Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 3174304, 11 pages
https://doi.org/10.1155/2019/3174304

Research Article

Securing Cooperative Spectrum Sensing against DC-SSDF Attack Using Trust Fluctuation Clustering Analysis in Cognitive Radio Networks

Feng Zhao, Shaoping Li, and Jingyu Feng

Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Correspondence should be addressed to Jingyu Feng; fengjy@xupt.edu.cn

Received 7 October 2018; Accepted 28 January 2019; Published 3 March 2019

Academic Editor: Daojing He

Cooperative spectrum sensing (CSS) has been recognized as a forceful approach to promote the utilization of spectrum bands. Nevertheless, all secondary users (SU) are assumed as honest in CSS, thus giving opportunities for attackers to launch the spectrum sensing data falsification (SSDF) attack. To defend against such attack, many efforts have been made to trust mechanism. In this paper, we argue that securing CSS with only trust mechanism is not enough and report the description of dynamic-collusive SSDF attack (DC-SSDF attack). To escape the detection of trust mechanism, DC-SSDF attackers can maintain high trust by submitting true sensing data dynamically and then fake sensing data in the collaborative manner to increase their attack strength. Noting that the resonance phenomenon may appear in the trust value curve of DC-SSDF attackers, a defense scheme called TFCA is proposed from the design idea of trust fluctuation clustering analysis to suppress DC-SSDF attack. In the TFCA scheme, the decreasing property of trust value in the resonance phenomenon is adopted to measure the similarity distance between two attackers. Based on the similarity distance computation, the binary clustering algorithm is designed by electing initial binary samples to identify DC-SSDF attackers. Finally, trust mechanism can be perfected by TFCA to correct DC-SSDF attackers' trust value. Simulation results show that our TFCA scheme can improve the accuracy of trust value calculation, thus reducing the strength of DC-SSDF attack successfully.

1. Introduction

Currently, spectrum bands are becoming more and more scarce with the rapid development of wireless communication and the huge access demand of IOT devices. However, a large number of the assigned spectra are not utilized efficiently by licensed primary users (PU). According to the Federal Communications Commission (FCC) [1], temporal and geographical variations in the utilization of the assigned spectrum range from 15% to 85%. To solve the contradiction between the spectrum scarcity and low spectrum utilization, it is possible that opportunistic access of the valid spectrum bands should be given to unlicensed secondary users [2]. Without any interference to PUs, cognitive radio (CR) has been considered as an emerging technology that can allow a secondary user (SU) to sense and make efficient use of any available valid PU spectrum bands.

To enhance the detection performance, cooperative spectrum sensing (CSS) [3] has recently received significant attention as a valuable method in CR technology to avoid the case of deep shadowing and multipath fading by exploiting spatial diversity via the sensing results of different SUs. Nevertheless, all secondary users are assumed as honest in CSS, thus giving opportunities for attackers to fake sensing data by SSDF attack [4]. At first, attackers submit false sensing data by the static and individual way. This common SSDF attack can be easily suppressed by trust mechanism. Such attackers will hold a lower trust value when they always submit false sensing data individually. Various trust mechanism studies have been proposed [5–8]. They can estimate whether cooperating SUs are honest or not by their sensing behaviors in the past and then give low weights to malicious SUs or even delete their sensing data when making a final decision.

To escape the detection of trust mechanism, attackers have to change their attack strategies. They can launch SSDF attack with a dynamic manner to maintain high trust value [9] (hereinafter "DSSDF"). In addition, some

attackers conspire with each other to submit false sensing data intentionally [10] (hereinafter "CSSDF"). If there are a sufficient number of CSSDF attackers, conspirators can increase the strength of attack and mislead FC make a wrong final decision. Fortunately, it is easy to crush DSSDF attackers one by one, if they launch DSSDF attack individually. CSSDF attackers can be easily detected with an abnormality detection algorithm by analyzing their highest similarities, if they launch SSDF in the static manner.

In this paper, we report the description of dynamic-collusive SSDF attack (hereinafter "DC-SSDF"), in which the attackers with the dynamic and collaborative manner can not only escape the detection of trust mechanism but also increase the attack strength. From the design idea of trust fluctuation clustering analysis, we propose a defense scheme called TFCA to suppress DC-SSDF attack. The main contributions of this paper are as follows:

(i) Conduct an in-depth investigation on DC-SSDF attack, which is conducted by three attack procedures in a round mode: "Dynamically prompting", "Collaborative attack", and "Self-check". The harmfulness of such attack is great. With high trust value, DC-SSDF attackers can damage the fairness and usability of CSS more easily. A high trust value means that an SU's sensing data can be accepted by CSS. By faking sensing data with together, DC-SSDF attackers with high trust value can deceive honest SUs to interfere with PUs or monopolize primary spectrum bands via CSS.

(ii) Estimate the sensing similarity among SUs from the decreasing property of trust value in the resonance phenomenon. DC-SSDF attackers cooperate together in the "Collaborative attack" phase while prompting their trust value, respectively, in the "Dynamically prompting" phase. So, we can find that the resonance phenomenon may appear in the trust value curve of them. It specially is obvious that DC-SSDF attackers may behave as the sensing similarity related on decreasing trust value in the "Collaborative attack" phase. In the TFCA scheme, we utilize the decreasing property of trust value in the resonance phenomenon to estimate the sensing similarity by measuring the distance between any two SUs, which can avoid mass mathematical analysis and computation.

(iii) Design a binary clustering algorithm to differentiate DC-SSDF attackers and honest SUs. The resonance frequency of the two SUs is recorded in line with their trust value's decreasing property in the resonance phenomenon. Then, DC-SSDF attackers can get the higher resonance frequency among themselves while honest SUs can get the lower resonance frequency among themselves. Based on this, we can use the analysis of the maximum and minimum for the resonance frequency to elect DC-SSDF samples and honest samples rapidly, thus avoiding more iterations of the algorithm.

Table 1: Abbreviations used in this paper.

Abbreviations	Explanation
CSS	Cooperative spectrum sensing
SU	Secondary user
C	Set of Cooperating SUs
SU_i	The i-th SU
PU	Primary user
SSDF	Spectrum sensing data falsification
DC-SSDF	Dynamic-collusive SSDF
TFCA	Trust fluctuation clustering analysis
FC	Fusion center
Φ_1	DC-SSDF samples
Φ_2	Honest samples
Ψ_1	Set of DC-SSDF attackers
Ψ_2	Set of honest SUs

Table 2: Key variables used in this paper.

