Defending Against Cooperative Attacks in Cooperative Spectrum Sensing

Zhengrui Qin, *Student Member, IEEE,* Qun Li, *Senior Member, IEEE,* and George Hsieh, *Member, IEEE*

***Abstract*—Accurate spectrum sensing is important in cognitive radio networks. False sensing results in either waste of spectrum or harmful interference to primary users. To improve accuracy, cooperative spectrum sensing, in which a set of secondary users cooperatively sense the presence of the primary user, has emerged. This technique, however, opens a window for malicious users and attackers, who may remotely or physically capture the sensors and manipulate the sensing reports. In this paper, we consider the attack model whereby the attacker injects self-consistent false data simultaneously, and propose a modified *COI* (combinatorial optimization identification) algorithm to defend against such attacks. We also provide a theorem that detection uncertainty may exist in cooperative spectrum sensing. We intensively evaluate our algorithm with simulations, and the results show that our algorithm is a good technique to complement an existing algorithm, called *IRIS*.**

***Index Terms*—Bad data identification, cooperative spectrum sensing, combinatorial optimization, cooperative attack.**

I. Introduction

WITH the recent rapid growth of wireless services, spectrum space is becoming increasingly crowded. Within the traditional spectrum regulation framework, spectrum bands are exclusively licensed to specific services (primary users), and no violation from unlicensed services (secondary users) is allowed. As a result, the licensed spectrum is greatly under-utilized temporally and spatially. To solve this dilemma, cognitive radio has emerged to enable secondary users to opportunistically share the licensed spectrum with the help of spectrum sensing methods [1]–[5].

Among the spectrum sensing methods, cooperative spectrum sensing [6]–[8] has shown good performance in improving the accuracy of primary user detection. In cooperative spectrum sensing, a set of secondary users is selected to share the sensing results with each other to make a cooperative decision on the presence/absence of the primary user. However, this opens a window for malicious users and attackers, who may remotely or physically capture some sensors and manipulate their sensing reports simultaneously. As a result, such sensing reports are not trustworthy, nor is the primary user detection decision.

A key challenge is identifying compromised sensing reports under attack and making a detection decision only with uncompromised sensing reports. Min *et al.* [9] has proposed a novel framework called *IRIS* to pinpoint compromised sensing reports by iteratively utilizing the *largest normalized residual method* [10]. *IRIS* works well for most of attack scenarios. In this paper, we introduce an attack model in which an attacker injects some false data simultaneously and the injected false data themselves are self-consistent. Under this specific kind of attacks, a good measurement, instead of a compromised one, may have the largest normalized residual. As a consequence, the *largest normalized residual method*, as well as *IRIS*, may eliminate good measurements while keeping compromised ones. Thus, we argue that *IRIS* may not work well under this kind of attacks, and its performance still has room for improvement. We hence propose a modified *COI* (combinatorial optimization identification) [11] to complement *IRIS*.

Overall, our paper makes the following contributions:

- We design an attack model, called *cooperative attack*, in which an attacker injects self-consistent false data to multiple sensors simultaneously.
- We propose a theorem that the center node of a cognitive radio network may face uncertainties under a cooperative attack, especially in the case when a large portion of sensors are compromised.
- We propose a modified *COI* algorithm to deal with cooperative attacks. Our algorithm is a good scheme to complement *IRIS* for cooperative attacks, and can be flexibly adjusted to fulfill the detection delay requirement.
- We intensively evaluate our algorithm through simulation, with the results validating its performance.

The rest of paper is organized as follows. Section II is related work. Section III describes the system model, the attack model and the problem formulation. In Section IV, we propose our solution. We conduct simulations in Section V and conclude our paper in Section VI.

II. Related Work

Cooperative spectrum sensing has emerged in recent years to improve spectrum sensing accuracy. However, it suffers from various security vulnerabilities. In cooperative spectrum sensing, multiple secondary users can easily alter the detection decision by reporting unreliable or compromised sensing results simultaneously. Thus, it is essential to designing robust cooperative sensing schemes to defend against malicious users, and much effort has already been made in recent years, such as [12]–[19]. In [12], a reputation-based mechanism is proposed to address data falsification, in which reputations of users are used to weigh their sensing reports. In [13], an outlier

Manuscript received April 13, 2012; revised September 4 and December 4, 2012; accepted March 9, 2013. The associate editor coordinating the review of this paper and approving it for publication was Y. Chen.

Z. Qin and Q. Li are with the Department of Computer Science, College of William and Mary, VA 23187 (e-mail: {zhengrui, liqun}@cs.wm.edu).

G. Hsieh is with the Department of Computer Science, Norfolk State University, VA 23504 (e-mail: ghsieh@nsu.edu).

Digital Object Identifier 10.1109/TWC.2013.041913.120516

detection scheme is proposed to filter out extreme values in the sensing data. In [14], the authors convert the area of interest into a grid of square cells and use it to identify and discard the outlier measurements. In [15], an approach based on abnormality detection in data mining is proposed. In [16], the authors present an abnormality detection scheme by using the path-loss exponent in signal propagation. In [17], an approach based on machine learning is proposed, in which an initial trusted set of signal data is used to build a classifier which is subsequently used to detect integrity violations. In [18], a consensus-based scheme is proposed to defend against data falsification, in which the cooperative decision is gradually reached in a distributed manner. In [19], the authors propose to use total error probability to evaluate spectrum sensing accuracy and also propose a weighted sensing framework. Public-key schemes, such as [20], [21], can also be implemented to defend against malicious users.

