Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105(Online)
Volume **2**, Number **1**, February **2017**

A Distributed Cross-Layer Compromise Detection Mechanism for Wireless Sensor Networks

Ying-Jun Chen[1], Gwo-Jiun Horng[2*], and Sheng-Tzong Cheng[1]

[1]Department of Computer Science and Information Engineering
National Cheng Kung University, Taiwan
[2]Department of Computer Science and Information Engineering
Southern Taiwan University of Science and Technology, Taiwan
[*]email: grojium@gmail.com

Received September 2016; revised February 2017

Abstract. *Since wireless sensor networks (WSNs) are always deployed in open environments, the sensors are exposed directly to attack by adversary. Because sensors of WSNs are physically scattered in public areas, the security issues in WSNs are critical. Although there are many security mechanisms in WSNs, such as intrusion detection and fault tolerant system, few papers propose detecting compromised for WSNs. In this paper, a distributed cross-layer compromise detection mechanism (CLCD) is proposed and analysze. The proposed system combines the information of each layer in the communication protocol and a distributed mechanism in order to detect which sensors were already compromised. Due to the characteristics of low power, low computation ability and low storage space, the design goals are simplicity and high-efficiency. When compromised sensors can be detected, the WSNs could be safer in practice.*
Keywords: Wireless Sensor Network, Intrusion Detection, Cross-layer, Compromised, Distributed Mechanism.

1. **Introduction.** The wireless sensor network (WSN) is the technology which integrate the sensing, computation and network ability into a small package. It uses sensors to monitor the environment or the specific target in sensing area and collects the information and then transfer to the monitor by wireless network. WSNs are usually applied in military systems and surveillance systems. Since WSNs always deployed in open environments [2], the adversary attacks WSNs by variant avenues [14]. For example, physical attack [15, 16, 17, 18], denial-of-service, battery exhaustion [1], time synchronization [9], location discovery, attacks on routing. Few researches propose the compromise detection mechanisms. Furthermore, they are designed in centralized model for specific wireless sensor networks, such as static wireless sensor networks. Consequently, we believe that a general model for all wireless sensor networks should be developed. This paper proposes a general distributed compromise detection mechanism for wireless sensor networks. Besides, the cross-layer mechanism will enable the detection mechanism more accurate and more comprehensive. Finally, for some limitation of wireless sensor network, like power, computation, storage space, etc., therefore simple and high-efficiency is also our design purpose.

A few researches on detection issue focus on compromise detection [11][10][8]. The design in WSNs must consider the characteristics of WSNs. The WSN nodes use battery power and their power capabilities are limited due to its small size of node. The

transmission bandwidth of the network is low. Also, the network needs enough power to work steadily for a long time. As a result, this paper proposes a distributed system model in order to conserve the communication resources. This distributed system model is designed for detecting compromised sensors in WSNs. This paper classify 5 types for detection mechanisms. Those types of detection include intrusion detection and fault detection. Type 1 and 2 detections belong to intrusion detection. And type 3, 4 and 5 detections belong to fault detection. In this paper, we use the detection mechanisms around every protocol layers for the perfection of our compromise detection mechanism. And we employ intrusion and fault detection mechanisms to achieve our objective.

