sensors

MDPI

Article

Securing Cooperative Spectrum Sensing Against Collusive SSDF Attack using XOR Distance Analysis in Cognitive Radio Networks

Jingyu Feng [1,†], Man Zhang [1,†], Yun Xiao [2,*,†] and Hongzhou Yue [3,†]

[1] Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China; fengjy@xupt.edu.cn (J.F.); ZhangMan3418@163.com (M.Z.)

[2] School of Information Science and Technology, Northwest University, Xi'an 710127, China

[3] Information Security Research Center of State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China; yuehz@nipc.org.cn

* Correspondence: yxiao@nwu.edu.cn; Tel.: +86-134-8814-1618

† These authors contributed equally to this work.

Received: 12 November 2017; Accepted: 24 January 2018; Published: 27 January 2018

Abstract: Cooperative spectrum sensing (CSS) is considered as a powerful approach to improve the utilization of scarce spectrum resources. However, if CSS assumes that all secondary users (SU) are honest, it may offer opportunities for attackers to conduct a spectrum sensing data falsification (SSDF) attack. To suppress such a threat, recent efforts have been made to develop trust mechanisms. Currently, some attackers can collude with each other to form a collusive clique, and thus not only increase the power of SSDF attack but also avoid the detection of a trust mechanism. Noting the duality of sensing data, we propose a defense scheme called XDA from the perspective of XOR distance analysis to suppress a collusive SSDF attack. In the XDA scheme, the XOR distance calculation in line with the type of "0" and "1" historical sensing data is used to measure the similarity between any two SUs. Noting that collusive SSDF attackers hold high trust value and the minimum XOR distance, the algorithm to detect collusive SSDF attackers is designed. Meanwhile, the XDA scheme can perfect the trust mechanism to correct collusive SSDF attackers' trust value. Simulation results show that the XDA scheme can enhance the accuracy of trust evaluation, and thus successfully reduce the power of collusive SSDF attack against CSS.

Keywords: cooperative spectrum sensing; cognitive radio; trust; collusive attack; network security

1. Introduction

In a recent study made by the Federal Communications Commission, it was found that most of the licensed radio frequency spectrum is not efficiently utilized by the primary users [1]. In order to improve spectrum utilization, it has been suggested that opportunistic access to the spectrum should be given to secondary users [2]. Cognitive Radio is an emerging technology that would allow an secondary user (SU) to sense and use any available valid spectrum from primary users (PU) at a given time.

To avoid the case of deep shadowing and multipath fading, cooperative spectrum sensing (CSS) [3–5] has been considered as a viable method to enhance the detection performance by exploiting spatial diversity via the observations of spatially located SUs. However, CSS is often established randomly among SUs that are unrelated and unknown with each other [6], and thus offering opportunities for attackers to manipulate sensing data by launching SSDF attack [7]. Such SSDF attack pattern can be launched by two ways: individual or collusive. Compared with collusive attack, individual attack is less harmful and can be suppressed. In the collusive pattern, the attackers who collude with each other to form a collusive clique can increase the power of SSDF attack and fake the sensing data intentionally. If there are the adequate attackers, a collusive clique can lead to a wrong spectrum sensing decision.

To suppress SSDF attack, various trust mechanism studies have been presented [8–11]. They evaluate whether an SU is trusted or not by his historical sensing behaviors and give the low weights to less trusted SUs or even delete their sensing data when making a final decision. Nevertheless, collusive SSDF attackers can improve their trust value with the help of each other, except for increasing the attack power. Therefore, they may bypass the detection of the trust mechanism.

In this paper, we analyze the characteristics of collusive SSDF attack and present a defense scheme called XDA to suppress such an attack. The main contributions of this paper are as follow:

- Analyze the three types of threats of collusive SSDF attackers in detail. The first one is profit-driven, in which collusive SSDF attackers conspire with each other to falsify the sensing data inspired by some profits such as monopolizing vacant PU spectrums. The second is manipulating the trust mechanism, in which collusive SSDF attackers can improve their trust value quickly. The third is disturbing data fusion, in which collusive SSDF attackers can submit their false sensing data to disturb the data fusion of FC successfully with high trust value.
- Noting the duality of sensing data submitted by collusive SSDF attackers, the XOR distance calculation in line with the type of "0" and "1" historical sensing data is introduced. Based on the fact that collusive SSDF attackers hold the the lower XOR distance and high trust value simultaneously, a lightweight algorithm to detect collusive SSDF attackers is designed.
- Enhance the accuracy of trust evaluation. By reducing the increase of "the number of honest sensing data" with XDA, collusive SSDF attackers will not get a high trust value again. As a result, they can be detected by a trust mechanism.

The remainder of this paper is organised as follows: In Section 2, preliminaries related on CSS and trust mechanism are described. We analyze collusive SSDF attack and constructs the XDA scheme to suppress it in Section 3. Simulation analysis of the XDA scheme is performed in Section 4. Finally, we conclude this paper in Section 5.

2. Preliminaries

2.1. Cooperative Spectrum Sensing

The CSS process can be modeled as a parallel fusion network [10]. As shown in Figure 1, a central identity called fusion center (FC) controls the process of CSS: individual sensing, data reporting and data fusion [12]. First, each SU exploits the energy detection to sense the signal of a PU via the sensing channel which is the selected licensed frequency band where a physical point-to-point link between the PU transmitter and each SU for observing the primary spectrum. Second, all SUs report their sensing data to FC via the reporting channel, which is a control channel with a physical point-to-point link between each SU and FC for sending individual sensing information. Finally, FC fuses the received individual sensing data as a single decision value to determine the presence of PU. Such a final decision can be made according to three typical data fusion rules: the "AND", "OR" and "Majority" rule [13].