As mentioned previously, Min *et al.* [9] have developed an attack detection framework in cooperative spectrum sensing, called *IRIS*. They introduce system state estimation to determine the presence/absence of a primary user, in which the system state variables are the transmission power of the primary transmitter and the path-loss exponent. Their insight is that the sensing reports are governed by the network topology and the signal propagation principle. In *IRIS*, they utilize the *largest normalized residual method* [10] to delete abnormal sensing reports iteratively until the norm of measurement residuals drops below a detection threshold or the number of remaining reports hits a protection threshold. Though *IRIS* works well for most of attack scenarios, we show in this paper that it may not work well against *cooperative attacks*, and we propose a modified *COI* to complement *IRIS*.

III. Problem Formulation

In this section, we will first present a centralized cooperative spectrum sensing system. Then we will describe the attack model and formulate our problem.

A. System Model

In a cognitive radio network, both primary users and secondary users coexist in the same geographical area. Since secondary users are allowed to opportunistically access the licensed spectrum originally allocated to primary users, methods to detect the presence/absence of primary users are required. Cooperative spectrum sensing is a robust scheme, in which several secondary users cooperatively determine whether a primary user is using its licensed spectrum or not. In this paper, we consider a centralized model. As shown in Fig. 1, there is a primary user, N secondary users (sensors), and a center node. During each sensing period, sensors measure primary user signals and then send them to the center node using a dedicated control channel. Then the center node will determine the presence/absence of the primary user based on the collected data and broadcast the final decision to all sensors.

According to the path-loss signal propagation model, the received signal strength of the primary user at a cooperative sensor i, P_i, can be modeled as follows:

$$P_i = P_0 + \alpha 10 log_{10}(d_0/d_i) + \epsilon_i,\ i \in [1, N] \qquad (1)$$

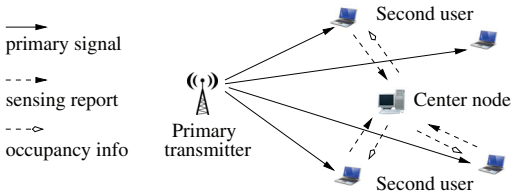

Fig. 1: The centralized model of cooperative spectrum sensing.

where P_0 is the received power at the reference distance d_0, α is the path-loss exponent, d_i is the distance between the primary user and sensor i, and ϵ_i is the measurement error of sensor i.

B. Attack Model: Cooperative Attack

We assume that:

- The attacker can compromise some of the sensors in the network.
- Once a sensor is compromised, the attacker can read and manipulate its sensing report arbitrarily.
- The attacker can manipulate compromised sensors' reports simultaneously.
- The attacker knows a portion of the network topology, i.e., the location of the primary user and those of the sensors he has compromised.

During an attack, the attacker aims to mislead the control node to make a wrong decision. The attacker first compromises some sensors and reads their sensing reports. Based on these sensing reports, the attacker can conclude whether or not the primary user is present. Then the attacker injects false sensing reports into these compromised sensors simultaneously. The attacker's ultimate goal is to mislead the control node into making a decision totally opposite to the real operating conditions. Specifically, when the primary user is present, the attacker will inject false sensing reports as if the primary user is absent; when the primary is absent, the attacker will inject false data as if the primary user is present. To maximize the attack effect, the attacker will inject self-consistent false sensing reports; that is, all injected data are based on a single power level which is inconsistent with the real value. We name this kind of attack *cooperative attacks*. For a cooperative attack, the control center may be easy to detect its presence, but may be difficult to figure out what it is.

C. Problem Formulation

We denote the set of all N sensors in a cognitive radio network by $\mathbf{S} = \{S_1, S2, ..., S_N\}$. During a sensing period, we assume that the sensing reports are $\mathbf{P} = [P_1, P_2, ..., P_N]^T$ before any attack, where T is the matrix transpose operator. Now an attacker has compromise a set of N_{com} sensors, denoted by $\mathbf{S}'$, where $\mathbf{S}' \subset \mathbf{S}$. Without loss of generality, we assume that the compromised sensors are the first N_{com} sensors in $\mathbf{S}$ to simplify the description. Based on the sensing reports of $[P_1, P_2, ..., P_{N_{com}}]^T$, the attacker conducts a linear fitting according to Eq (1) and obtains the primary transmission power P_0, and the path-loss exponent α. Then the attacker will inject false data into the N_{com} compromised sensors as if the two values are P_0' and α', where P_0' tells a scenario opposite to the real one and α' can be any normal value. Suppose that the compromised sensing reports are $\mathbf{P'} = [P_1', P_2', ..., P_{N_{com}}']^T$. The remaining

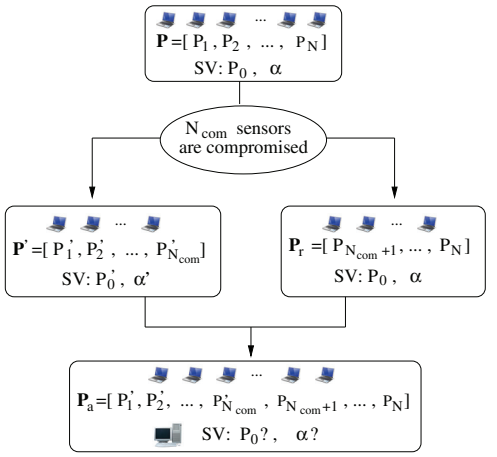

Fig. 2: Attack model and problem formulation. SV stands for state variables.

sensing reports, $\mathbf{P}_r = [P_{N_{com}+1}, ..., P_N]^T$, are intact. Then on the control node side, the collected sensing reports are $\mathbf{P}_a = \mathbf{P'} \cup \mathbf{P}_r = [P'_1, P'_2, ..., P'_{N_{com}}, P_{N_{com}+1}, ..., P_N]^T$. Our problem, illustrated in Fig. 2, is to obtain the real transmission power P_0 from $\mathbf{P}_a$. At the same time, it will be a plus if all compromised sensors can be identified.