Variables	Explanation
d	Final decision of FC
d_i	Final decision of SU_i
h_i	Number of true sensing of SU_i
f_i	Number of false sensing of SU_i
t_i^k	Trust value of SU_i at sensing time k
T_i	Trust vector of SU_i
r_{ij}	Resonance frequency between SU_i and SU_j
dis_{ij}	Similar distance between SU_i and SU_j
$R_{n\times n}$	Resonance frequency among cooperating SUs
R_i	Resonance frequency of SU_i corresponding to the other cooperating SUs
ρ_i	Attenuation penalty factor of SU_i at sensing time k

The rest of this paper is as follows. In Section 2, preliminaries related to CSS and trust mechanism are described. DC-SSDF attack is analyzed in Section 3 and the TFCA scheme is designed to defend against it in Section 4. Simulation analysis of DC-SSDF attack and the TFCA scheme is performed in Section 5. Finally, we give the conclusions of this paper in Section 6. In addition, the abbreviations and key variables used in this paper are listed in Tables 1 and 2, respectively.

2. Preliminaries

2.1. Cooperative Spectrum Sensing. A CSS action can be modeled as a parallel fusion network, in which the fusion center (FC) controls the action of CSS, including the process of individual sensing, data reporting, and decision-making, as shown in Figure 1 [3]. Firstly, the method of energy detection is exploited by each SU to individually sense the PU signal through the sensing channel, which is the preselected licensed frequency band for observing the primary spectrum between the PU transmitter and each cooperating SU. Secondly, all individual sensing data are submitted to FC through the reporting channel, which is a control channel for sending

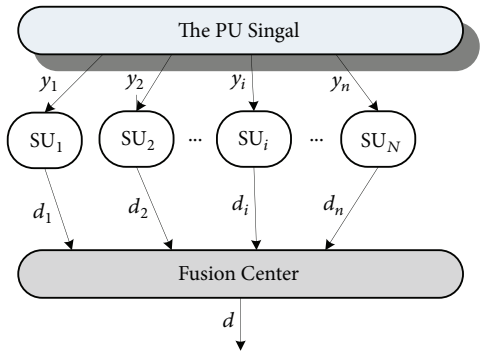

Figure 1: Modeling CSS as a parallel fusion network.

individual sensing information between each cooperating SU and the initiator SU. With the two types of given channels, it can be seen that the CSS process between SUs seems to not waste any more spectrums. Finally, the received individual sensing data are fused by FC to determine the presence of PU. With the "AND", "OR", or "Majority" rule, FC can make the final decision [11].

Specially, individual sensing for PU signal with the energy detection can be abstracted as the "0-1" hypothesis problem [12]:

$$y(t) = \begin{cases} w(t), & H_0 \\ g(t) \cdot p(t) + w(t), & H_1 \end{cases} \quad (1)$$

where $y(t)$ represents the detected PU signal by each SU, $p(t)$ is the transferred PU signal, $g(t)$ is the sensing channel gain, $w(t)$ is the zero-mean additive white Gaussian noise, and t is the sample parameter. H_0 and H_1 denote the hypothesis of absent and present PU signals, respectively. When the estimated $y(t)$ is greater than the energy threshold, the PU signal can be determined as present. Otherwise, no PU signal is detected.

In the process of individual sensing, the sensing data at each SU can be expressed as a binary variable as. For example, d_i indicates the sensing data of SU_i, which is expressed as

$$d_i = \begin{cases} 0, & H_0 \\ 1, & H_1 \end{cases} \quad (2)$$

where "0" and "1" represent the hypothesis of the inexistence and the existence of PU signal, respectively. Correspondingly, FC also make the final decision binary with the "AND", "OR", and "Majority" rule. Under the "AND" rule, the final decision d=1 if all d_i=1. On the contrary, d=1 if one d_i=1 under "OR" rule. The "Majority" rule requires at least a half of SUs to report "1". The "AND" rule works well when the number of cooperating SUs is small, whereas the "OR" rule works best when the number of SUs is large, and the "Majority" rule can be obtained from the k out of N rule under the condition when $k \geq N/2$ [3]. Generally, the "Majority" rule is the best choice to make the final decision, whereas one false sensing data can disturb the decision result of the "AND" and "OR" rule. In Section 5, the simulation of suppressing DC-SSDF attack success ratio is performed to analyze the three fusion rules more clearly.

2.2. Trust Mechanism. Trust mechanism has become more and more significant in many application scenarios, including e-commerce [13], P2P networks [14], internet of things [15], and online social networks [16].

In CSS area, trust mechanism also plays important roles. Typical CSS trust mechanism schemes are as follows. In [5], the authors proposed a trust-aware hybrid spectrum sensing scheme, in which the Beta reputation is employed to calculate trust value. In [6], the authors proposed a reliable CSS scheme with the assistance of trusted SUs to mitigate SSDF attack. In [7], the authors considered the construction of trust mechanism from the perspective of the access competition related to vacant PU spectrum bands. In [8], the authors proposed a trust management scheme by considering multiple decision factors (hereinafter "MFTM"), including (*a*) history-based trust factor, the trust level of an SU during the period of spectrum sensing, (*b*) active factor, the level of activity of an SU in the process of spectrum sensing, (*c*) incentive factor, a reward or incentive for the honest SUs, also serving as a punishment with decrease in trust level for the attackers, and (*d*) consistency factor, the constancy of maintaining a good trust level. The commonality of these existing typical trust schemes is that the trust value of an SU can be calculated by his previous sensing behaviors and the sensing data of malicious SUs should be deleted when making a final decision.

With this commonality, a basic trust mechanism called BTM is abstracted to depict the existing typical trust schemes. Since the sensing data of each SU can be regarded as the "0-1" variable in CSS, it is possible for each SU to conduct two types of sensing behaviors: true and false. In this case, we can calculate the trust value of each SU with two indexes: the number of true sensing behaviors (*tru*) and the number of false sensing behaviors (*fal*). Currently, the beta function is considered as one of the most popular modes using binary input to calculate trust value. It first counts the number of true and false behaviors that a user has conducted and then calculates the trust value with the beta probability density function-$Beta(\alpha, \beta)$ [17].

$$Beta(\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \omega^{\alpha-1} (1-\omega)^{\beta-1} \quad (3)$$

where ω is the probability of sensing behaviors, $0 \leq \omega \leq 1$, $\alpha > 0$, $\beta > 0$.