Typically, the energy detection method is used to detect a PU signal, and thus the individual sensing for energy detection method can be depicted as a binary hypothesis problem [14]:

$$y(t) = \begin{cases} n(t), & H_0 \\ h(t) \cdot s(t) + n(t), & H_1 \end{cases} \tag{1}$$

where $y(t)$ is the sensed signal at each SU, $s(t)$ is the transferred PU signal, $h(t)$ is the channel gain of the sensing channel, $n(t)$ is the zero-mean additive white Gaussian noise, and t is the sample index. H_0 and H_1 represent the hypothesis of the inexistence and the existence of the PU signal, respectively.

If $y(t)$ is larger than the decision threshold of energy detection, the existence of PU can be signal declared. Otherwise, no PU signal is detected.

After the individual sensing, the sensing data of each SU is determined. d_i represents the sensing data of SU_i, which is generally described as a binary variable:

$$d_i = \begin{cases} 0, H_0 \\ 1, H_1 \end{cases} \tag{2}$$

where "0" and "1" indicate the hypothesis of the inexistence and existence of the PU spectrum status, respectively. Correspondingly, the final decision of FC is also binary via the data fusion. In CSS, the common methods of the data fusion are the "AND", "OR" and "Majority" rule. In the "AND" rule, FC makes d = 1 when all d_i = 1. The "OR" rule refers to d = 1 when one d_i = 1. The "Majority" rule requires at least half of SUs to report "1". The "OR" rule works the best if the number of SUs is large, whereas the "AND" rule performs well if the number of cooperating SUs is small, and the "Majority" rule can be derived from the k out of N rule under the case $k \geq N/2$ [3].

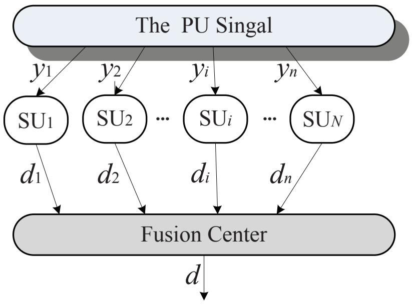

Figure 1. Modeling CSS as a parallel fusion network.

2.2. Trust Mechanism

Trust mechanism has been widely used in many application scenarios, including e-commerce [15], P2P network [16], ad hoc network [17], online social communities [18], etc.

Currently, the trust mechanism also plays a significant role in the CSS area. Representative trust mechanism schemes in CSS are as follows. In [8], the authors propose a trust-aware hybrid spectrum sensing scheme with the Beta reputation. However, they make the base station provide its sensing result for the trust weighted sensing result aggregation at each CSS action, which will cause a heavy overload to the base station if it also needs to detect the PU singal at each CSS action. Zeng et al. propose a secure CSS scheme with the assistance of trusted SUs to mitigate SSDF attack in [9], but a great deal of probability analysis used to identify attackers can lead to the increase of the computation complexity. In [10], the authors use a reputation based method called weighted sequential probability ratio test, but this method needs the location of PU and SUs to obtain some required prior probabilities and requires large number of samples and in worst case may lead to a deadlock situation with an endless sensing sampling. In [11], the authors propose a multi-factor trust management scheme by involving multiple decision factors, including the history-based trust factor, active factor, incentive factor and consistency factor, but the evaluation of these factors has caused more mathematical computation.

To avoid the heavy overload and suppress collusive SSDF attack simultaneously, we design a lightweight detection algorithm of collusive SSDF attackers in the basis of reducing the mathematical complexity of trust mechanism. Noting that the sensing data from SUs can be viewed as a binary variable ("0" or "1"), it is easy for them to produce two types of sensing results: honest or false. Based on the binary variable, the design idea of XOR distance analysis can be introduced to suppress collusive SSDF attack. Obviously, the fast XOR operation built on the "0" and "1" sensing data can make the detection algorithm lightweight, in which the SUs with the lower XOR distance and high trust value will be detected as collusive SSDF attackers.

Based on the binary variables, we can also abstract a simple trust mechanism scheme called Baseline, in which the trust value of each SU can be initialized by two indexes: the number of honest sensing (*hon*) and the number of false sensing (*fal*). After detecting collusive SSDF attackers, the Baseline can be perfected by preventing the increase of *hon*.

Generally, the beta function is one of the most popular designs using binary input (i.e., positive or negative) to evaluate trust value [8]. It counts the number of positive and negative behaviors a user has performed, and then calculates the trust value with beta probability density function denoted by Beta(α, β) [19].

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

where θ is the probability of behaviors, $0 \leq \theta \leq 1$, $\alpha > 0$, $\beta > 0$.

Take the *i*-th SU (SU$_i$) as an example, hon_i and fal_i denote the number of honest sensing (positive) and false sensing (negative) conducted by SU$_i$. In the Baseline, the trust value of SU$_i$ can be calculated as

$$T_i = Beta(hon_i + 1, fal_i + 1) \tag{4}$$

Consider the condition $\Gamma(n) = (n-1)!$ when n is an integer [20]. Thus, the expectation value of the beta function is $E[Beta(\alpha,\beta)] = \alpha/(\alpha+\beta)$. In this case, T_i can be further calculated as

$$T_i = \frac{1 + hon_i}{2 + hon_i + fal_i} \tag{5}$$

Let δ denote the threshold of trust value. For $T_i \geq \delta$, SU_i will be identified as an attacker, and vice versa. In order to guarantee the performance of CSS, δ should satisfy two requirements: (1) δ should be a rational value between 0 and 1 as $T_i \in [0,1]$; (2) the value of δ can be adjusted to suppress malicious responses generated by attackers who submit false sensing data.