IV. Bad Data Identification

In this section, we will propose our solution to the problem formulated in the previous section. Since our algorithm is to complement *IRIS*, we will first review *IRIS* and point out where to enhance *IRIS*. Then we will present a theorem that the center node may face uncertainties under a cooperative attack. Finally we will present our algorithm in detail.

A. IRIS Algorithm [9]

In [9], the authors introduce state estimation that provides a useful approach to detect the presence of attacks and further pinpoint the compromised sensors. In state estimation, the state variables are P_0 and α in Eq(1), denoted by vector $\mathbf{x}$:

$$\mathbf{x} \triangleq [P_0 \quad \alpha]^T \tag{2}$$

Note that $\mathbf{P} = [P_1, P_2, ..., P_N]^T$ is the vector of the primary transmitter's signal strength received by N sensors. Then Eq(1) for N sensors can be expressed in matrix notation as follows:

$$\mathbf{P} = \mathbf{H}\mathbf{x} + \boldsymbol{\epsilon} \tag{3}$$

where

$$\mathbf{H} \triangleq \begin{bmatrix} 1 & 10log_{10}(d_0/d_1) \\ \vdots & \vdots \\ 1 & 10log_{10}(d_0/d_i) \\ \vdots & \vdots \\ 1 & 10log_{10}(d_0/d_N) \end{bmatrix}_{N\times 2} \tag{4}$$

and $\boldsymbol{\epsilon}$ is the vector of measurement errors:

$$\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, ..., \epsilon_N]^T \tag{5}$$

Three statistical estimation criteria are commonly used to obtain the state estimator $\mathbf{x}$: the maximum likelihood criterion, the weighted least-square criterion, and the minimum variance criterion. When measurement errors follow the normal distributed with zero mean, i.e., $\epsilon_i \sim N(0, \sigma_i^2)$, these criteria result in the same solution:

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{P} \tag{6}$$

where $\mathbf{W}$ is a diagonal matrix whose elements are σ_i^{-2}, $1 \leq i \leq N$:

$$\mathbf{W} = \begin{bmatrix} \sigma_1^{-2} & & & \\ & \sigma_2^{-2} & & \\ & & \cdots & \\ & & & \sigma_N^{-2} \end{bmatrix}_{N\times N} \tag{7}$$

After obtaining state estimator $\hat{\mathbf{x}}$, the center node will first examine whether there is an attack by checking the norm of normalized residuals. The normalized residuals are defined as follows:

$$\mathbf{r}_N = \mathbf{D}^{-1/2}\mathbf{r} \tag{8}$$

where

$$D = diag(\mathbf{W}^{-1} - \mathbf{H}(\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T) \tag{9}$$

$$\mathbf{r} = \mathbf{P} - \mathbf{H}\hat{\mathbf{x}} \tag{10}$$

If the L_2 norm $||\mathbf{r}_N||$ is greater than a pre-defined threshold η, the center node will know that some data has been compromised. *IRIS* adopts the *largest normalized residual method* [10] to pinpoint and eliminate the compromised data. It eliminates the data with the largest normalized residual iteratively until the norm of normalized residuals of the remaining data is less than the threshold η or the number of remaining sensors hits a pre-defined threshold N_{min}.

Under a cooperative attack, a good measurement, however, may have the largest normalized residual. The *largest normalized residual method*, as well as *IRIS*, may eliminate good measurements while keeping compromised ones. The ultimate consequence is that the center node may get an incorrect power level for the primary transmitter and make a decision inconsistent with the reality.

Here we give a counter-example to support our arguments. Suppose there are N = 7 cooperative sensors. To make it simple here, we assume there is no measurement error (we do consider measurement errors in the following sections), and we also omit the units. Let d_0 = 1, and $\mathbf{d}$ = $[2, 3, 3, 4, 4, 5, 6]^T$ [1]. Just as in [9], here we adopt the threshold-based rule, in which the determination of whether the primary user is active is based on two thresholds, P_h and P_l. When $P_0 > P_h$, the primary user is active. When $P_0 < P_l$, the primary user is idle. When $P_l \leq P_0 \leq P_h$, secondary users can partially use the licensed spectrum with transmission power control. Let us assume P_h = 4 and P_l = 3. Suppose before any attack, the measurements are $\mathbf{P}$ = $[-7.04, -14.08, -14.08, -19.08, -19.08, -22.96, -26.13]^T$. By Eq(6), we can get $\hat{\mathbf{x}} = [P_0 \quad \alpha]^T = [5, 4]^T$ (here $\mathbf{W}$ is the identity matrix). Therefore, P_0 is greater than P_h, and the primary user is active. Now suppose an attacker has compromised the first four sensors, and he manipulates the measurements on these sensors as if the primary user is idle. After the attack, the measurements are $\mathbf{P}_a$ = $[-10.04, -17.08, -17.08, -22.08, -19.08, -22.96, -26.13]^T$, with the last three measurements

[1] Same d_i does not necessarily mean that the two sensors are at the same location, since d_i is the distance between sensor i and the primary transmitter.

intact. Initially, this data cannot pass the residual test; that is, the norm of normalized residuals is greater than the threshold η. By applying *IRIS*, the measurements from sensors 5, 6, and 7 are eliminated successively. After eliminating these three measurements, the final state vector is $\hat{\mathbf{x}}' = [P_0' \quad \alpha']^T = [2, 4]^T$. Then the center node will conclude that the primary user is idle since P_0' is less than P_l, which is opposite to the reality.