For instance, h_i and f_i represent the number of true and false sensing behaviors conducted by SU_i. Then, the trust value of SU_i can be calculated as

$$t_i = Beta(h_i + 1, f_i + 1) \quad (4)$$

Note that the case $\Gamma(n) = (n-1)!$ when n is an integer [18]. The expectation value of (4) can be deduced as $E[Beta(\alpha, \beta)] = \alpha/(\alpha + \beta)$. Therefore, t_i can be further calculated as

$$t_i = \frac{1 + h_i}{2 + h_i + f_i} \quad (5)$$

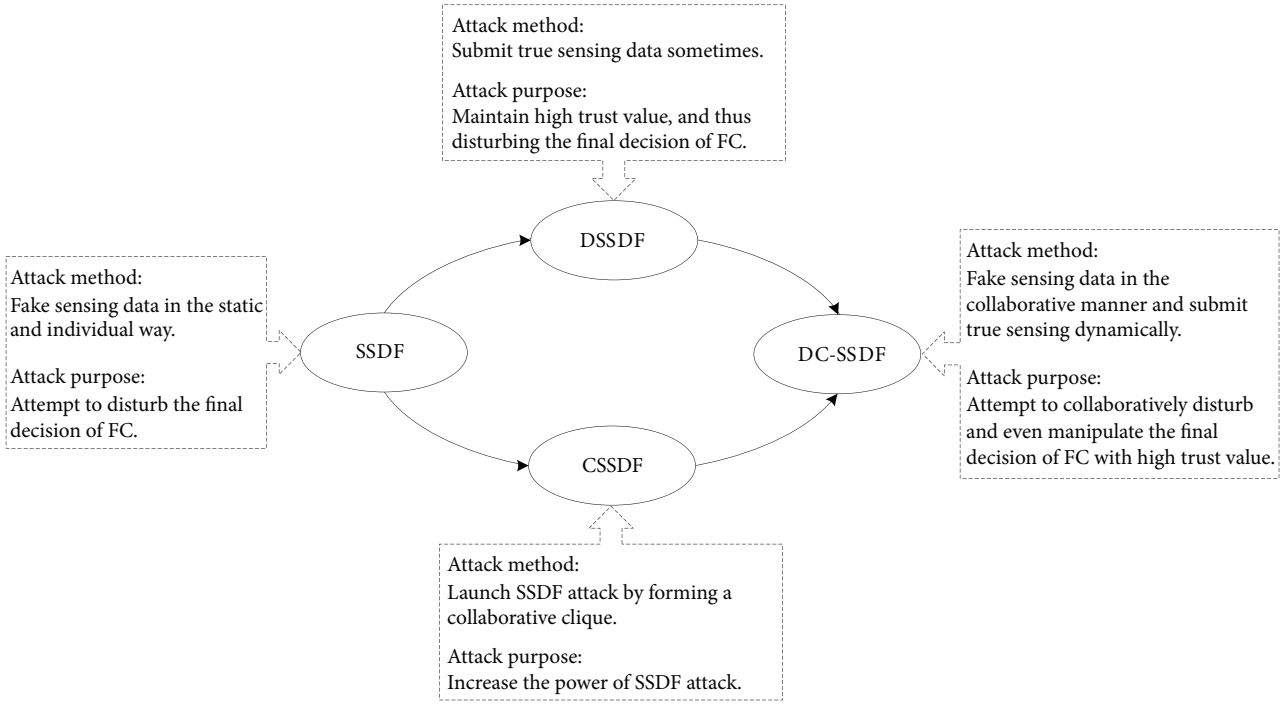

Figure 2: Evolution of SSDF attack.

3. DC-SSDF Attack Overview

Since the sensing data are usually viewed as the "0-1" variable, it is possible for attackers to disturb CSS and launch SSDF attack by submitting false sensing data.

Actually, the basic goal of SSDF attack is to illegally occupy or disturb the PU spectrum bands by manipulating the final decision of FC with using the two patterns [9].

(i) *Always-absent*: some attackers submit false "0" sensing data to show the PU signal is absent, even though some PUs are using their spectrums. As a result, a wrong final decision is made by FC to show that the PU spectrum bands are absent. The intention of such attackers is to give interference to some PUs.

(ii) *Always-present*: some attackers submit false "1" sensing data to declare that the PU signal is present, even though no PU signals are detected. As a result, a wrong final decision is made by FC to show that the PU spectrum bands are present. The intention of such attackers is to monopolize the PU spectrum bands via CSS.

At first, attackers always submit false sensing data individually. That is, such two kinds of SSDF attack patterns are launched in the static and individual way. This original SSDF attack patterns [4] can be easily detected by current trust mechanism such as [5–8], since the original SSDF attackers will get a lower trust value when they always submit false sensing data individually.

In this case, attackers have to change their strategies, thus finding two types of attack modes: DSSDF and CSSDF. For the first attack mode, attackers launch SSDF in a dynamic manner to escape the detection of trust mechanism. They can utilize dynamic behaviors that allow them to maintain high trust value in an alternant process of submitting true and false sensing data [19]. But, it is easy to crush DSSDF attackers one by one, if they launch DSSDF attack individually. To increase the strength of SSDF attack, some attackers form collusion with each other to fake sensing data. They can fake honest SUs' statistical characteristics by collusion when they launch SSDF attack [20]. But, CSSDF attackers may be easily detected with an abnormality detection algorithm by analyzing their highest similarities, if they launch SSDF in the static manner. Except for DSSDF and CSSDF attack, CFF attack is also found in our previous publication [21]. Since the feedback data from initiator SUs are generally unchecked, one of CFF attackers can disguise as an initiator SU who sends the feedback in accordance with the sensing data of their conspirators who play the role of cooperating SUs, resulting in promoting their conspirators' trust value quickly. A two-level defense scheme called FeedGuard from the design ideas of feedback trust and I-C frequency correlation analysis is proposed in [21] to defend against CFF attack.

In this paper, we find that the attackers with the dynamic and collaborative manner can not only escape the detection of trust mechanism but also increase the attack strength. This new SSDF attack mode is named as DC-SSDF in this paper. Obviously, DC-SSDF attack is the latest evolution of SSDF attack, as shown in Figure 2.