2. **Related work.** A compromised sensor is an authorized sensor that has been captured by an adversary. An adversary can inject false data into a compromised sensor; the adversary also can modify, forge, or discard data received from another sensor without being detected because he has access to the identifier and secret key that allowed the sensor to be a valid part of the network. Security management in WSNs consists of the intrusion prevention and intrusion detection. The intrusion prevention is the first line of defense, such as encryption and authentication mechanisms and so on. The mechanisms provide the function of authentication and encryption by using the disclosed key of a time interval [3][4]. The proposed schemes utilize less storage, and they are very efficient to defend many attacks in the WSNs. The intrusion detection is the second line of defense, such as DoS detection and so on. Many investigations on detecting DoS attacks and faults are proposed. We define the compromised node and design an effective model for compromise detection. Intrusion detection mechanisms can be used to identify the intruded sensor nodes [11][10][8]. Bhuse et al. [11] proposed some straightforward and efficient detection at all protocol layers. Silva et al. [10] proposed a decentralized IDS which is based on the inference of the network behavior. It analysis the events detected by a monitor node, for instance, a node that implements the IDS system. On at et al. [8] proposed an IDS system. Sensor nodes with IDS system in the network are responsible for monitoring their neighbors and looking for intruders. Fault detection and anomaly detection mechanisms can be used to identify the failed or misbehaving nodes. Besides, the fault tolerant system can be used to tolerate the fault data [13][7][5]. Chen et al. [13] proposed a distributed fault detection algorithm which diagnoses the faulty nodes by detecting the anomaly of monitored data in sensors and base station. Du et al. [7] proposed anomaly detection by using a rang-free localization scheme. Gupta et al. [5] proposed a fault-tolerant model in a heterogeneous sensor network (with two types of nodes). Compromise detection mechanism that can identify compromised nodes in WSNs has been developed and analyzed. Zhang et al. [12] provided a sample to identify compromised nodes in an application where the specific beacon nodes that have their location are responsible for providing location reference to other sensors; there are two phases in this algorithm. In first phase, it computes the compromised core including some contingent compromised nodes. The second phase uses maximum matching to further eliminate compromised nodes and identifies the approximate compromised nodes.

3. **System model.** The WSN nodes use battery power, and their power capabilities are limited due to its small size of node. So we designed a system model for compromise detection for WSNs by some characters of WSNs. In this chapter, we proposed a distributed system model to detect compromised sensors for WSNs.

3.1. **The environment of CLCD system model.** In the open environment, the WSN suffers from many variant attacks. If crypto graphic keys of sensor nodes are compromised, an attacker may monitor and control it unscrupulously. It is adventurous in that situation