B. Uncertainty Under Cooperative Attacks

Here we take a deep look at the counter example in Section IV-A, which is a cooperative attack. We can divide the seven measurements into two sets, the first four as set A and the last three as set B, and set A is compromised by the attacker. After the attack, set A alone can determine a set of state variables $\hat{\mathbf{x}}_A = [2, 4]^T$; we say set A alone is consistent. Set B can determine another set of state variables $\hat{\mathbf{x}}_B = [5, 4]^T$, which is the same as the original; we also say set B is consistent. That is, the attacker actually mixes two feasible state vectors[2] together. At the same time, the attacker makes *IRIS* eliminate the measurements in set B first rather than in set A. As a consequence, it misleads the center node to make a completely incorrect decision. Since both set A and set B are two feasible sets of measurements, by no means can we favor one set over the other. However, if we know the attacker's capability, we may be able to make a decision. For instance, if we assume that the attacker can at most compromise three measurements, then we know that set B must be compromised. While if we assume that the attacker can compromise four measurements, we still do not know which set is compromised. This raises the question, how many measurements must the attacker compromise to cause such uncertainty? We answer this question by the following theorem.

Theorem 1: For a cooperative sensing system with N sensors that do not have measurement errors, there exists a critical number $N_c = \lfloor N/2 \rfloor - 1$ such that: (1) if the attacker can at most compromise N_c sensors and all d_i are different, then there is only one feasible state vector; (2) otherwise, there may be more than one feasible state vector.

Proof: Since a state vector only has two variables, P_0 and α, any two measurements whose channel gain matrix $\mathbf{H}_{2\times 2}$ has full rank can determine a state vector. Since all d_i are different, any two measurements can determine a state vector. Thus, any two feasible sets of measurements that result in two different state vectors have at most one measurement in common. Suppose N is even (when N is odd, the proof is similar), and $N = 2f + 2$. Now the attacker has compromised $N_c = f$ measurements. There are two cases in total:

(1) These f measurements are consistent and can perfectly determine a state vector different from the original one. Thus, there are two feasible sets of measurements in total. If these two sets have one measurement in common, as shown in the top of Fig. 3, then for the original feasible set, the center node will conclude that there are f compromised measurements; for the compromised feasible set, the center node will conclude that there are $f + 1$ compromised measurements. If these two have no measurement in common, as shown in the middle of Fig. 3, then for the original feasible set, the center node will still conclude that there are f compromised measurements; for the compromised feasible set, the center node will conclude that there are $f + 2$ compromised measurements. Since the attacker can at most compromise f measurements, in either case, the center node can deterministically identify the set of compromised measurements.

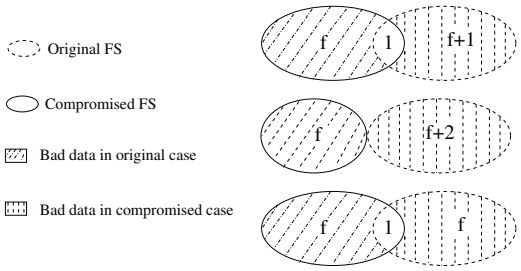

Fig. 3: Illustration of two feasible sets. FS is the abbreviation of feasible set.

(2) These f measurements are not consistent. In this case, for the original feasible set, the center node will conclude that there are f compromised measurements. For any other feasible set, its cardinality is at most f ($f - 1$ of the f compromised measurements plus one of the intact measurements), so the center node will conclude that there are at least $f + 2$ compromised measurements. Again, the center node can identify the set of compromised measurements.

To prove that f is tight, we show by an example that the uncertainty can still exist if f out of $2f+1$ sensors, instead of $2f+2$ sensors, are compromised. In this example, the attacker compromises f measurements, and the original feasible set and the compromised feasible set have one measurement in common, as shown in the bottom of Fig. 3. As a result, the compromised feasible set has $f + 1$ measurements, which are the f compromised measurements plus the measurement in common. The original feasible set also has $f + 1$ measurements. Therefore, for both feasible cases, the center node will conclude that there are f compromised measurements. Thus the center node still does not know which feasible case is the real case. Therefore, f is tight.

In *Theorem 1*, we assume that the sensing reports from uncompromised sensors have no measurement error. When measurement errors do exist, the value of N_c will decrease. In the case without measurement errors, we look for a set of feasible measurements by checking whether the norm of residuals is zero. In the case with measurement errors, however, we check whether the norm of residuals is less than a positive threshold. That is, the check condition is loosened. Therefore, N_c will decrease.

Though it is hard to know in advance how many sensors are compromised, we can estimate how many sensors an attacker can compromise, considering the effort and resource the attacker can take. We do not consider the case that the attacker compromise all or most of the sensors, since in this case there is no way to pinpoint the compromised sensors. As in *IRIS* [9], the authors set a threshold, N_{min}, which is the required minimum number of remaining sensors.

Following the theorem above, we argue that *IRIS* may eliminate data from good sensors instead of bad sensors and get a wrong feasible solution under cooperative attacks,

[2]Here by feasible state vector, we mean a state vector $[P_0, \alpha]$ that is obtained from a subset of total measurements. The subset, with its cardinality maximized, is called a set of feasible measurements accordingly.

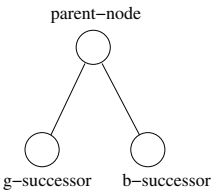

Fig. 4: Successors of a node.

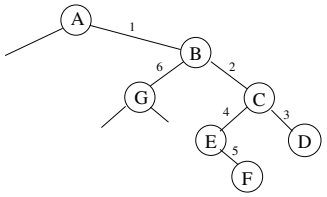

Fig. 5: Branch strategy in the decision tree.

especially when the algorithm hit the threshold N_{min} first. In the following, we will propose a modified version of *Combinatorial Optimization Identification* (COI) [11] to complement *IRIS*.

C. Modified COI

The original *COI* is an approach for identifying multiple instances of bad data in power system state estimation. The essential idea is to construct a partial decision tree using the branch-and-bound method to obtain a feasible solution with the minimum number of bad data. We borrow this idea and make two modifications to fit our problem.

