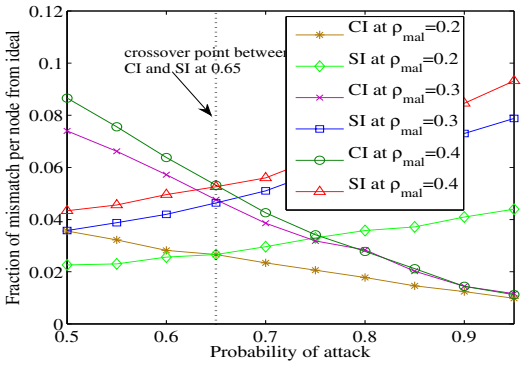

Fig. 7. Crossover point for different malicious node densities

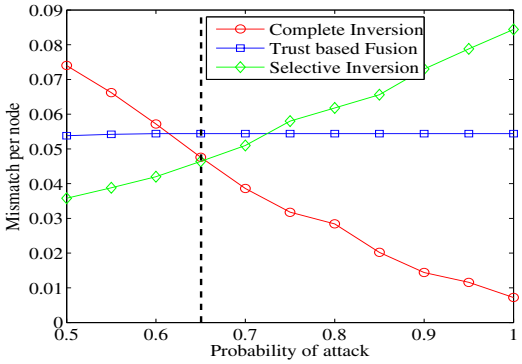

Fig. 6. Comparison of proposed fusion schemes with Trust based Fusion

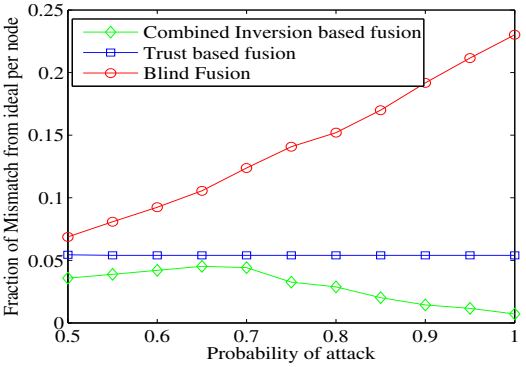

Fig. 8. Combined inversion compared with Trust based and Blind Fusion

the crossover point. Fig. 7 confirms that $P_{attack} = 0.65$ is the cross-over point for both inversion schemes for $\rho_{mal} = 0.2, 0.3$ and 0.4. Though it is obvious to use an inversion scheme based on the probability of attack, the problem is that the regular nodes would not know the probability of attack. However, using the log-weight based trust evaluation, they can compute the weights, w_i, which indirectly captures the probability of attack. Thus, there is a one-to-one correspondence between p_{attack} and w_i^{mal}.

It is interesting to find that the average W_i for malicious nodes that corresponds to $P_{attack} = 0.65$ is almost the same. For instance, for $\rho_{mal} = 0.2, 0.3$ and 0.4 $W_i = -1.023909, -1.009977$, and -1.037043 respectively. We call this $W_{crossover}$ and make $W_{crossover} = W_{min} = -1.0$ the threshold which decides which inversion scheme is to be invoked. Knowing W_{min} and noting that p_{attack} and W_i are inversely related, we simply use selective inversion for $W_i > W_{min}$ and complete inversion for $W_i < W_{min}$. The result of the combined inversion fusion is compared with blind fusion and trust based fusion in Fig. 8. It is apparent that the combined inversion based inclusive technique works better than a single trust based technique.

VI. Conclusions

In this paper, we proposed a technique that utilizes misleading information sent by malicious nodes for the purpose of cooperative sensing in cognitive radio networks. Contrary to common approaches, where information sent by malicious nodes are simply excluded for any decision making, we follow inclusive approaches to exploit even the misleading information. We argue that if the trustworthiness of each malicious node can be computed, then we can appropriately negate the false information. To this end, we use a log-weighted function to compute the trust value of every node. We present two schemes (selective inversion and complete inversion) for inverting the occupancy information of the channels. The combination of these two inversion schemes is also proposed which yields better spectrum occupancy estimates than trust-based and blind fusion schemes for all probabilities of attack. We also find the conditions for which one scheme works better than the other.

References

[1] S. Bhattacharjee, S. Debroy, M. Chatterjee and K. Kwiat "Trust based fusion over Noisy channels through Anomaly Detection in Cognitive Radio Networks", *ACM International Conference on Security of Information and Networks (SIN)*, 2011.

[2] R. Chen, Jung-Min Park, K. Bian, "Robust Distributed Spectrum Sensing in Cognitive Radio Networks", *Proc. IEEE INFOCOM*, 2008.

[3] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio, Part II: Multiuser networks," IEEE Trans. Wireless Commun., pp. 2214-2233, 2007.

[4] A. Ghasemi and E. S. Sousa, Collaborative spectrum sensing for opportunistic access in fading environments, The 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN, Nov. 2005.

[5] S.-W. Jeon, N. Devroye, M. Vu, S.-Y. Chung and V. Tarokh, "Cognitive networks achieve throughput scaling of a homogeneous network", *IEEE Trans. Inform. Theory*, March 2008.

[6] H. Li and Z. Han, "Catching attacker(s) for Collaborative Spectrum Sensing in Cog. Radio Systems: An Abnormality Detection Approach", *IEEE DySpan*, 2010.

[7] www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/wifich2.html