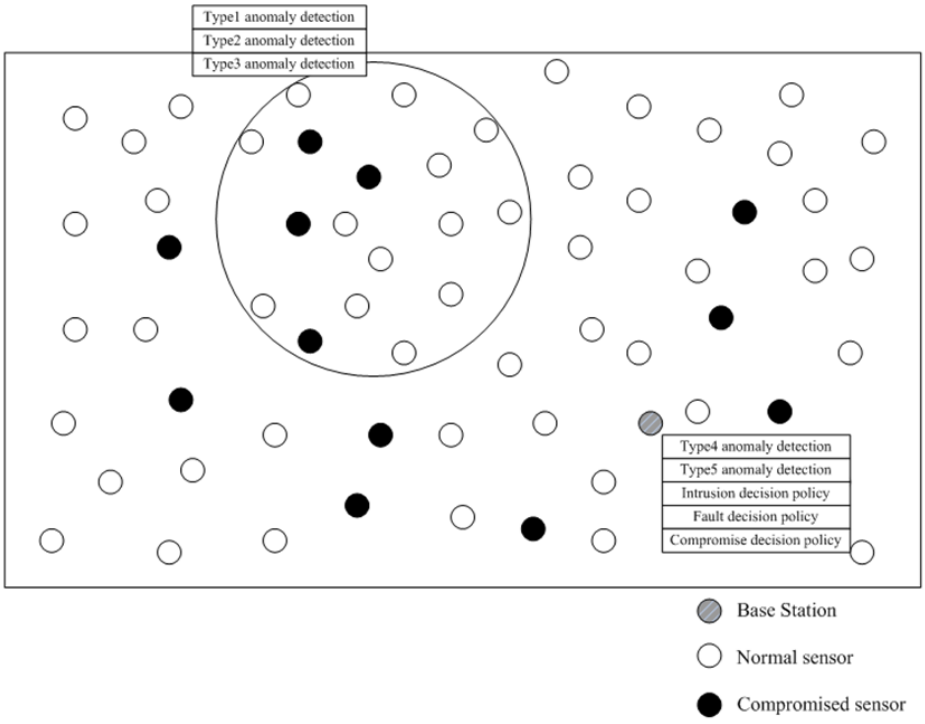


FIGURE 1. The environment of CLCD system model

so WSNs requires a model to guard sensors against such situation. The compromised sensors we observed often issue abnormal messages and then attack other normal sensors actively. However, attackers have no knowledge about the detailed meaning of messages in the compromised sensor nodes. Attackers do not know which fields represent the sensor's neighbors and which fields represent the data that the sensor sensed and so on. They always collaborate with each other to compromise other normal sensors. In this paper, we put our efforts on the efficiency and precision issues. The proposed compromise detection model shown as Fig.1 depict the environment and our compromise detection mechanism. The little white, gray and black cycles represent the normal sensor, base station and compromised sensor respectively. The bigger cycle represents the event region. When sensors begin to sense surrounding information, they will execute Type1, Type2 and Type3 detection mechanisms and then send messages to the base station. When the base station received a message from a sensor, it will execute Type4, Type5 detection mechanisms and intrusion, fault, compromise decision policies. At last, the compromised nodes are detected in the base station.

3.2. **Assumptions.** In the proposed mechanism, some assumptions about precision and efficiency are made. The following paragraphs are assumptions applying to this mechanism. First, most WSNs are based on event-driven services. The detection mechanism is designed for event-driven applications exclusively. Event-driven services are the mainly services in the sensor networks so that they can be used in most of the applications in WSNs. Secondly, in our detection mechanism, parameters are determined according to which service that the sensor network applies. Therefore, it will be more efficient. Finally, this paper assume the base station of WSN has sufficient storages, computation and communication capabilities. And it always can be trusted.

3.3. **Anomaly Detection classification.** In the detection mechanism, an anomaly detection is combined with two types detection methods. The first kind of detection is fault detection, which diagnoses if the sensor behaves abnormally or generates passing failures. The second kind of detection is intrusion detection, which seeks out the malicious intruders in the WSNs. Many fault detection and intrusion detection mechanisms are discussed and delivered by many investigations. We integrate these kinds of detection mechanisms in order to detect the compromised sensor nodes. These cause our compromise detection mechanism more comprehensive and precise.

3.4. **Data structure.** There are two kinds of data structure in our system model. Because we proposed mechanism is distributed, it needs a data structure to store some useful information which is stored for communicating among sensors and the base station. The two tables are Individual Compromise Detection Table (ICDT) and Entire Compromise Detection Table (ECDT) respectively. The ICDT which is built for preparing the information for individual detection mechanisms and for the base station is stored in each sensor node. In addition, the ECDT which is built for storing all sensors' information and the result of the compromise detection mechanism is stores in the base station. Our detection mechanism includes two mechanisms that are local detection engine and cooperative detection engine. These are built for efficiently constructing the distributed system. The ICDT stores all the information which the local detection engine generates and uses. In addition, the ECDT stores all the information for the cooperative detection engine uses and generates.

3.5. **System architecture.** This system architecture we designed for distributed compromise detection uses the two tables we proposed for information's storages. Fig.2 shows the diagram of our system architecture. At first, the sensors are triggered by an event from the base station. Then, the sensors start to sense and collect surrounding information and execute their local detection engine. After executing the engine, the sensors transfer some information to the base station. When the base station receives messages from sensors, it will collect some information in ECDT and execute the global detection engine. At last, the compromised nodes will be detected.

Our compromise detection mechanism consists of two main engines: local detection engine and cooperative detection engine. These two engines perform detection mechanisms including fault detection and intrusion detection. Besides, ICDT and ECDT are designed for information storages. These are described in the last sub-section. Moreover, local and global data collections are responsible for data collection in sensor nodes and base station respectively. This mechanism is triggered by events. Finally, the base station of WSNs detects the compromised sensor nodes.