As mentioned above, there may be more than one feasible solution. Therefore, our first modification is to find all feasible solutions instead of only the one with the minimum number of bad data. The second modification is setting a time threshold to meet the time requirement in cooperative spectrum sensing. For instance, in IEEE WRANs, the center node must make a detection decision once every 2 seconds. We will run the branch-and-bound method with increasing bound until hitting the time threshold.

We first illustrate the branch-and-bound strategy, and then present the complete algorithm. The branch-and-bound method will construct a partial decision tree. Since the data with the largest normalized residual is usually more likely to be compromised, after each state estimation run, we pick the sensor with the largest normalized residual as the target node. Each target node has two branches, the right branch representing the case that the sensor is compromised and its left branch representing the other case that the sensor is good, as shown in Fig 4.

The state estimation is conducted at each target node assuming that all undeclared sensors are good. The next sensor to target is the one whose sensing report has the largest normalized residual among all the undeclared sensors. In Fig 4, the b-successor will be the sensor with the largest residual among all undeclared sensors assuming the parent node is compromised, and the g-successor will be the sensor with the largest residual among all undeclared sensors assuming the parent node is good. The key strategy of the tree construction is to first move down towards the right until a feasible solution is reached, and then backtrack to find better solutions. In backtracking, the algorithm stays as far as possible to the right. Let us use an example to illustrate this strategy. As shown in Fig 5, the tree is constructed in the following order:

(1) at the beginning, sensor A has the largest normalized residual, and it becomes the root of the tree; with node A declared bad, run state estimation and node B emerges to have the largest normalized residual; then node B becomes the b-successor of node A;

(2) with node B declared bad, run state estimation and node C has the largest normalized residual; then node C becomes the b-successor of node B;

(3) similar to step (2), construct node D; suppose a feasible solution is found;

(4) backtrack to node C, assume node C is good and run state estimation; node E becomes the g-successor of node C;

(5) construct node F; suppose another feasible solution is found;

(6) backtrack to node B and construct nod G.

The above construction only shows branching. The bounding is taken care of by a heuristic parameter h. In the tree, a g-successor means that one refuses to make a decision that the data with largest normalized residual is compromised, and instead asks for more information about the remaining data. During tree construction, the algorithm keeps a record on how many times a candidate solution takes a g-successor. If a node already has h g-branches between itself and the root, no more g-branches are considered for itself and its successors.

Before we detail our modified *COI*, we define some parameters and functions. We use b_i to denote three states of sensor i, declared bad, declared good, undeclared, with 0, 1, -1 respectively. We use a vector to represent a candidate problem $v = [b_1, b_2, ..., b_N]$; for instance, in the very beginning, the candidate problem is $v = [-1, -1, ..., -1]$. We set the parameter $h = 3$. As pointed out in [11], $h = 2$ is usually enough. We define two functions, $Nzeros(\mathbf{v})$ and $Nones(\mathbf{v})$, which operate on a vector $\mathbf{v}$ and return the number of zero elements in $\mathbf{v}$ and the number of one elements in $\mathbf{v}$ respectively. t_0 is the acceptable time delay for a detection decision, η is the attack detection threshold. The candidate problem with different numbers of g-successors are stored in different stacks. Unless t_0 is used up, the algorithm will first examine candidate problems with no more than one g-successor in stack S_1, then those with 2 g-successors in stack S_2, and finally those with 3 g-successors in stack S_3. Our algorithm is trying to find better feasible solutions than that from *IRIS* without violating the detection delay requirement. The algorithm is detailed in *Algorithm 1*.

After running *Algorithm 1*, we get set F, which contains sets of feasible measurements. Since all of them are feasible, we cannot favor some over others without further information. In practice, we can estimate the attacker's capability, i.e., the maximum number of sensors he can compromise. Suppose the attacker can at most compromise r sensors, then we can filter out some of them by *Algorithm 2*.

D. Discussion

Our modified *COI*, *Algorithm 1*, is complementary to *IRIS*, since its first feasible solution completely relies on *the largest normalized residual method*. The solution from *IRIS* may or may not be in $\mathbf{F}$. If we set $r = N_{min}$, where N_{min} is a threshold in *IRIS*, and *IRIS* hits the threshold η first, then the solution from *IRIS* is definitely in $\mathbf{F}$. Otherwise, it may not be in $\mathbf{F}$. In this sense, our modified *COI* complements *IRIS* in two ways. First, if the solution from *IRIS* is not in $\mathbf{F}$, *IRIS* pinpoints the compromised measurements incorrectly, while our algorithm most likely finds the truth. Second, if the solution from *IRIS* is the one and only one in $\mathbf{F}$, our algorithm

Algorithm 1: Modified *COI*.

Input: $h = 3$, t_0;
Output: Set **F** that contains all sets of feasible measurements.
$t = 0$; $b_i = -1, 1 \le i \le N$; $v = [b_1, b_2, ..., b_N]$;
$S_1 = push(v)$;
while *($S_1 \neq \emptyset$ .or. $S_2 \neq \emptyset$ .or. $S_3 \neq \emptyset$) .and.* $t < t_0$ **do**
  if S_1 *not empty* **then**
    ccp $= S_1.pop$; // **ccp** for **c**urrent **c**andidate **p**roblem;
  else if S_2 *not empty* **then**
    ccp $= S_2.pop$;
  else
    ccp $= S_3.pop$;
  end
  run estimation with data having $b_i \neq 0$ in **ccp**, $i \in [1, N]$;
  calculate $||\mathbf{r}||$ and find index i^* such that P_{i^*} has the largest normalized residual;
  $Cost = Nzeros(\mathbf{ccp})$; // $Cost$ counts how many reports declared bad in **ccp**;
  if $||\boldsymbol{r}|| < \eta$ // *find a feasible solution* **then**
    $CB = Cost$; // CB stands for current best solution;
    $\mathbf{F} = \mathbf{F} \cup \mathbf{ccp}$;
  else
    if $Nones(\boldsymbol{ccp}) < h$ **then** // construct g-successor
      replace i^*th element of **ccp** by 1;
      $j = Nones(\mathbf{ccp})$;
      $S_j.push(\mathbf{ccp})$;
    end
    if $Cost < CB - 1$ **then** // construct b-successor
      replace i^*th element of **ccp** by 0;
      $j = Nones(\mathbf{ccp})$;
      $S_j.push(\mathbf{ccp})$;
    end
  end
  $t = t + processing\ time$;
end