Similar to DSSDF attackers, DC-SSDF attackers are extremely sensitive to trust value before launching attack. Assuming SU_i is one of DC-SSDF attackers, he launches DC-SSDF attack under the constraint

$$\delta \le t_i \le \delta + \lambda \tag{6}$$

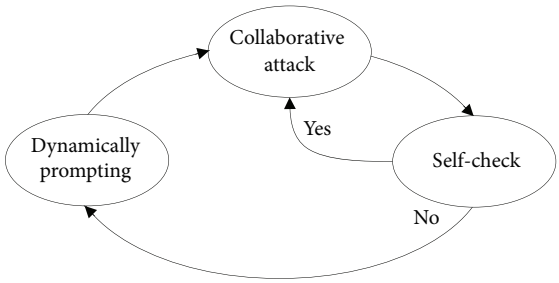

Figure 3: A round of DC-SSDF attack procedure.

where δ is the threshold of trust value and λ ($0 \le \lambda \le 1 - \delta$) is the trust warning line of DC-SSDF attackers.

In (5), $t_i = 0.5$ when $h_i = f_i = 0$. For $t_i \in [0, 1]$, δ can be set to the moderate value 0.5. For $t_i \ge \delta$, SU_i will not be identified by trust mechanism since he is marked as honest. This inspires DC-SSDF attackers to maintain high trust value for themselves. That is, SU_i should maintain his trust value within $[\delta, \delta + \lambda]$. It is late for prompting trust value when $t_i \le \delta$. In this case, SU_i is marked as malicious by trust schemes and anyone will not trust him again. Under the constraint $\delta \le t_i \le \delta + \lambda$, the DC-SSDF attack procedure can be conducted in a round mode including "Collaborative attack"⟶"Self-check"⟶"Dynamically prompting" phases, as shown in Figure 3.

(i) *Dynamically prompting*: DC-SSDF attackers submit true sensing data dynamically to prompt their trust value all by themselves until $t_i \ge \delta + \lambda$.

(ii) *Collaborative attack*: DC-SSDF attackers fake sensing data in the collaborative manner until the half of them cannot maintain $\delta \le t_i \le \delta + \lambda$.

(iii) *Self-check*: Each SU_i self-checks whether $t_i \le \delta$ at the end of each collaborative attack. Yes means continue to the "Collaborative attack" phase. No means go to the "Dynamically prompting" phase.

4. Defending against DC-SSDF Attack Using Trust Fluctuation Clustering Analysis

We capture the core phases, "Dynamically prompting" and "Collaborative attack" of DC-SSDF attack, and then introduce the design idea of Trust Fluctuation Clustering Analysis including trust fluctuation analysis for distance measure and binary clustering analysis to detect DC-SSDF attackers. Meanwhile, the implementation strategies of TFCA are designed to perfect trust mechanism.

4.1. Trust Fluctuation Analysis for Similarity Distance Measure. We have known that the attackers who conduct true or false sensing behaviors alternately would maintain high trust value. Accordingly, the trust value should be calculated at each sensing time. For SU_i, his trust value at sensing time k can be described as t_i^k, and then (7) can be further modified as

$$t_i^k = \frac{1 + h_i^k}{2 + h_i^k + f_i^k} \tag{7}$$

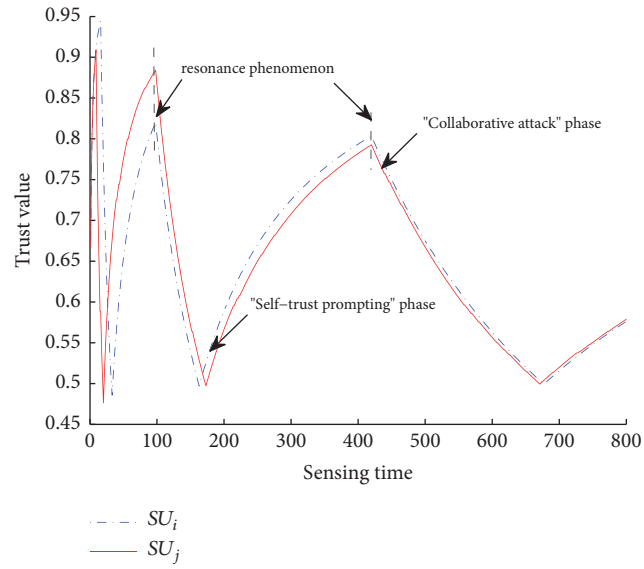

Figure 4: Fluctuation analysis of trust value for any two of DC-SSDF attackers.

where h_i^k and f_i^k represent the true of honest and false sensing behaviors of SU_i at sensing time k. For SU_i, his trust value from the initial sensing time to the current sensing time can form the trust vector $\boldsymbol{T}_i = \{t_i^1, \ldots, t_i^k, \ldots, t_i^h\}$. Without loss of generality, if no CSS action has been involved by SU_i at sensing time k, we can set $t_i^k = t_i^{k-1}$.

When launching DC-SSDF attack, SU_i's trust value will increase in the "Dynamically prompting" phase but decrease in the "Collaborative attack" phase. To further analyze the fluctuation of trust value for any two of DC-SSDF attackers (such as SU_i and SU_j), we perform a simple simulation scenario with a higher value in λ such as 0.4, as shown in Figure 4. It can be found that the two DC-SSDF attackers' trust value fluctuates between δ and $\delta + \lambda$. With the increase of sensing time, the more efforts of "Dynamically prompting" they make, the more opportunities to "Collaborative attack" they will get. Specially, we can find that the resonance phenomenon appears in the trust value curve of them. To depict the similarity of any two DC-SSDF attackers in the process of fluctuation, we can record the resonance frequency (r_{ij}) between the two SUs by Procedure 1. Considering that DC-SSDF attackers cooperate together in the "Collaborative attack" phase while prompting their trust value all by themselves in the "Dynamically prompting" phase, r_{ij} should be recorded when the trust value of SU_i and SU_j decreases at the same time after coming into the "Collaborative attack" phase.

The distance metric is generally useful to measure the similarity among the vectors (such as T_i and T_j). When SU_j is one of DC-SSDF companions of SU_i, they may behave similar to trust fluctuation after several rounds of "Collaborative attack" phase. In the clustering analysis, such similar trust fluctuation can make SU_i and SU_j get a shorter distance. In our TFCA scheme, we should improve the calculation of the distance in line with the characteristic of trust fluctuation between T_i and T_j before designing the binary clustering

Input: T_i, T_j;
Output: r_{ij};
(1) Initialize $r_{ij} = 0$;
(2) **for** $k = 1, k \le \max(|T_i|, |T_j|), k++$ **do**
(3) **if** $t_i^{k-1} > t_i^k \&\& t_j^{k-1} > t_j^k$ **then**
(4) $r_{ij}++$;
(5) **end if**
(6) **end for**

Procedure 1: Record r_{ij} value.