Obviously, δ cannot be set as a low value in [0,1]. If this was done, then the attackers with high trust value would get more opportunities to submit false sensing data, resulting in the most malicious responses. In addition, δ cannot be set as the maximum of trust value since sometimes honest SUs may submit false sensing data at a lower probability due to the case of deep shadowing and multipath fading. To find the rational value of δ, the simulation method is a good choice. In this paper, we perform the simulation of suppressing malicious responses in Section 4, and find the rational value of δ is 0.8.

3. Collusive SSDF Attack and Defense Scheme

In this section, we first describe collusive SSDF attack, and then propose a defense scheme by using XOR clustering analysis called XDA to suppress collusive SSDF attack.

3.1. Collusive SSDF Attack Overview

Because the individual sensing report is usually regarded as a binary variable, it is very easy for attackers to launch an SSDF attack by submitting false individual sensing data, resulting in a wrong final decision of FC.

At first, attackers launch an SSDF attack individually and respectively. The power of this individual SSDF attack is finite and can be suppressed by a current trust mechanism easily, such as [8–11]. To avoid the detection of trust mechanism, some attackers attempt to collude with each other and submit false sensing data collusively at the same time. This attack pattern can be called a collusive SSDF attack. Generally, the collusive attack pattern can increase the power of attackers, which has undergone three stages.

In the first stage, an attacker can acquire multiple IDs to falsify data through the sybil attack [21]. There are many methods to defend against the Sybil attacks [22,23]. In particular, if each IP is restricted to acquire an ID, this attack pattern can be addressed easily. In the second stage, an attacker can control multiple computers by embedding trojan viruses. This attack pattern can be suppressed by using a good antivirus software. In the third stage, multiple attackers collaborate together to falsify the data. In this attack pattern, each attacker only has an ID. Currently, this attack pattern is used as a popular collusive attack, especially in CSS to falsify sensing data.

By further analyzing the characteristics of CSS and the collusive attack demand, we have found that three types of threats can be achieved by collusive SSDF attackers, namely: profit-driven, manipulate trust mechanism and disturb data fusion.

- **Profit-driven**: Inspired by some profits, collusive SSDF attackers can conspire with each other to form a collusive clique to falsify the sensing data intentionally. For example, they can monopolize vacant PU spectrums in the CSS environment. They submit false sensing data together to show the spectrum of a PU is in use, although it is idle. In this case, other SUs would recognize that the licensed spectrum is present and would not use the spectrum. Thus, collusive an SSDF attackers clique can gain exclusive access to the target licensed spectrum.
- **Manipulate trust mechanism**: By collusion, collusive SSDF attackers can improve their trust value quickly. For example, one of the collusive SSDF attackers who knows the actual status of a PU spectrum would tell this PU spectrum status to his conspirers in advance, and then sends a query message to the FC. Their trust values can be improved quickly if their sensing data are as consistent as the PU spectrum status.
- **Disturb data fusion**: As we know, the most primal attackers need to disturb some systems. With high trust value, collusive SSDF attackers can bypass the detection of the trust mechanism, and thus submit their false sensing data to indicate that the spectrum band of a PU spectrum is idle. As a result, they can disturb the data fusion of FC successfully, and ultimately deceive honest SUs to interfere PUs. In fact, SUs must never interfere with PUs in CSS [2]. If cause any interference to PUs, the availability of CSS will be questioned.

3.2. Design of XDA Scheme

To design the defense scheme of collusive SSDF attack, we analyze its attack threats, and thus find three kinds of general features as follows.

- **Duality**: SUs often submit the type of "0" and "1" sensing data to represent the hypothesis of the inexistence and the existenc of PU spectrum status. Thus, the sensing behaviors of SUs in the CSS environment can be abstracted as the duality due to the type of "0" and "1" sensing data.
- **Action together**: Collusive SSDF attackers always submit false sensing data together no matter which threats they would launch. They can fake "1" data together to monopolize vacant PU spectrums, or fake "0" data together to disturb the data fusion.
- **High trust value**: Collusive SSDF attackers often have high trust value. With the help of each other, they can improve their trust value by manipulating the trust mechanism.

Considering the "Duality" and "Action together" of general features, we introduce the design idea of XOR distance analysis to construct the defense scheme called XDA to suppress collusive SSDF attack. Meanwhile, the "Action together" of the general feature can make collusive SSDF attackers behave like the lower XOR distance among themselves. Based on this, we analyze the "High trust value" of a general feature to design the algorithm of detecting collusive SSDF attackers.

As shown in Figure 2, the XDA scheme is conducted in three successive stages: XOR distance calculation, collusive SSDF detection and perfect trust mechanism. In the first stage, we design the XOR distance calculation between any two SUs in line with the type of "0" and "1" sensing data. In the second stage, an algorithm is designed to detect collusive SSDF attackers. In the third stage, we can perfect trust mechanism by correcting collusive SSDF attackers' trust value after detecting them at each CSS action.