Algorithm 2: Filtering out some feasible solutions.

for *any vector* $\boldsymbol{v} \in F$ **do**
  if $Nzeros(\boldsymbol{v}) > r$ **then**
    $\mathbf{F} = \mathbf{F} \backslash \mathbf{v}$
  end
end

is as good as *IRIS*, while our algorithm reveals the uncertainty if there are more than one solution in **F**. Note that if there are more than one solutions in **F**, the center node has to take all of them into consideration, or has to resort to some verification approaches to narrow **F** down to one feasible solution.

The problem in this paper is purely combinatorial. Our modified *COI* does not scan the decision tree thoroughly, while it only scans the partial tree that most likely contains most of the feasible solutions. To figure out all feasible solutions, one has to use the brute-force search method, which, however, takes too much computational time. The brute-force search method are summarized in $Algorithm$ 3. In $Algorithm$ 3, we assume there are at most r bad measurements and check every possible combination (line 1). The r assumed bad measurements, however, are not necessarily all bad; we re-examine them one by one to see if they are really bad (line 5-9), where ϵ is a pre-defined parameter.

Algorithm 3: The brute-force search algorithm.

Input: r, the attacker's capability;
  P, the set of all sensing measurements;
Output: Set **F** that contains all sets of feasible measurements.
1 **for** *each of* $\binom{N}{r}$ *bad data combinations,* $\boldsymbol{P}_B$ **do**
2 **if** $\boldsymbol{P} \backslash \boldsymbol{P}_B$ *are consistent* **then**
3 $\mathbf{G} = \mathbf{P} \backslash \mathbf{P}_B$; //**G** is the set of good measurements;
4 Calculate the state vector, **x**, based on **G**;
5 **for** *each measurement* $P_i \in \boldsymbol{P}_B$, $1 \le i \le r$ **do**
6 **if** $||P_i - H_{P_i}\boldsymbol{x}|| \le \epsilon$ **then**
7 $\mathbf{G} = \mathbf{G} + P_i$;
8 **end**
9 **end**
10 **end**
11 $\mathbf{F} = \mathbf{F} \cup \mathbf{G}$;
12 **end**

TABLE I: The measurements from sensors before and after the attack.

Sensor i	Distance d_i	$perfect$	$with\ noise$	$compromised$
1	4.46	-113.01	-112.81	no change
2	4.62	-113.62	-113.27	no change
3	4.90	-114.64	-114.42	no change
4	5.08	-115.27	-114.94	no change
5	5.20	-115.68	-115.36	-120.18
6	5.44	-116.46	-116.23	-120.96
7	5.72	-117.33	-117.23	-121.83
8	5.88	-117.81	-117.55	no change

V. Evaluation

In this section, we will evaluate our algorithm by comparing it with *IRIS*. First, we will present a concrete example to show the process of COI, and compare with *IRIS* using different N_{min}. Next, we will set up a simulation for cooperative attacks, and statistically compare the performance between our algorithm and *IRIS*.

A. An Example

In section IV-A, we showed a counter example in which more than half (four out of seven) sensing measurements are compromised by the attacker and there is no measurement error. Here we detail an example with measurement errors and fewer compromised sensors. Our purpose here is to show the process of *COI* and point out why *IRIS* may not work correctly.

System setting: The transmission power P_0 at the reference distance d_0 is 5dB. The radius of the secondary network is about 1km, and the distance between the secondary network and the primary transmitter is 5km. There are eight sensors, whose distances away from the primary transmitter are listed in Table I. When there is no measurement error and the primary user is active, sensors will get perfect reports, which are marked as $perfect$ in Table I. The column $with\ noise$ shows the reports with some random noise. Now suppose an attacker has compromised three sensors, sensors 5, 6, and 7, and changed their values to those listed in column *compromised* in Table I.

By applying our algorithm, it finds four feasible solutions in order, as shown in Table II. By applying *IRIS* with different N_{min}, where N_{min} is the number of required remaining reports after elimination, we get the results shown in III.

Using the data before the attack, we get the original state vector $\mathbf{P} = [P_0, \alpha] = [5.0, 4.0]$. As we can see from Table

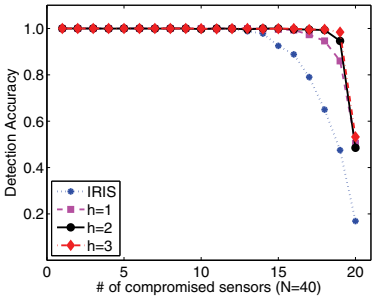

Fig. 6: P_0 detection accuracy comparison.

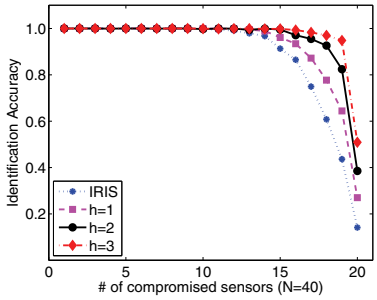

Fig. 7: Identification accuracy comparison.