Input: $R_{n\times n}$;
Output: Φ_1 and Φ_2;
(1) Initialize $\Phi_1 = \Phi_2 = \emptyset$;
(2) **for** $i = 1, i \le n, i++$ **do**
(3) $\Phi_1 = \{\text{argmax}(R_i)\} \cup \Phi_1$;
(4) $\Phi_2 = \{\text{argmin}(R_i)\} \cup \Phi_2$;
(5) **end for**

Procedure 2: Elect binary clustering samples.

algorithm. For SU_i and SU_j, their distance can be measured as

$$sd_{ij} = \frac{r_{ij}}{\max\left(\left|T_i\right|, \left|T_j\right|\right)} \sqrt{\sum_{k=1}^{r_{ij}} \left|t_i^k - t_j^k\right|^2} + \frac{\max\left(\left|T_i\right|, \left|T_j\right|\right) - r_{ij}}{\max\left(\left|T_i\right|, \left|T_j\right|\right)} \tag{8}$$

under the constraint

$$t_i^{k-1} > t_i^k \&\& t_j^{k-1} > t_j^k \tag{9}$$

where $|T_i|$ and $|T_j|$ are the number of elements in T_i and T_j, respectively. Obviously, if SU_i and SU_j often launch the "Collaborative attack" phase together, they will get a shorter distance. If not, d_{ij} will be stretched by the second part of (11).

4.2. Binary Clustering Algorithm Design. The ultimate purpose of our TFCA scheme is to detect DC-SSDF attackers. To achieve this goal, the K-means algorithm is a good choice since DC-SSDF attackers show some clustering features in the light of their collaborative behaviours. However, the K-means algorithm [22] cannot be applied directly in detecting DC-SSDF attackers due to the two problems. The one is that the selected sample should be definitely differentiated as a DC-SSDF attacker or an honest SU. So, it is unsuitable to select K samples as the initial mean vectors. Another one is that we utilize the decreasing property of trust value in the resonance phenomenon to measure the distance between two SUs. In this case, if we use the mean of all vectors in a cluster to measure the distance, the decreasing property would be cleared.

In our TFCA scheme, we design a binary clustering algorithm to differentiate DC-SSDF attackers and honest SUs. Firstly, we select two samples as the initial mean vectors by analyzing the resonance frequency of cooperating SUs set (denoted as C) at the current sensing time. For all SUs in C, their r_{ij} value can compose a matrix $R_{n\times n}$ in which n is the number of elements in C.

$$R_{n\times n} = \begin{pmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ r_{i1} & \cdots & r_{ij} & \cdots & r_{in} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ r_{n1} & \cdots & r_{nj} & \cdots & r_{nn} \end{pmatrix} \tag{10}$$

In the matrix, the resonance frequency vector of SU_i corresponding to the other cooperating SUs can be denoted as $R_i = (r_{i1}, \ldots, r_{ij}, \ldots, r_{in})$. It is worth noting that DC-SSDF attackers can get the higher resonance frequency among themselves while honest SUs can get the lower resonance frequency among themselves. We can elect the SUs who hold the maximum from each R_i ($i \in 1, 2, \ldots, n$) to compose the DC-SSDF samples (Φ_1). Meanwhile, we can elect the SUs who hold the minimum from each R_i ($i \in 1, 2, \ldots, n$) to compose the honest samples (Φ_2). This election can be performed by Procedure 2.

Secondly, to avoid clearing the decreasing property of trust value in the resonance phenomenon, we employ the elements of the generated cluster belonging to Φ_1 to calculate the new mean vector at each clustering iteration, rather than using all the elements of the generated cluster.

Finally, the binary clustering algorithm can be designed by Procedure 3 to detect DC-SSDF attackers.

4.3. Perfect Trust Mechanism. When DC-SSDF attackers are detected, typical issues in perfecting trust mechanism focus on (1) reducing their *hon* data with the attenuation penalty factor (ρ) and (2) deleting their sensing data.

For the first issue, it will be difficult for DC-SSDF attackers to maintain high trust value, thus ensuring the accuracy of trust calculation. If SU_i is detected as a DC-SSDF attacker, his penalty factor at sensing time k can be calculated as

$$\rho_i^k = \frac{\text{var}\left(R_i\right)}{\text{var}\left(R_i\right) + \max\left(R_i\right)} \tag{11}$$

where $\text{var}(R_i)$ and $\max(R_i)$ denote the variance and maximum of R_i, respectively. The smaller value of $\text{var}(R_i)$ means that the resonance frequency of SU_i corresponding to the

Intput: C, Φ_1, Φ_2 and $R_{n\times n}$;
Output: the set of DC-SSDF attackers (Ψ_1) and honest SUs (Ψ_2);
(1) Randomly select an element from Φ_1 and Φ_2 respectively and use their trust vector as the initial mean vector $\{\mu_1, \mu_2\}$;
(2) **repeat**
(3) Initialize $\Psi_1 = \Psi_2 = \emptyset$;
(4) **for** $i = 1, i \le |C|, i++$ **do**
(5) Measure the distance d_{ij} between SU_i and μ_j $(1 \le j \le 2)$ with equation (11);
(6) **if** $d_{i1} < d_{i2}$ **then**
(7) $\Psi_1 = \{SU_i\} \cup \Psi_1$;
(8) **else**
(9) $\Psi_2 = \{SU_i\} \cup \Psi_2$;
(10) **end if**
(11) **end for**
(12) **for** $j = 1, j \le 2, j++$ **do**
(13) Calculate the new mean vector $\tilde{\mu}_j = (1/|\Phi_j \cap \Psi_j|) \sum_{SU_k \in (\Phi_j \cap \Psi_j)} \boldsymbol{T}_k$;
(14) **if** $\tilde{\mu}_j \ne \mu$ **then**
(15) Update $\mu = \tilde{\mu}_j$;
(16) **else**
(17) Keep the current mean vector unchanged;
(18) **end if**
(19) **end for**
(20) **until** the current mean vector is not updated again

Procedure 3: Binary clustering algorithm.