This architecture achieves the distributed computing by using local detection engine. It deals properly with independent data computing in all sensor nodes to distribute the centralized computing overhead. It is triggered as receiving package every time. Otherwise, the cooperative detection engine processes the remainder sequential mechanisms in the base station. It deals properly with dependent data of all sensor nodes, and is triggered as receiving responses of event from all sensor nodes. This system model utilizes the distinguishing features of WSNs effectively and improves the computing performance. Moreover, it adapts many variant applications easily and keeps the adaptability, scalability and flexibility.

4. **Detection mechanism.** Based on above-mentioned system model, we propose a cross-layer compromise detection mechanism which follows this model for event-driven

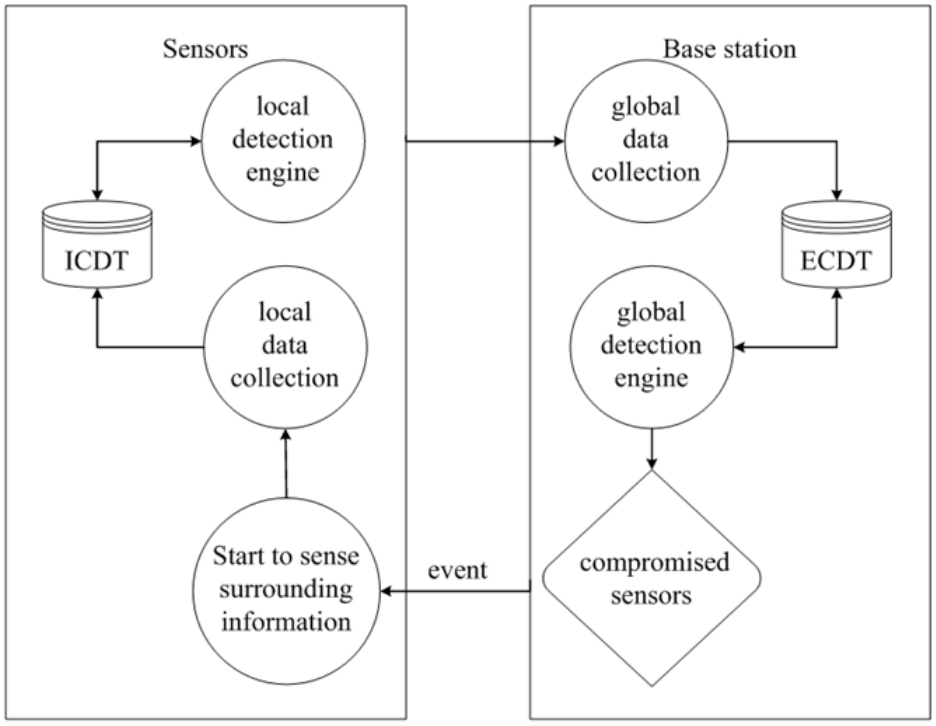


FIGURE 2. System architecture

WSNs. This mechanism can be implemented in real world and is practical for specific applications.

4.1. **Mechanism Description.** The following gives the detailed description of the mechanism we proposed. First, we classify the detection mechanisms to various types. The following lines the classifications.

- Type 1schedule checking
- Type 2Anomaly Detection Table (ADT) checking
- Type 3local measurement difference checking
- Type 4cross-node measurement difference checking
- Type 5response checking

Among these classifications, Type 1 and Type 2 detections belong to intrusion detection. And Type 3, Type 4 and Type 5 detections belong to fault detection. Overall, this detection mechanism spans across three layers of protocol layers. Type 1 and Type 2 detections are in MAC layer and routing layer respectively. The others are in application layer.

These detection mechanisms collaborate with each other in order to detect the compromised sensor nodes within all sensor nodes and the base station. They have some information to communicate with each other. Fig.3 shows the data flows of the proposed mechanism. The sensors transfer four kinds of information to the base station. We show data are generated and used in which mechanism in this figure. The notifications definition and detailed detection algorithms are described in later sections.

Second, we design the detailed information in ICDT and ECDT. The following shows the information of two table structures.