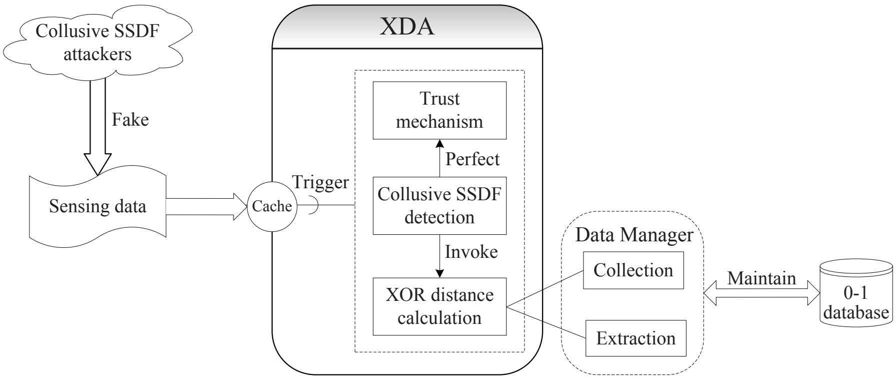

Figure 2. Architectural view of the XDA scheme.

3.2.1. XOR Distance Calculation

It can be seen in Figure 2 that a data manager is responsible for performing two main tasks in this stage. It first collects the sensing data during each CSS action, in which FC is required to store the type of "0" and "1" historical sensing data to a small database called the 0-1 database, rather than discarding them again. The 0-1 database is designed as an extensible database, whose size corresponds to the sensing times of CSS actions. After each CSS action, FC should add a row in the 0-1 database to record the sensing data reported by cooperating SUs. When the current CSS action is numbered as the *k*-th sensing time, the size of the 0-1 database is *k*. The description of the 0-1 database is shown in Table 1.

Table 1. Description of the 0-1 database style.

Sensing Times	**ID(Sensing Data)**					
1	$SU_1(d_1)_1$	$SU_2(d_2)_1$	$\cdots$	$SU_i(d_i)_1$	$\cdots$	$SU_n(d_n)_1$
2	$SU_1(d_1)_2$	$SU_2(d_2)_2$	$\cdots$	$SU_i(d_i)_2$	$\cdots$	$SU_n(d_n)_2$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
k	$SU_1(d_1)_k$	$SU_2(d_2)_k$	$\cdots$	$SU_i(d_i)_k$	$\cdots$	$SU_n(d_n)_k$

Take SU_i as an example, $SU_i(d_i)_k$ is recorded as $SU_i(1)_k$ when SU_i reported "1" at the *k*-th sensing time, $SU_i(d_i)_k \rightarrow SU_i(0)_k$ when reported "0" and $SU_i(d_i)_k \rightarrow SU_i(-)_k$ when reported nothing.

Generally, distance metrics play a very important role in order to measure the similarity among the data sets [24]. For the convenience of calculating the distance between any two SUs, the second task of the data manager is to extract each cooperating SU's sensing data from the 0-1 database as a vector in the current CSS action. For SU_i, his sensing vector can be represented as $\mathbf{D}_i = [SU_i(d_i)_1, SU_i(d_i)_2, \cdots, SU_i(d_i)_k]$. If $SU_i(d_i)_1 \rightarrow SU_i(1)_1$, $SU_i(d_i)_2 \rightarrow SU_i(-)_2$, $SU_i(d_i)_k \rightarrow SU_i(0)_k$, $\mathbf{D}_i$ can be definitely described as [1, -,$\cdots$, 0].

Obviously, the redundant data such as $SU_i(-)_2$ or $SU_j(-)_5$ are useless to calculate the distance between SU_i and SU_j. In this case, we can perfrom Procedure 1 to eliminate the redundant data for the two SUs.

Procedure 1 Eliminate redundancy

```
Input: $\mathbf{D}_i, \mathbf{D}_j$
Output: $\tilde{\mathbf{D}}_i, \tilde{\mathbf{D}}_j$
Initialize $\tilde{\mathbf{D}}_i = \tilde{\mathbf{D}}_j = \emptyset$
for $k = 1, k \leq n, k++$ do
    if $(SU_i(d_i)_k \to SU_i(-)_k) || (SU_j(d_j)_k \to SU_j(-)_k)$ then
        $SU_i(d_i)_k$ and $SU_j(d_j)_k$ are deleted simultaneously;
    else
        $\tilde{\mathbf{D}}_i = \{SU_i(d_i)_k\} \cap \tilde{\mathbf{D}}_i$
        $\tilde{\mathbf{D}}_j = \{SU_j(d_j)_k\} \cap \tilde{\mathbf{D}}_j$
    end if
end for
```

For all the cooperating SUs at the current sensing time, their sensing vector without the redundant data can compose a matrix $\tilde{\mathbf{D}}_{h\times h}$.

$$\tilde{D}_{h\times h} = \begin{pmatrix} d_{11} & \cdots & d_{1h} \\ \vdots & \ddots & \vdots \\ d_{h1} & \cdots & d_{hh} \end{pmatrix}$$

For SU_i and SU_j, the XOR operation between $\tilde{\mathbf{D}}_i$ and $\tilde{\mathbf{D}}_j$ can be described as

$$\tilde{\mathbf{D}}_{ij} = \tilde{\mathbf{D}}_i \oplus \tilde{\mathbf{D}}_j \tag{6}$$

Then, the XOR distance between SU_i and SU_j can be calculated as

$$xd_{ij} = \sum_{d_{ij}(k)\in\tilde{\mathbf{D}}_{ij}, k=1}^{|\tilde{\mathbf{D}}_{ij}|} k * d_{ij}(k) \tag{7}$$

where $d_{ij}(k)$ is the k-th element of $\tilde{\mathbf{D}}_{ij}$ and $\mid \tilde{\mathbf{D}}_{ij} \mid$ is the number of $\tilde{\mathbf{D}}_{ij}$.