Input: C, h_i^{k-1}, $f_i^{k-1}\Psi_1$, Ψ_2 and $R_{n\times n}$;
Output: h_i^k, f_i^k
(1) **for** each $SU_i \in C$ **do**
(2) **if** $SU_i \in \Psi_1$ **then**
(3) $h_i^k = h_i^{k-1} * \dfrac{\mathrm{var}(R_i)}{\mathrm{var}(R_i) + \max(R_i)}$
(4) $f_i^k = f_i^{k-1} + 1$
(5) deleted SU_i's sensing data
(6) **else**
(7) **if** $d_i == d$ **then**
(8) $h_i^k = h_i^{k-1} + 1$
(9) **else**
(10) $f_i^k = f_i^{k-1} + 1$
(11) **end if**
(12) **end if**
(13) **end for**

Procedure 4: Perfect trust mechanism.

other cooperating SUs is more consistent. Thus, h_i^k would be punished by more attenuation, and vice verse. The more $\max(R_i)$ also makes more attenuation to h_i^k.

For the second issue, it would be hard for DC-SSDF attackers to manipulate the final decision again.

Procedure 4 is performed to perfect trust mechanism, in which d_i is the sensing data of SU_i and d is the final decision made by FC.

5. Simulation Results and Discussion

5.1. Simulation Setup. We perform computer simulations with Matlab to validate the performance of the TFCA scheme. The simulation elements are shown in Table 3.

Table 3: Description of simulation elements.

Parameters	Description	Default
N_s	Number of SUs	60
N_p	Number of PUs	5
cycle	Number of cycle simulation	100
round	Rounds of attack	50
p_a	Percentage of attackers	0~50%
λ	Trust warning line	0.4
δ	Threshold of trust value	0.5

The cycle-based fashion is used to perform simulations. At each cycle, some SUs are selected randomly to execute a number of CSS actions by cooperation. Trust mechanism is employed to execute all CSS actions and update the trust value on the corresponding SUs. By several cycles, a trusted CSS network will be gradually formed with trust mechanism.

5.2. Simulation Results. To analyze the simulation result of our TFCA scheme better, we compare it with BTM and MFTM [9].

As we know, an attacker such as SU_i can be detected when $t_i^k < \delta$ at sensing time k. So, the main goal of DC-SSDF attackers is to prompt trust value. To increase the attack strength, SU_i must become a high-trust attacker; i.e., $st_i \ge \delta$. Due to the "Dynamically prompting" phases, DC-SSDF attack can make attackers deviate the actual trust value and cause some network trust errors (*nte*) by forming high-trust attackers. Higher errors mean the lower accuracy in the trust value calculation. *nte* can be specified by

$$nte = \frac{1}{N_s} \sum_{i=1}^{N_s} \sqrt{\frac{1}{t'^k_i} \left(t'^k_i - t^k_i\right)^2} \tag{12}$$

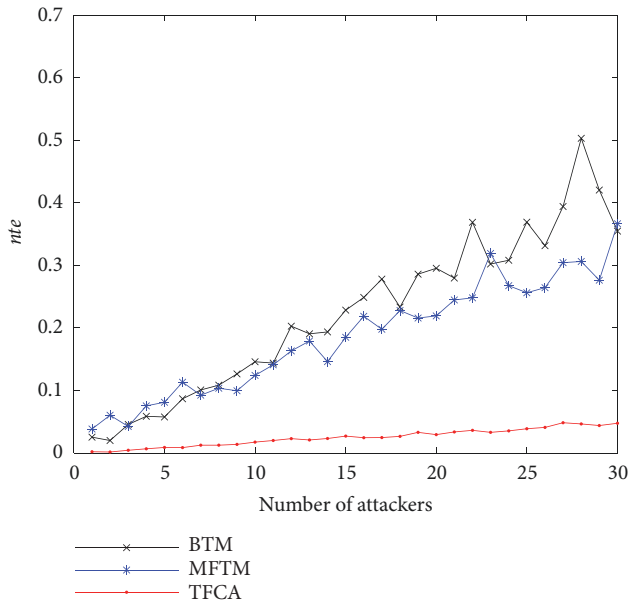

Figure 5: *nte* with the guard of TFCA.

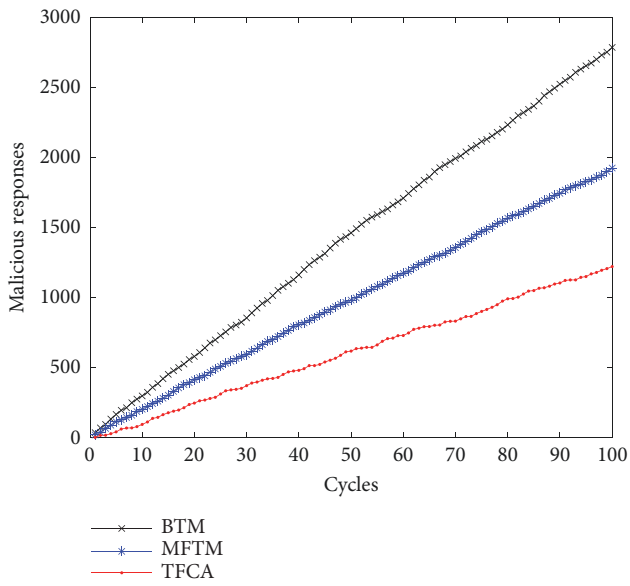

Figure 6: Suppressing malicious responses.

where t'^{k}_{i} and t^{k}_{i} are the actual and measured trust value of SU_i at sensing time k, respectively.

In *nte* simulation, the actual trust value for an attacker is randomly assigned in the interval $(0, \delta]$. As shown in Figure 5, the TFCA scheme is better than BTM and MFTM in reducing *nte*. Without any guard measures, the *nte* curve increases rapidly in the BTM scheme. Although four decision factors are involved in the MFTM scheme to perfect the trust value calculation, it ignores the factor that attackers may prompt trust value by collusion. Consequently, the MFTM scheme also fails to reduce *nte*. By reducing DC-SSDF attackers' *hon* data with the attenuation penalty factor (*ap*), it can be found that *nte* curve with TFCA increases smoothly. Even when the number of DC-SSDF attackers is 30, *nte* of TFCA achieves 0.0475.