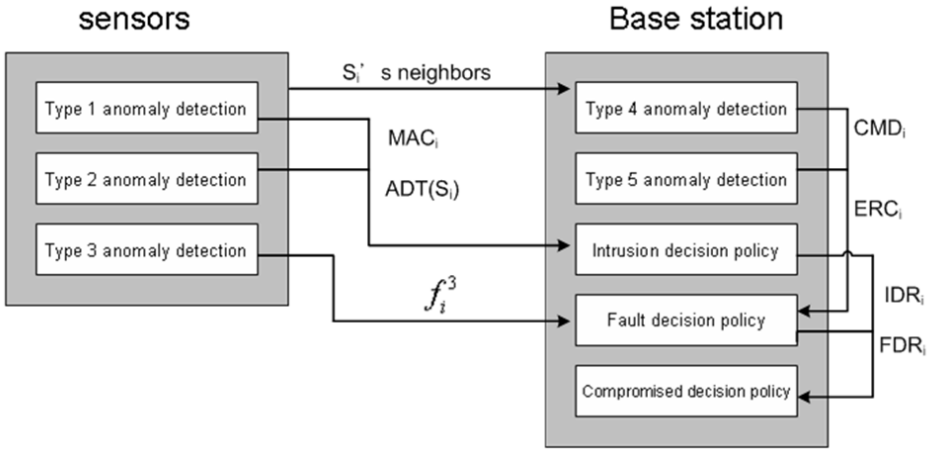


FIGURE 3. Compromise detection mechanism

TABLE 1. ICDT(Individual Compromise Detection Table)

$N(S_i)$: My neighbors
MAC_i : Type1 detection result
$ADT(S_i)$: Type2 detection result
f_i^3 : Type3 anomaly occurred ratio
$x_i^{t_l - t_{l-1}}$: Previous measurement
$x_i^{t_l}$: Immediate measurement

Besides the information and sensed data, these two table structures also store other useful information and results. All sensor nodes obtain ICDT and the base station obtains ECDT constructed by all information in sensor nodes.

4.2. **Notation Definition.** These are many notifications we use in our algorithm. And the following lines theirs definition.

- CompromisedNodes that is detected intrusion and fault is compromised
- $n$total number of sensors
- $k$number of neighbor sensors
- $N(S_i)$set of all the neighbors of S_i
- MAC_iType1 detection result MAC_i=1, 0=detected, Good
- $ADT(S_i)$set of all the sensors detected by Type2 detection
- x_i^t measurement of S_i at time t
- d_{ij}^t measurement difference between S_i and S_j at time t, $d_{ij}^t = x_i^t - x_j^t$
- $d_{ij}^t = x_i^t - x_j^t$
- $\Delta d_{ij}^{\Delta t_l}$ measurement difference between S_i and S_j from time t_{l-1} to t_l , $\Delta d_i^{\Delta t_l} = x_i^{t_l} - x_i^{t_{l-1}}$
- T_i tendency value of a sensor, $T_i \in \{LG, LT, GD, FT\}$

TABLE 2. ECDT(Entire Compromise Detection Table)

i：Sensor's ID
N(Si) ：Sensor i's neighbors
MACi：Type1 detection result
ADT(Si)：Type2 detection result of Si
f_i^3 ：Type3 anomaly occurred ratio
$x_i^{t_i - t_{i-1}}$ ：Previous measurement
$x_i^{t_i}$ ：Immediate measurement
IDRi: Intrusion detection result
ERCi: Type 5 result
CMDi: Type 4 result
FDRi: fault detection result
CDRi: compromise detection result

- θ_1 θ_2 θ_3 θ_4 Four predefined threshold values, that defined according to variant application
- w_m Three predefined weights of Type i anomaly. $\sum_{m<=3} w_m = 1$they are defined according to variant application
- f_i^3 ratio for accumulation of Type 3 anomaly occurred of S_i and detections from the event beginning,$0 \leq f_i^3 \leq 1$
- CMD_i cross-node measurement difference checking result of S_iCMDi=1,0=Fault, Good
- ERC_i response checking result of S_i ERC_i=1, 0=incorrect, correct
- IDR_i intrusion detection result of S_i IDR_i=1, 0=Intruded, Good
- FDR_i fault detection result of S_i FDR_i= 1, 0=fault, Good
- CDR_i compromise detection result of S_i CDR_i=1, 0=compromised, Good
- c_{ij} test between S_i and $S_j c_{ij} \in \{0, 1\} c_{ij} = c_{ji}$ The following sub-sections show the purposes of the notifications.