For SU_i, the XOR distance related to the cooperating SUs at the current CSS action can compose the XOR distance vector $\boldsymbol{XD}_i = \{xd_{i1}, \cdots, xd_{ij}, \cdots, xd_{ih}\}$, in which h is the number of the cooperating SUs.

It is necessary to normalize the XOR distance in $\boldsymbol{XD}_i$. Otherwise, some SUs may be assigned arbitrarily high XOR distance with SU_i, and another SUs may have an arbitrarily low XOR distance, which introduces a difficulty in comparing them. To ensure that all XOR distance of SU_i lie in [0, 1], xd_{ij} can be normalized as

$$\overline{xd}_{ij} = \frac{xd_{ij}}{max(\boldsymbol{XD}_i)}, xd_{ij} \in \boldsymbol{XD}_i \tag{8}$$

For all cooperating SUs, their normalized XOR distance can form a matrix $\overline{XD}_{h\times h}$.

$$\overline{XD}_{h\times h} = \begin{pmatrix} \overline{xd}_{11} & \cdots & \overline{xd}_{1h} \\ \vdots & \ddots & \vdots \\ \overline{xd}_{h1} & \cdots & \overline{xd}_{hh} \end{pmatrix}$$

3.2.2. Collusive SSDF Detection

We know that collusive SSDF attackers often fake sensing data together, so there may be a low XOR distance between them. With the help of each other, collusive SSDF attackers can get high trust value when $T_j > \delta$, but the lower XOR distance will make them exposed. So, we can detect the SUs who hold high trust value and the minimum from all normalized XD_i $(i \in 1, 2, \cdots, h)$ as the collusive SSDF attackers set (Y_1). In contrast, honest SUs also have high trust value, but there is a higher XOR distance between them. So, we can identify the SUs who hold high trust value and the maximum from

from all normalized XD_i as the honest SUs set (Υ_2). Assuming that $\Gamma = \{T_1, \cdots, T_j, \cdots, T_h\}$ is the trust value set of cooperating SUs, an algorithm is designed in Procedure 2 to detect collusive SSDF attackers.

Procedure 2 Detect collusive SSDF attackers

```
Input: $\Gamma$ and $\overline{XD}_{h \times h}$
Output: $\Upsilon_1$ and $\Upsilon_2$
Initialize $\Upsilon_1 = \Upsilon_2 = \emptyset$
for $i = 1, i \leq h, i++$ do
    Initialize $\Omega_i = \emptyset$
    for $j = 1, j \leq h, j++$ do
        if $T_j > \delta$ then
            $\Omega_i = \{\overline{xd}_{ij}\} \cup \Omega_i$
        end if
    end for
    $\Upsilon_1 = \{argmin(\Omega_i)\} \cup \Upsilon_1$
    $\Upsilon_2 = \{argmax(\Omega_i)\} \cup \Upsilon_2$
end for
```

To ensure the reliability of the data fusion, the sensing data of collusive SSDF attackers should be deleted. Of course, the sensing data of the SUs whose trust values are less than δ should also be deleted. Only the sensing data of honest SUs can be adopted in the data fusion.

3.2.3. Perfect Trust Mechanism

When collusive SSDF attackers are detected, typical issues in perfecting the trust mechanism focus on (1) preventing the increase of their *hon* data in the current CSS action; (2) reducing their *hon* data until $T_i < \delta$; and (3) deleting their sensing data in the process of data fusion, which can be performed by Procedure 3. For the former two issues, collusive SSDF attackers will not get high trust value, thus enhancing the accuracy of trust evaluation. For the last one, collusive SSDF attackers will find it hard to manipulate the final decision of FC.

Let H, F is the set of current SUs' *hon* data and *fal* data, respectively. For $SU_i \in \Theta$ (the set of cooperating SUs), $hon_i \in H$ and $fal_i \in F$. As mentioned in Figure 1, d_i is the individual sensing data from SU_i and d is the final decision from FC.

Procedure 3 Perfect trust mechanism

```
Input: $\Theta$,$\Gamma$, $\Lambda_1$ and $\Lambda_2$
Output: $H$, $F$
for each $SU_i \in \Theta$ do
    if $SU_i \in \Lambda_1$ then
        $hon_i = hon_i + 0$ and his sensing data are deleted
        if $T_i \geq \delta$ then
            repeat $hon_i - -$
            until $T_i < \delta$
        end if
    else
        if $d_i == d$ then
            $hon_i = hon_i + 1$
        else
            $fal_i = fal_i + 1$
        end if
    end if
end for
```

4. Simulation Results and Discussion

We perform simulations to validate the performance of the XDA scheme and discuss the simulation results. The general simulation setup is shown in Table 2.

Table 2. Description of simulation elements.

Parameters	Description	Default
N_s	Number of SUs	60
N_p	Number of PUs	3
cycle	Number of cycle simulation	100
round	Rounds of attack	50
p_a	Percentage of attackers	10∼50%
δ	Threshold of trust value	0.8

The simulations are performed by cycle-based fashion. At each cycle, SUs are selected randomly to execute CSS actions with each other. After several cycles, a trusted network topology is gradually generated by trust mechanism. FC then utilizes it to execute the following CSS actions, and update the trust value on the corresponding SUs. In addition, the case of deep shadowing and multipath fading can cause false detection for honest SUs. Therefore, sometimes, honest SUs may submit false sensing data at a lower probability in CSS. Without loss of generality, the behavior pattern for honest SUs in the simulations is modeled to submit false sensing data at the probability of 0.2.

