SCIENCE CHINA
Information Sciences

• LETTER •

May 2017, Vol. 60 059101:1–059101:3
doi: 10.1007/s11432-015-0337-x

Avoiding monopolization: mutual-aid collusive attack detection in cooperative spectrum sensing

Jingyu FENG[1,3], Guangyue LU[1*], Yuqing ZHANG[2] & Honggang WANG[1]

[1]*Department of Communication Engineering, Xi'an University of Posts and Telecommunications, Xi'an* 710121*, China;*
[2]*National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing* 101408*, China;*
[3]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093*, China*

Received June 12, 2016; accepted July 8, 2016; published online November 9, 2016

Citation Feng J Y, Lu G Y, Zhang Y Q, et al. Avoiding monopolization: mutual-aid collusive attack detection in cooperative spectrum sensing. Sci China Inf Sci, 2017, 60(5): 059101, doi: 10.1007/s11432-015-0337-x

Dear editor,

Federal Communications Commission found that most of the allocated spectrum bands are not efficiently utilized by the licensed primary users (PU) [1]. In order to improve spectrum utilization, it has been suggested that opportunistic access of any available valid spectrum from PUs should be given to unlicensed secondary users (SU) [2].

Cooperative spectrum sensing (CSS) has been recently considered as a viable means to enhance the detection performance by exploiting the observations of spatially located SUs. However, CSS assumes all SUs are honest, and thus offering opportunities to launch the spectrum sensing data falsification (SSDF) attack. This attack can be launched by collusive pattern, in which attackers conspire with each other to form a collusive clique to falsify the sensing data intentionally.

Fortunately, the organization of current collusive attack is relaxed, which can be suppressed by trust mechanism easily. Various studies of trust mechanism have been proposed [3–5]. They estimate whether an SU is trustworthy or not by his historical sensing behaviors and give low weights to less trustworthy SUs or even discard their sensing data when generating a final decision.

* Corresponding author (email: tonylugy@163.com)

The authors declare that they have no conflict of interest.

In this letter, we argue that securing CSS with only trust mechanism is not enough and find mutual-aid collusive (MAC) attack. A quick recovery to trust can be employed by MAC attackers to escape the detection of trust mechanism. In this base, MAC attackers falsify sensing data together to indicate that PUs always exist, thereby depriving other SUs of their spectrum opportunities. To avoid such spectrum monopolization, a defense scheme called DMAC using '0-1' similarity measure is proposed to detect MAC attack.

MAC attack overview. Driven by the profit that the attackers who help other attackers can get help from them, MAC attackers falsify sensing data to indicate the spectrum bands of all PUs are in use, although they are unused. In this case, honset SUs will be misled that PUs are present and give up their spectrum opportunities, while attackers belonging to the mutual-aid collusive clique can gain the exclusive access to vacant PU spectrum.

Specially, MAC attackers are extremely sensitive to their trust value. Assuming m is the number of MAC attackers and SU_k is one of them, MAC attack is launched under the constraint

$$\|\varepsilon \leqslant t_k \leqslant \varepsilon + \lambda\| \leqslant m/2.$$

t_k denotes SU_k's trust value and ε is the threshold of trust value. For $t_k \geqslant \varepsilon$, SU_k will be identified as honest. MAC attackers would begin to improve their trust when $\|\varepsilon \leqslant t_k \leqslant \varepsilon + \lambda\| \leqslant m/2$. $\|\varepsilon \leqslant t_k \leqslant \varepsilon + \lambda\|$ is the number of MAC attackers under the case $\varepsilon \leqslant t_k \leqslant \varepsilon + \lambda$. Actually λ ($\varepsilon \leqslant \lambda \leqslant 1 - \varepsilon$) is the trust warning line. Under the above constraint, the MAC attack procedure can be conducted in a round mode with four phases.

- **MAC-launching.** MAC attackers always report "1" no matter whether PU signals are present.
- **Self-evaluating.** Each SU_k calculates t_k after attack and broadcasts it to his conspirators.
- **Trust-warning.** Each SU_k checks whether $\|\varepsilon \leqslant t_k \leqslant \varepsilon + \lambda\| \leqslant m/2$. Yes, go to the next phase. No, continue the "MAC-launching" phase.
- **Trust-improving.** A quick recovery to trust is performed in this phase. One of MAC attackers who knows the status of a PU spectrum tells it to his conspirers in advance. MAC attackers' trust value can be improved quickly when their sensing data are the same as the PU status. This phase continues until $\|t_k \geqslant \varepsilon + \lambda\| = m$.