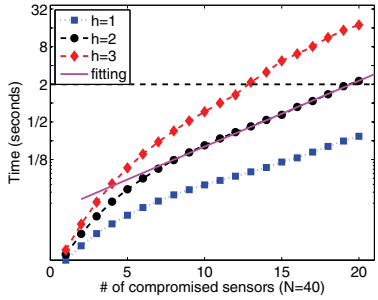

Fig. 8: Processing time.

TABLE II: Feasible solutions from modified *COI*. CS stands for compromised sensors.

order	# of CS	CS	P_0	α
1	6	{2,3,4,5,6,8}	133.4	8.3
2	5	{1,2,3,4,8 }	0.4	4.0
3	4	{5,6,7,8}	1.8	3.9
4	3	{5,6,7}	4.9	4.0

TABLE III: The transmission power with different N_{min}.

N_{min}	Compromised sensors	P_0	α
7	{8}	168.5	9.5
6	{8,4}	171.1	9.6
5	{8,4,3}	162.2	9.3
4	{8,4,3,5}	155.9	9.1
3	{8,4,3,5,6}	143.0	8.7
2	{8,4,3,5,6,2}	133.4	8.3

III, no matter what N_{min} is, *IRIS* eliminates both good measurements and compromised measurements, and the calculated P_0 is far from the original one. However, our algorithm can find the exact correct solution if we assume the attacker can at most compromise three sensors, shown as the 4th feasible solution in Table II. If we assume the attacker can at most compromise four sensors, there will be two feasible solutions, the 3rd and 4th in Table II. In this case, the center node has to rely on some verification approaches to filter out one of them. Nevertheless, our algorithm does not miss the correct solution.

B. Statistical Comparison

In this subsection, we statistically compare the performance of our algorithm with that of *IRIS*. We consider a network with $N = 40$ sensors. The radius of the secondary network is about 2km, and the nodes are randomly distributed inside the circle. The distance between the secondary network and the primary transmitter is 8km. Before any attack, we introduce some random noise to all sensing reports; the noise for each sensing report is less than 1% of the original sensing report. Then we randomly compromise n_{com} of N sensors and inject cooperative bad sensing reports. n_{com} varies from 1 to 20. For each value of n_{com}, we run the simulation $n = 1000$ times and count the following numbers: (1) n_1, the number of times that *IRIS* gets the nearly correct P_0; (2) n_2, the number of times that *IRIS* identifies the right set of compromised sensors and thus gets the correct P_0; (3) n_3, the number of times that our algorithm gets the nearly correct P_0; (4) n_4, the number of times that that our algorithm identifies the right set of compromised sensors and thus gets the correct P_0. For our algorithm, we evaluate three cases with $h = 1$, $h = 2$ and $h = 3$. We set $N_{min} = 20$ in *IRIS* and $r = 20$ in our algorithm.

After obtaining all the counts, we calculate the accuracy ratio using two different criteria. The first criterion is to check the performance of P_0 detection. If the resulting power lever is within the range of $[0.9, 1.1]$ of the real P_0, we count it as a correct detection. The accuracy ratios for IRIS and our algorithm are n_1/n and n_3/n respectively. The second criterion is to check whether all compromised sensors are identified, and the accuracy ratios for IRIS and our algorithm are n_2/n and n_4/n respectively. Also, we examine the time delay of our algorithm. All the results are summarized in Fig 6, Fig 7 and Fig 8.

Fig 6 shows P_0 detection accuracy, and we have the following observations. (1) When $1/3$ of all sensors or less are compromised, *IRIS* and our algorithm work identically and both yield nearly 100% accuracy. (2) When more sensors are compromised, our algorithm demonstrates improved performance over *IRIS* by a noticeable margin. For instance, when $n_{com} = 17$ our algorithm improves *IRIS* by more than 20% for all three h values. (3) When half of all sensors ($N_{com} = 20$) are compromised, our algorithm also performs poorly. The accuracy ratio is only about 50%, regardless of the value of h. This is actually what *Theorem 1* indicates; because there are two feasible solutions with the same number of compromised sensors and our algorithm picks one of them, the ratio is on average 50%. (4) The case with $h = 2$ is almost identical to that with $h = 3$, which coincides with the statement in [11] that $h = 2$ is empirically enough.

Fig 7 illustrates identification accuracy. All ratios for $n_{com} > 13$ are less than those in Fig 6, indicating that for both algorithms they may not identify the correct set of compromised sensors even though they obtain the correct P_0. We also can see that the margin between the case with $h = 1$ and the case with $h = 2$ is larger than that in Fig 6. The margin between the case with $h = 2$ and the case with $h = 3$, however, is still very small, even though it is slightly larger than that in Fig 6.

Fig 8 shows the processing time for different values of h. Since the processing time for *IRIS* is less than 60 milliseconds in our simulation, we do not plot it in the figure. Our simulations are run on an Intel Xeon CPU at 3.07GHz with MATLAB. As we can see, for the case with $h = 1$ and the case with $h = 2$, the processing time is within 2 seconds. For the case with $h = 3$, when $n_{com} \leq N/3$, the time is still within 2 seconds; as n_{com} increases, the time goes up to 18 seconds. We can also see, when $n_{com} \in [N/5, N/2]$, the time increases exponentially, indicated by the linear fitting in the log-scaled figure. In our algorithm, we set a time threshold t_0 and $h = 3$. The algorithm may not finish all three stacks. For instance, if we set $t_0 = 2$ seconds, it may only finish the first

two stacks and terminate during processing the third stack. No matter whether or not it finishes all the stacks, our algorithm will complement *IRIS*, since it starts with *IRIS* solution and then tries to find other possibilities.