To perform the simulations better, it is necessary to select a rational value of δ. As $T_i \in [0, 1]$, δ can be considered from the three types of optional states [low, medium, high]. Then, we can perform the simulation of suppressing malicious responses to validate the effectiveness of the XDA scheme under the three types of optional states of δ. In this simulation, 0.3, 0.5 and 0.8 denotes the low, medium and high state of δ respectively. As shown in Figure 3, the performance of the XDA scheme at $\delta = 0.8$ is the best. Therefore, the rational value of δ should be selected as 0.8 in the simulations. We can also see that the XDA scheme is better than Baseline at suppressing malicious responses, even though δ is selected as 0.8 for Baseline.

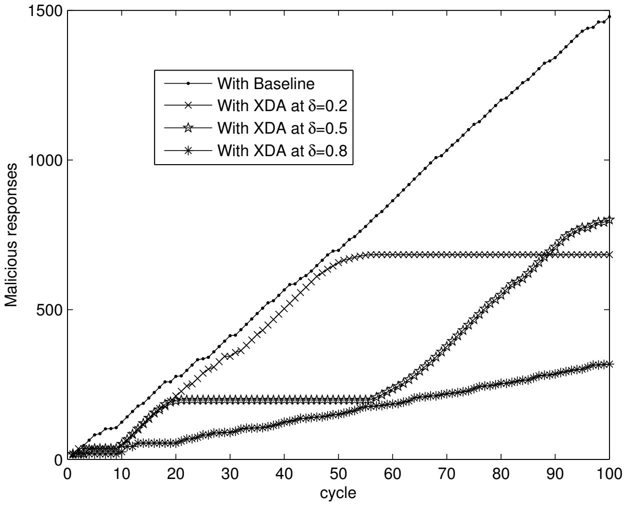

Figure 3. Suppressing malicious responses.

To suppress a collusive SSDF attack, an important measure is to suppress the increase of collusive SSDF attackers' trust value. Therefore, we choose a collusive SSDF attacker randomly to observe the variation of his trust value with Baseline and XDA. Figure 4 shows that collusive SSDF strategies make the attacker's trust value fluctuate along with the various cycles. The trust value usually outweighs δ in Baseline. Fortunately, the trust value can be reduced by the XDA scheme after 10 cycles. This is because the XDA scheme can delete *hon* data of collusive SSDF attackers and reduce this data in the trust mechanism. In addition, collusive SSDF attackers will deviate from the real trust value by forming high-trust attackers, and thus cause some network trust errors (*nte*). Higher errors indicate lower accuracy in calculating the trust value. The *nte* can be specified by:

$$nte = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{T_i'}(T_i' - T_i)^2} \tag{9}$$

where T_i' and T_i are the real and measured trust value of SU_i, respectively.

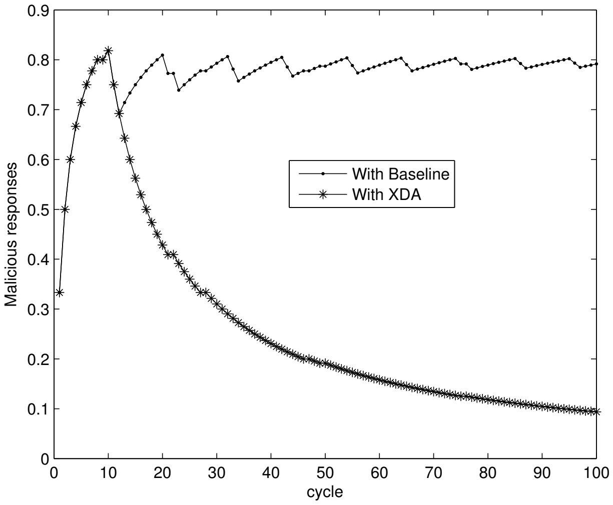

Figure 4. Variation of a collusive SSDF attacker′s trust value.

Generally, the power of attackers is limited when the percentage of attackers decrease below 10%, but the network may become unavailable when the percentage of attackers decreases more than 50%. It can be found in Figure 4 that the network trust errors with Baseline is below 0.08 when the percentage of collusive attackers decreases below 10%, but the number of network trust errors is greater than 0.42 when the percentage of collusive attackers decreases by more than 50%. Thus, we vary the percentage of collusive attackers from 10% to 50% in the following simulations.

In the simulation of *nte*, the real trust value of an attacker is randomly assigned in the interval (0, δ]. However, collusive SSDF attackers will fake sensing data on the basis of maintaining their trust value in the interval [δ, 1]. Without loss of generality, we employ the averaged *nte* data of 100 cycles as the simulation results. As shown in Figure 5, the XDA scheme can reduce *nte* effectively. Without any guard measures, the *nte* curve with Baseline increases rapidly. By perfecting trust management after collusive SSDF attackers are detected, the *nte* curve with XDA increases smoothly.

Finally, we validate the performance of XDA in terms of attack success ratio when attackers launch collusive SSDF attackers with high trust value. This simulation is performed at two types of attack patterns including monopolizing vacant PU spectrums and disturbing data fusion. Without loss of generality, we employ the averaged attack success ratio data of 50 rounds of attack as the simulation results. At each round of attack, several cooperating SUs are selected randomly to perform a CSS action from honest SUs and collusive SSDF attackers.