Design of DMAC scheme. MAC attackers often falsify sensing data together, so they may behave high similarity among themselves. We also note that the individual sensing report of each SU in CSS is a binary variable where "1" denotes the presence of the PU signal and "0" is the absence [6]. Based on these, '0-1' similarity measure can be introduced to the DMAC scheme, and thus avoiding mass mathematical computation.

In the current CSS action, we firstly extract each cooperating SU's historical sensing data as a vector. For any two vectors (C_i, C_j) derived from two SUs, such as SU_i and SU_j, Procedure 1 is performed to eliminate the redundant data, where $C_i(k) =$ " $-$ " when SU_i reported nothing at the k-th CSS action.

Procedure 1. Eliminate redundancy. Input: C_i, C_j; Output: $\tilde{C}_i$, $\tilde{C}_j$. (1) Initialize $\tilde{C}_i = \tilde{C}_j = \emptyset$. (2) For each $C_i(k)$ and $C_j(k)$, if $(C_i(k) =$ " $-$ ")$||(C_j(k) =$ " $-$ "), then $C_i(k)$ and $C_j(k)$ are discarded simultaneously; else, $\tilde{C}_i \leftarrow C_i(k)$ which is placed to $\tilde{C}_i$ in a sequence, and $\tilde{C}_j \leftarrow C_j(k)$ which is placed to $\tilde{C}_j$ in a sequence.

Then, the '0-1' similarity between any two vectors can be measured as

$$\mathrm{sim}_{ij} = 1 - \left| \frac{\|\tilde{1}\|_i}{\|\tilde{C}_i\|} - \frac{\|\tilde{1}\|_j}{\|\tilde{C}_j\|} \right|. \tag{1}$$

$\|\tilde{1}\|_i$ is the amount of "1" in $\tilde{C}_i$. $\|\tilde{C}_i\|$ is the amount of elements in $\tilde{C}_i$.

It can be proofed that $\left| \frac{\|\tilde{0}\|_i}{\|\tilde{C}_i\|} - \frac{\|\tilde{0}\|_j}{\|\tilde{C}_j\|} \right| = \left| \frac{\|\tilde{C}_i\| - \|\tilde{1}\|_i}{\|\tilde{C}_i\|} - \frac{\|\tilde{C}_j\| - \|\tilde{1}\|_j}{\|\tilde{C}_j\|} \right| = \left| \frac{\|\tilde{1}\|_i}{\|\tilde{C}_i\|} - \frac{\|\tilde{1}\|_j}{\|\tilde{C}_j\|} \right|$.

For all cooperating SUs, their sensing vectors can compose a matrix $\mathrm{SIM}_{n \times n}$.

Generally, attackers behave honestly sometimes to improve their trust, so their historical trust values fluctuate from high to low. Similarly, MAC attackers' historical trust values also fluctuate due to "trust-improving". To reduce computational complexity, we can identify the cooperating SUs whose historical trust values fluctuate, and then filter out MAC attackers from them. Such identified cooperating SUs are called anoles in the DMAC scheme.

$\mathrm{SIM}_{l \times l}$ is extracted from $\mathrm{SIM}_{n \times n}$, l is the number of anoles. Assuming SU_i is an anole identified in the current CSS action and $\mathrm{SIM}_i = [\mathrm{sim}_{i1}, \mathrm{sim}_{i2}, \ldots, \mathrm{sim}_{ij}, \ldots, \mathrm{sim}_{ih}]$. The outlier value of SU_i can be calculated as

$$o_i = 1 - \frac{1}{l-1} \sum_{j=1, j \neq h}^{l-1} \left| \mathrm{sim}_{i(j+1)} - \mathrm{sim}_{ij} \right|. \tag{2}$$

Procedure 2 is performed to detect MAC attackers, in which δ is the threshold of outlier value.

Procedure 2. Detect MAC attackers. Input: A (the set of anoles); Output: M (the set of MAC attackers). (1) Initialize $M = \emptyset$. (2) For each $\mathrm{SU}_i \in A$, if $O_i \geqslant \delta$, then $M \leftarrow \{\mathrm{SU}_i\}$ which is placed to M; else SU_i behaves honestly sometimes.

When MAC attackers are detected, typical issues in perfecting trust mechanism focus on (1) preventing the increase of their "the number of honest sensing" and (2) filtering out their sensing data in the process of data fusion, which can be performed by Procedure 3 where d_i is the individual sensing data from SU_i and d is the final decision.

Procedure 3. Perfect trust. Input: M, hon, fal; Output: hon, fal. For each $\mathrm{SU}_i \in$ cooperating SUs, if $\mathrm{SU}_i \in M$, then $\mathrm{hon}_i = \mathrm{hon}_i + 0$ and discard his sensing data; else if $d_i == d$, $\mathrm{hon}_i = \mathrm{hon}_i + 1$; else $\mathrm{fal}_i = \mathrm{fal}_i + 1$.