Generally, high-trust DC-SSDF attackers submit false sensing data, which would cause a mass of malicious responses at each cycle. The effectiveness of the TFCA scheme can be also validated in terms of reducing malicious responses, as shown in Figure 6. Without any guard measures, DC-SSDF attackers' trust value decreases slowly in the BTM scheme, which can make them get more opportunities to submit false sensing data, resulting in the increase of malicious responses. Since the punishment to trust level for the attackers is considered in the MFTM scheme, they get less attack chances. So, the MFTM scheme is better than BTM. In the TFCA scheme, the identified DC-SSDF attackers have no right to request CSS since their trust value can be attenuated to below δ. Then, it is difficult to prompt their conspirators' trust value again, thus suppressing malicious responses more effectively.

We also analyze the performance of our TFCA scheme in terms of attack success ratio. This simulation is performed at the always-absent and always-present attack patterns.

It can be found that the TFCA scheme is also better in suppressing attack success ratio than BTM and MFTM under the "OR" rule at the always-absent attack pattern and the "AND" rule at always-present attack pattern, as shown in Figure 7. At always-present attack pattern, the damage of attacks is the biggest under the "OR" rule only when one false "1" data can make the final decision as "1". At always-absent attack pattern, the damage of attacks is the biggest under the "AND" rule only when one false "0" data can make the final decision as "0". Under the "Majority" rule, the TFCA scheme can reduce attack success ratio to some extent with using the binary clustering algorithm. Although the damage of attacks against the "Majority" rule amplifies with the number of attackers, the TFCA scheme is better in suppressing attack success ratio better than BTM and MFTM at the always-absent and always-present attack patterns. The reason is that the majority of sensing data are "1" or "0" under the "Majority" rule; the final decision will be "1" or "0". According to this simulation analysis, we can validate that the "Majority" rule is the best choice to make the final decision in CSS.

We have known that the binary clustering algorithm can make our TFCA scheme suppress DC-SSDF attack. Another question is how about the convergence of the binary clustering algorithm. To validate the binary clustering algorithm better, we compare it with the K-means algorithm by observing the convergence of the number of DC-SSDF attackers. As shown in Figure 8, the binary clustering algorithm begins to converge after 9 iterations when the ratio of attackers is 20%, whereas the K-means algorithm begins to converge after 15 iterations. In the binary clustering algorithm, we can analyze the maximum and minimum of the resonance frequency to definitely differentiate the selected samples as DC-SSDF samples and honest samples. Then, we can employ the elements of the generated cluster who also belong to the DC-SSDF samples to calculate the new mean vector at each clustering iteration. In the K-means algorithm, samples are selected randomly, and then all the elements

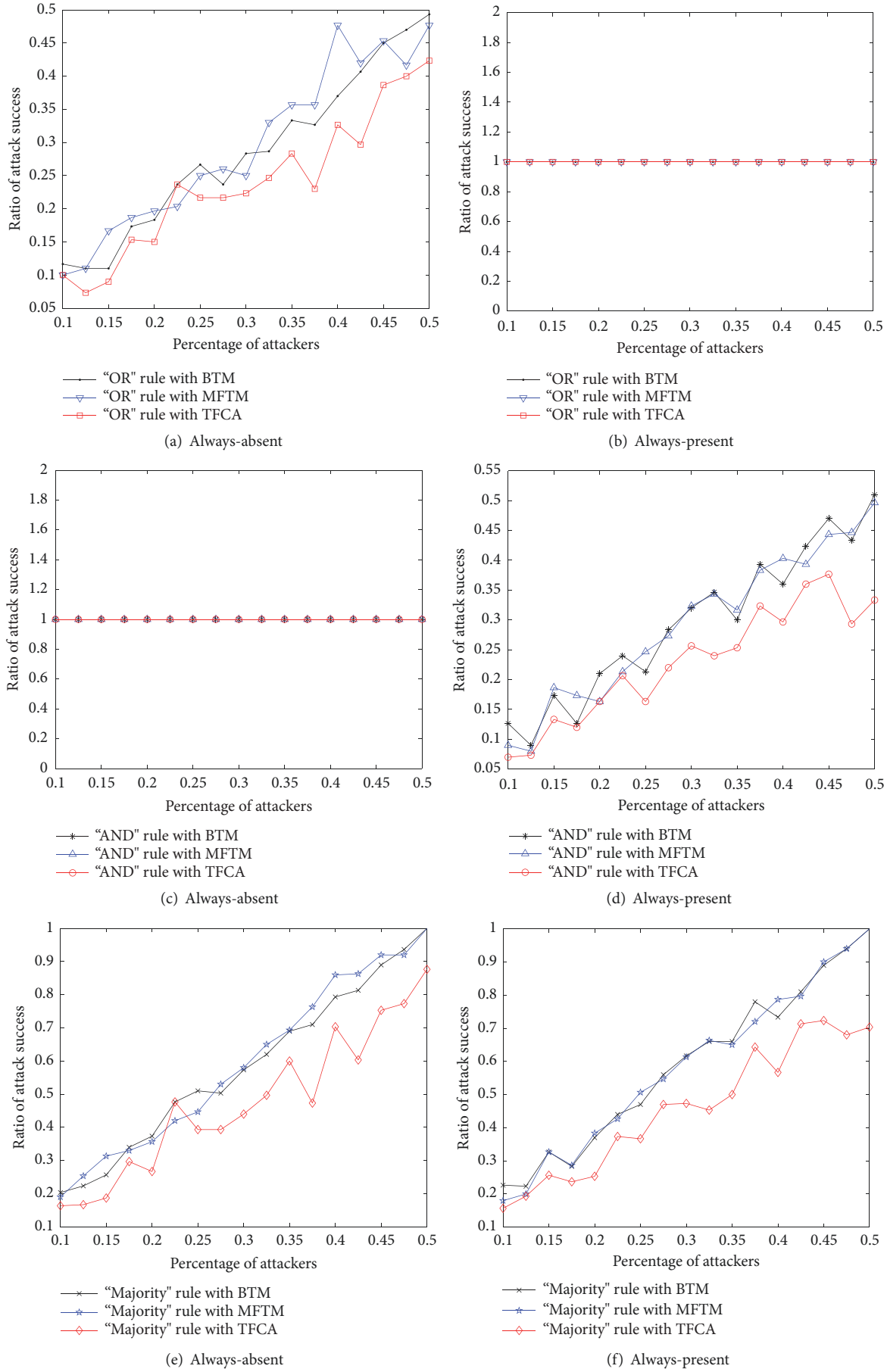

Figure 7: Suppressing DC-SSDF attack success ratio.