As shown in Figure 6, the XDA scheme can suppress the attack success ratio better than Baseline under the "AND" and "Majority" rule when collusive SSDF attackers fake all "1" sensing data to monopolize vacant PU spectrums. In the "OR" rule, only one false "1" sensing data can mislead the final decision as "1". Both of the schemes are difficult to defend against collusive SSDF attack. Therefore, collusive SSDF attackers can easily monopolize vacant PU spectrum in the "OR" rule, and thus gain exclusive access to the target licensed spectrum. In summary, to gain a reliable final decision against the threat of monopolizing vacant PU spectrums, the "OR" rule may not be a good choice.

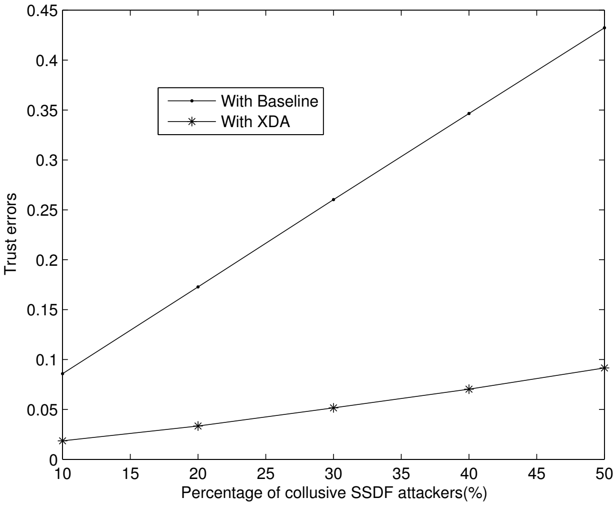

Figure 5. *nte* with the guard of XDA.

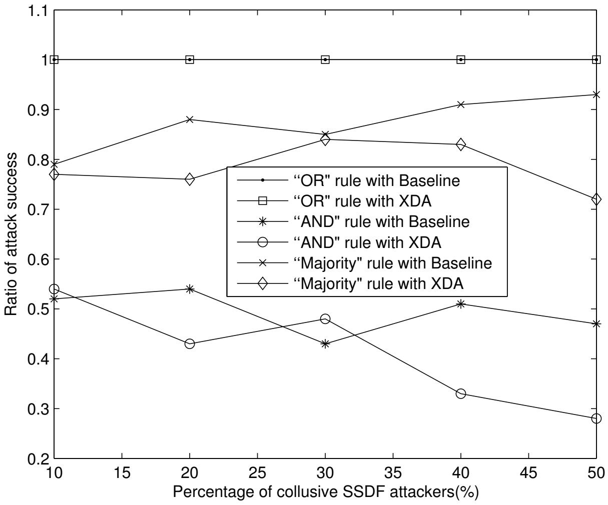

Figure 6. Suppressing MAC attack success ratio against monopolizing vacant PU spectrum.

We can also see in Figure 7 that the XDA scheme can suppress the attack success ratio better than Baseline under the "OR" and "Majority" rule when collusive SSDF attackers fake all "0" sensing data to disturb data fusion. In the "AND" rule, only one false "0" data can mislead the final decision as "0". Both the two schemes are difficult to defend against collusive SSDF attack. Therefore, collusive SSDF attackers can easily disturb data fusion in the "AND" rule, and thus deceive honest SUs to interfere with PUs. In summary, to gain a reliable final decision against the threat of disturbing data fusion, the "AND" rule may be not a good choice with the threat of MAC attack.

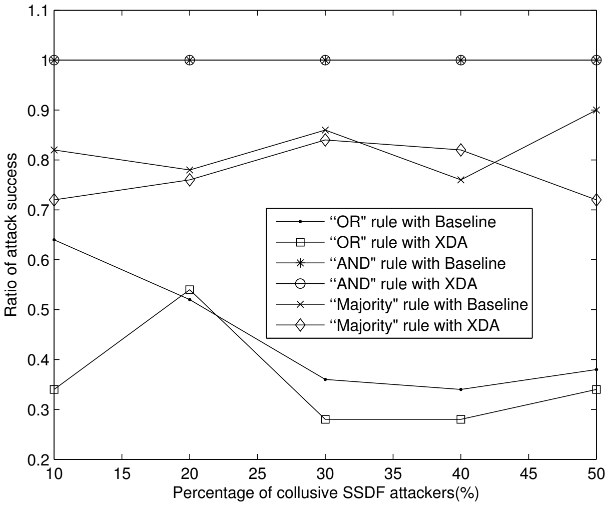

Figure 7. Suppressing MAC attack success ratio against disturbing data fusion.

5. Conclusions

In this paper, we analyzed the threats of collusive SSDF attack and proposed the XDA scheme to suppress such attacks. The XDA scheme was conducted in three successive stages: XOR distance calculation, collusive SSDF detection and perfect trust mechanism, in which XOR clustering analysis is introduced to design the XDA scheme due to the type of "0" and "1" historical sensing data. Simulation results show that our XDA scheme can enhance the accuracy of trust evaluation and suppress the collusive SSDF attack success ratio to some extent.

Acknowledgments: This work is supported in part by the National Natural Science Foundation of China(61602379), the International cooperation and exchange program of Shaanxi Province (2016KW-034), the Science and Technology Innovation Team of Shaanxi Province for Broadband Wireless and Application (2017KCT-30-02), the Natural Science Foundation of Education Department of Shaanxi Province (15JK1649), the New Star Team of Xi'an University of Posts and Telecommunications.