VI. Conclusion

In this paper, we aim to complement *IRIS* algorithm in cooperative spectrum sensing. We present a theorem to show uncertainties in cooperative spectrum sensing, and point out that from the view of the center node there may exist multiple feasible solutions under a cooperative attack. We propose a modified *COI* algorithm to improve spectrum sensing performance. Our algorithm can be flexibly adjusted to meet the time delay requirement. We intensively evaluate our algorithm with simulations, and the results show that our algorithm improves the detection accuracy rate compared to *IRIS*.

VII. Acknowledgment

This project was supported in part by US NSF grants CNS-1117412 and CAREER Award CNS-0747108; US Army Research Office under contract No. W911NF-12-1-0081 and US DOE grant No.DE-FG52-09NA29516/A000.

References

[1] J. Mitola III and G. Maguire Jr., "Cognitive radio: making software radios more personal," *IEEE Personal Commun.*, vol. 6, pp. 13–18, 1999.

[2] T. Jing, X. Chen, Y. Huo, and X. Cheng, "Achievable transmission capacity of cognitive mesh networks with different media access control," in *Proc. 2012 IEEE INFOCOM*, pp. 1764–1772.

[3] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Commun. Surveys & Tutorials*, vol. 11, pp. 116–130, 2009.

[4] C. Xin, M. Song, L. Ma, and C.-C. Shen, "Performance analysis of a control-free dynamic spectrum access scheme," *IEEE Trans. Wireless Commun.*, vol. 10, pp. 4316–4323, 2011.

[5] Y. Zhao, M. Song, C. Xin, and M. Wadhwa, "Spectrum sensing based on three-state model to accomplish all-level fairness for co-existing multiple cognitive radio networks," in *IEEE Proc. 2012 INFOCOM*, pp. 1782–1790.

[6] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio networks," in *Proc. 2005 IEEE DySPAN*, pp. 137–143.

[7] S. Mishra, A. Sahai, and R. Brodersen, "Cooperative sensing among cognitive radios," in *Proc. 2006 IEEE International Conf. Commun.*, vol. 4, pp. 1658–1663.

[8] I. Akyildiz, B. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: a survey," *Physical Commun.*, vol. 4, pp. 40–62, 2011.

[9] A. Min, K. Kim, and K. Shin, "Robust cooperative sensing via state estimation in cognitive radio networks," in *Proc. 2011 IEEE DySPAN*, pp. 185–196.

[10] E. Handschin, F. Schweppe, J. Kohlas, and A. Fiechter, "Bad data analysis for power system state estimation," *IEEE Trans. Power Apparatus Syst.*, vol. 94, pp. 329–337, 1975.

[11] A. Monticelli, F. Wu, and M. Yen, "Multiple bad data identification for state estimation by combinatorial optimization," *IEEE Trans. Power Delivery*, vol. 1, pp. 361–369, 1986.

[12] R. Chen, J. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *Proc. 2008 IEEE INFOCOM*, pp. 1876–1884.

[13] P. Kaligineed i, M. Khabbazian, and V. Bhargava, "Malicious user detection in a cognitive radio cooperative sensing system," *IEEE Trans. Wireless Commun.*, vol. 9, pp. 2488–2497, 2010.

[14] O. Fatemieh, R. Chandra, and C. Gunter, "Secure collaborative sensing for crowd sourcing spectrum data in white space networks," in *Proc. 2010 IEEE DySPAN*, pp. 1–12.

[15] H. Li and Z. Han, "Catch me if you can: an abnormality detection approach for collaborative spectrum sensing in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 9, pp. 3554–3565, 2010.

[16] S. Liu, Y. Chen, W. Trappe, and L. Greenstein, "Aldo: an anomaly detection framework for dynamic spectrum access networks," in *Proc. 2009 IEEE INFOCOM*, pp. 675–683, 2009.

[17] O. Fatemieh, A. Farhadi, R. Chandra, and C. Gunter, "Using classification to protect the integrity of spectrum measurements in white space networks," *Proc. NDSS*, vol. 11, 2011.

[18] F. Yu, H. Tang, M. Huang, Z. Li, and P. Mason, "Defense against spectrum sensing data falsification attacks in mobile ad hoc networks with cognitive radios," in *Proc. 2009 IEEE Military Commun. Conf.*, pp. 1–7.

[19] Y. Zhao, M. Song, and C. Xin, "A weighted cooperative spectrum sensing framework for infrastructure-based cognitive radio networks," *Comput. Commun.*, vol. 34, pp. 1510–1517, 2011.

[20] H. Wang and Q. Li, "Efficient implementation of public key cryptosystems on MICAz and TelosB motes," College of William and Mary, Williamsburg, VA, Tech. Rep., Oct. 2005.

[21] H. Wang, B. Sheng, C. C. Tan, and Q. Li, "WM-ECC: an elliptic curve cryptography suite on sensor motes," College of William and Mary, Williamsburg, VA, Tech. Rep., 2007.

Zhengrui Qin received his B.S. in Geophysics Department from Peking University, China, and M.S. in Department of Physics and Astronomy from Dartmouth College. He is currently a Ph.D. student in the Department of Computer Science at the College of William and Mary. His research interests include SmartGrid and Cognitive Radio.

Qun Li received the PhD degree in computer science from Dartmouth College. He is an associate professor in the Department of Computer Science at the College of William and Mary. His research focuses on wireless networks and embedded systems, including pervasive computing, cognitive radio, wireless LANs, mobile ad-hoc networks, sensor networks, and RFID systems. He received the US National Science Foundation (NSF) Career award in 2008. He is a senior member of the IEEE.

George Hsieh is a Professor in the Department of Computer Science at Norfolk State University. Prior to joining NSU in 2002, he worked at Bell Laboratories from 1982 to 2002. George Hsieh received his PhD in Computer Science from Northwestern University in 1982. His research interests include information assurance, information security, wireless communication, and critical infrastructure protection. Dr. Hsieh is a member of IEEE.