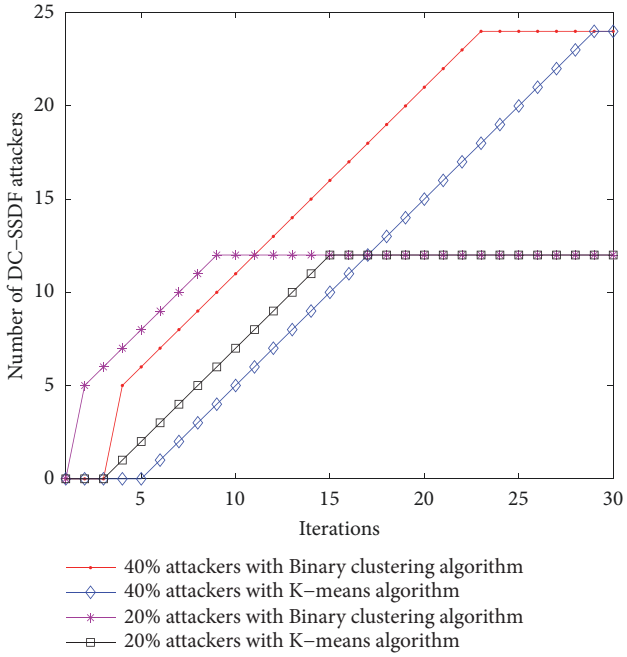

Figure 8: Convergence of the binary clustering algorithm.

of the generated cluster to calculate the new mean vector at each clustering iteration. Therefore, the binary clustering algorithm also converges faster than the K-means algorithm, even though the ratio of attackers becomes 40%.

6. Conclusions

We report the description of DC-SSDF attack and present the TFCA scheme to defend against this attack in this paper. The TFCA scheme is designed in three successive stages: trust fluctuation analysis, binary clustering algorithm design, and perfect trust mechanism, in which trust fluctuation clustering analysis is introduced to construct the TFCA scheme since the resonance phenomenon may appear in the trust value curve of DC-SSDF attackers. Simulation results show that our TFCA scheme can ensure the accuracy of trust value calculation and suppress DC-SSDF attack success ratio to some extent.

Data Availability

We perform computer simulations with Matlab to validate the performance of the proposed scheme. No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by the support plan for innovation ability project of Shaanxi Province under Grant 2017KCT-30-02 and the New Star Team of X'an University of Posts & Telecommunications.

References

[1] Federal Communications Commission, "Spectrum policy task force," Rep. ET Docket 02-135, 2002, https://transition.fcc.gov/sptf/files/SEWGFinalReport_1.pdf.

[2] J. Mitola, *Cognitive radio: An integrated agent architecture for software defined radio [Ph.D. thesis]*, Royal Instutute of Technology (KTH), Stockholm, Sweden, 2000.

[3] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: a survey," *Physical Communication*, vol. 4, no. 1, pp. 40–62, 2011.

[4] R. Chen, J.-M. Park, Y. T. Hou, and J. H. Reed, "Toward secure distributed spectrum sensing in cognitive radio networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 50–55, 2008.

[5] T. Qin, H. Yu, C. Leung, Z. Shen, and C. Miao, "Towards a trust-aware cognitive radio architecture," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 2, pp. 86–95, 2009.

[6] K. Zeng, Q. Peng, and Y. Tang, "Mitigating spectrum sensing data falsification attacks in hard-decision combining cooperative spectrum sensing," *Science China Information Sciences*, vol. 57, no. 4, pp. 1–9, 2014.

[7] J. Feng, G. Lu, and H. Chang, "Behave well: How to win a pop vacant band via cooperative spectrum sensing," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 2, pp. 1321–1336, 2015.

[8] S. Kar, S. Sethi, and R. K. Sahoo, "A multi-factor trust management scheme for secure spectrum sensing in cognitive radio networks," *Wireless Personal Communications*, vol. 97, no. 2, pp. 2523–2540, 2017.

[9] H. Rifà-Pous, M. J. Blasco, and C. Garrigues, "Review of robust cooperative spectrum sensing techniques for cognitive radio networks," *Wireless Personal Communications*, vol. 67, no. 2, pp. 175–198, 2012.

[10] J. Feng, G. Lu, Y. Zhang, and H. Wang, "Avoiding monopolization: mutual-aid collusive attack detection in cooperative spectrum sensing," *Science China Information Sciences*, vol. 60, no. 5, pp. 1–3, 2017.

[11] E. C. Y. Peh, Y. Liang, Y. L. Guan, and Y. Zeng, "Optimization of cooperative sensing in cognitive radio networks: a sensing-throughput tradeoff view," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 5294–5299, 2009.

[12] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, "CRAHNs: cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 810–836, 2009.

[13] M. A. Morid and M. Shajari, "An enhanced e-commerce trust model for community based centralized systems," *Electronic Commerce Research*, vol. 12, no. 4, pp. 409–427, 2012.

[14] X. Li, F. Zhou, and X. Yang, "Scalable feedback aggregating (SFA) overlay for large-scale P2P trust management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1944–1957, 2012.

[15] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *Journal of Network and Computer Applications*, vol. 42, pp. 120–134, 2014.

[16] M. Li, Y. Xiang, B. Zhang, Z. Huang, and J. Zhang, "A trust evaluation scheme for complex links in a social network: a link

strength perspective," *Applied Intelligence*, vol. 44, no. 4, pp. 969–987, 2016.

[17] A. Jøsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th Bled Electronic Commence Conference*, pp. 1–14, June 2002.

[18] Wikipedia, "Gamma function," August 2016, http://en.wikipedia.org/wiki/Gammafunction.

[19] F. Zhao and J. Feng, "Supporting trusted soft decision scheme using volatility decay in cooperative spectrum sensing," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 5, pp. 2067–2080, 2016.

[20] M. Wang, B. Liu, and C. Zhang, "Detection of collaborative SSDF attacks using abnormality detection algorithm in cognitive radio networks," in *Proceedings of the 2013 IEEE International Conference on Communications Workshops, ICC 2013*, pp. 342–346, Hungary, June 2013.

[21] J. Feng, S. Li, S. Lv et al., "Securing cooperative spectrum sensing against collusive false feedback attack in cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8276–8287, 2018.

[22] Wikipedia, "k-means clustering," https://en.wikipedia.org/wiki/K-means_clustering.