Author Contributions: Jingyu Feng engaged in the XOR distance calculation and collusive SSDF detection algorithm in the design of the XDA scheme, and prepared the manuscript. Man Zhang performed the simulations. Yun Xiao led the reasearch project and supervised the activities as the corresponding author. Hongzhou Yue engaged in perfecting trust mechanism in the design of the XDA scheme.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CSS	Cooperative Spectrum Sensing
SU	Secondary User
PU	Primary User
FC	Fusion Center
SSDF	Spectrum Sensing Data Falsification
XDA	XOR Distance Analysis

References

1. Federal Communications Commission. Spectrum Policy Task Force. Rep. ET Docket no. 02-135. 2002. Available online: http://www.fcc.gov/sptf/files/SEWGFinalReport_1.pdf (accessed on 25 September 2017).
2. Mitola, J. Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio. Ph.D. Dissertation, Royal Instutute of Technology (KTH), Stockholm, Sweden, June 2000.
3. Akyildiz, I.F.; Lo, B.F.; Balakrishnan, R. Cooperative spectrum sensing in cognitive radio networks: A survey. *Phys. Commun.* **2011**, *4*, 40–62.
4. Cabric, D.; Mishra, S.; Brodersen, R. Implementation issues in spectrum sensing for cognitive radios. In Proceedings of the Asilomar Conference on Signals, Systems, and Computers, Asilomar, Pacific Grove, CA, USA, 7–10 November 2004; pp. 772–776.
5. Lin, Y.P.; He, C.; Jiang, L.G.; He, D. Cooperative spectrum sensing based on stochastic resonance in cognitive radio networks. *Sci. China Inf. Sci.* **2014**, *57*, 1–10.
6. Feng, J.Y.; Zhang, Y.Q.; Lu, G.Y.; Zhang, L. Securing cooperative spectrum sensing against rational SSDF Attack in cognitive radio networks. *KSII Trans. Int. Inf. Syst.* **2014**, *8*, 1–17.
7. Chen, R.L.; Park, J.M.; Hou, Y.T. Toward secure distributed spectrum sensing in cognitive radio networks. *IEEE Commun. Mag.* **2008**, *46*, 50–55.
8. Qin, T.; Yu, H.; Leung, C. Towards a trust-aware cognitive radio architecture. *ACM Sigmob. Mob. Comput. Commun. Rev.* **2009**, *13*, 86–95.
9. Zeng, K.; Peng, Q.H.; Tang, Y.X. Mitigating spectrum sensing data falsification attacks in hard-decision combining cooperative spectrum sensing. *Sci. China Inf. Sci.* **2014**, *57*, 1–9.
10. Chen, R.; Park, J.M.; Bian, K. Robust distributed spectrum sensing in cognitive radio networks. In Proceedings of the 30th IEEE INFOCOM Conference, Phoenix, AZ, USA, 13–18 April 2008; pp. 1876–1884.
11. Kar, S.; Sethi, S.; Sahoo, R.K. A Multi-factor Trust Management Scheme for Secure Spectrum Sensing in Cognitive Radio Networks. *Wirel. Pers. Commun.* **2017**, *97*, 2523–2540.
12. Akyildiz, I.F.; Lo, B.F.; Balakrishnan, R. Cooperative spectrum sensing in cognitive radio networks: A survey. *Phys. Commun.* **2011**, *4*, 40–62.
13. Peh, E.; Liang, Y.C.; Guan, Y.L.; Zeng, Y. Optimization of cooperative sensing in cognitive radio networks: A sensing-throughput tradeo view. *IEEE Trans. Veh. Technol.* **2009**, *58*, 5294–5299.
14. Akyildiz, I.F.; Lee, W.Y.; Chowdhury, K.R. CRAHNs: Cognitive radio ad hoc networks. *Ad Hoc Netw.* **2009**, *7*, 810–836.
15. Morid, M.A.; Shajari, M. An enhanced e-commerce trust model for community based centralized systems. *Electron. Commer. Res.* **2012**, *12*, 409–427.
16. Li, X.Y.; Zhou, F.; Yang, X.D. Scalable feedback aggregating (SFA) overlay for large-Scale P2P trust management. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 1944–1957.
17. Boukerche, A.; Ren, Y.; Pazzi, R. An adaptive computational trust model for mobile ad hoc networks. In Proceedings of the 5th International Conference on Wireless Communications and Mobile Computing, Leipzig, Germany, 21–24 June 2009; pp. 191–195.
18. Mohaisen, A.; Hopper, N.; Kim, Y. Keep your friends close: Incorporating trust into social-network-based sybil defenses. In Proceedings of the 30th IEEE INFOCOM Conference, Shanghai, China, 10–15 April 2011; pp. 1943–1951.
19. Jϕsang, A.; Ismail, R. The beta reputation system. In Proceedings of the 15th Bled Electronic Commence Conference, Bled, Slovenia, 17–19 June 2002.
20. Gamma Function. Available online: http://en.wikipedia.org/wiki/Gammafunction (accessed on 18 October 2017).
21. Douceur, J.R. The sybil attack. In *International Workshop on Peer-To-Peer Systems*; Springer: Berlin/Heidelberg, Germany; pp. 251–260.
22. Yu, H.; Gibbons, P.; Kaminsky, M.; Xiao, F. Sybillimit: A near-optimal social network defense against sybil attacks. In Proceedings of the 2008 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 18–22 May 2008.

23. Yu, H.; Kaminsky, M.; Gibbons, P.; Flaxman, A. Sybilguard: Defending against sybil attacks via social networks. In Proceedings of the 2006 ACM SIGCOMM, Pisa, Italy, 11–15 September 2006; pp. 267–278.
24. Irani, J.; Pise, N.; Phatak, M. Clustering Techniques and the Similarity Measures used in Clustering: A Survey. *Int. J. Comput. Appl.* **2016**, *134*, 9–14.