To evaluate the performance of DMAC, a basic trust scheme called Baseline is described by abstracting the commonality of existing trust mechanism schemes. Noting that the binary individual sensing report again, the trust value of each SU can be calculated by two indexes: the number of honest sensing (hon) and the number of false sensing (fal). Take SU_i as an example, its trust value t_i can be calculated as

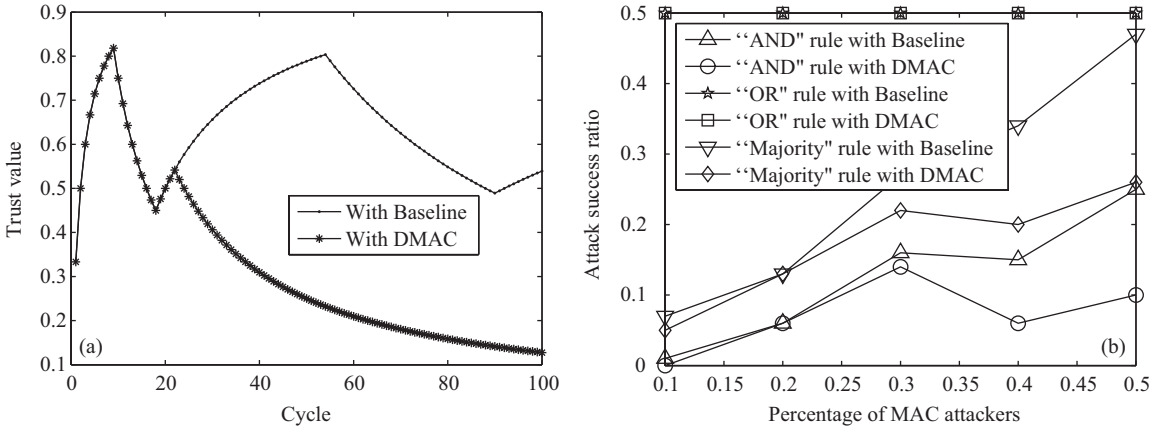

Figure 1 (a) Variation of an MAC attacker's trust value; (b) suppressing MAC attack success ratio.

$$t_i = \frac{1 + \mathrm{hon}_i}{2 + \mathrm{hon}_i + \mathrm{fal}_i}. \tag{3}$$

The simulations are performed by cycle-based fashion. The simulation parameters are set as: $\lambda = 0.3$, $\varepsilon = 0.5$, and $\delta = 0.8$. As shown in Figure 1(a), MAC strategies can make an attacker's trust value fluctuate along with cycles. Such trust value usually outweighs ε in Baseline. Fortunately, by preventing the increase of "hon" with DMAC, its trust value can be reduced after 25 cycles.

The final decision of CSS can be made by three typical fusion rules: the "AND", "OR" and "Majority" rule [6]. We can find in Figure 1(b), DMAC can suppress MAC attack success ratio better than Baseline under the "AND" and "Majority" rule. For the "OR" rule, only one false "1" data can make the final decision as "1". Therefore, to make a reliable final decision, the "OR" rule is not a good choice with the threat of MAC attack.

Conclusion. In this letter, we report the discovery of MAC attack and present the DMAC scheme to defend against this attack. '0-1' similarity measure is introduced in DMAC to identify MAC attackers. The DMAC scheme can be used to perfect trust mechanism and reduce the trust value of MAC attackers, and thus suppressing MAC attack success ratio to some extent.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61301091, 61572460, 61272481), Open Foundation of State Key Laboratory of Information Security (Grant No. 2015-MS-14), and New Star Team of Xi'an University of Posts and Telecommunications.

Supporting information The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Federal Communications Commission. Spectrum Policy Task Force. Technical Report, Rep. ET Docket No. 02-135. 2002
2 Mitola J. Cognitive radio: an integrated agent architecture for software defined radio. Dissertation for Ph.D. Degree. Stockholm: Royal Instutute of Technology (KTH), 2000
3 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: 042318
4 Feng J Y, Lu G Y, Chang H. Behave well: how to win a pop vacant band via cooperative spectrum sensing. KSII Trans Int Inf Syst, 2015, 9: 1321–1336
5 Pei Q Q, Yuan B B, Li L, et al. A sensing and etiquette reputation-based trust management for centralized cognitive radio networks. Neurocomputing, 2013, 101: 129–138
6 Akyildiz I F, Lo B F, Balakrishnan R. Cooperative spectrum sensing in cognitive radio networks: a survey. Phys Commun, 2011, 4: 40–62