4.3. **Algorithm.** In this sub-section, we describe our detection algorithm. It classifies two engines described in the past section. These engines are Local detection engine and Cooperative detection engine. We describe these algorithms in details in the following.

4.3.1. *Local detection engine.* In the first part, we describe the Local detection engine within all sensor nodes in details. This includes three types' detection and intrusion detection policy. The following shows the details.

//Type1: Check MAC layer's information Conduct Type 1(schedule checking) that the time slot allocated to the correct sensor node.
Record Type 1 anomaly occurred form sensor j

//Type2: Check Routing layer's information Conduct Type 2 (ADT checking) that the information comes from the correct sensor node (as shown in Fig. 4 [11])

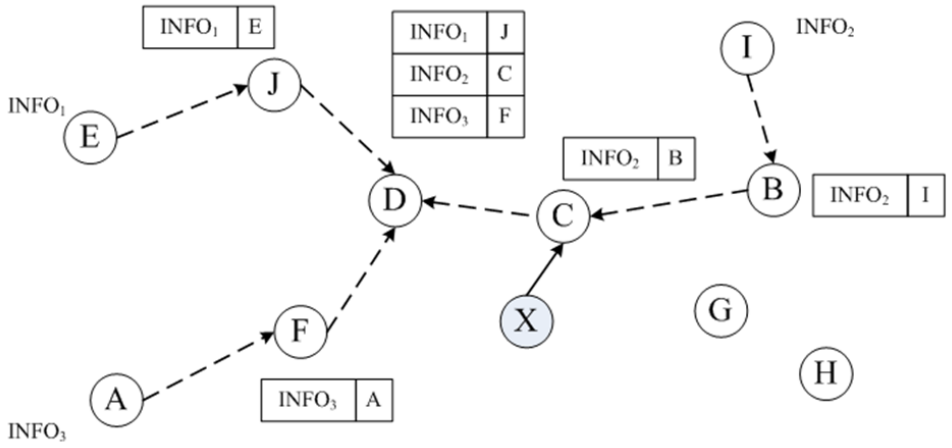


FIGURE 4. ADT checking

Step 1 : Test between each sensor S_i and every member of $S_j \in N(S_i)$ to generate test $C_{ij}\{0,1\}$ using the following method :

Set C_{ij}=0 and compute d_{ij}^t ;
IF $\left|d_{ij}^t\right| > \theta_2$ THEN
Calculate $\Delta d_{ij}^{\Delta t_l}$;
IF $\left|\Delta d_{ij}^{\Delta t_l}\right| > \theta_3$ THEN C_{ij}=1 ;

Record Type 2 anomaly occurred form sensor j

//Type3: Check sensed measurements itself Conduct Type 3(local measurement difference checking) that compare sensed measurement

IF $\left|\Delta d_i^{\Delta t_l}\right| > \theta_1$ THEN Record the ratio Type 3 anomaly occurred (f_i^3)

We have an example of Type 2 detection in the above-mentioned algorithm in Fig. 4 [11]. The Anomaly Detection Table in a sensor contains the stamp of information and which sensor will transfer the information. Sensor X is a compromised sensor. If Sensor X transfers a personate message to Sensor C, Sensor C will detect the source is an intruder.

This algorithm is processed in all sensor nodes. In addition, it also included Intrusion decision policy that diagnoses which node is intrude by attackers and stores the useful information for Cooperative detection engine and the result of the intrusion decision policy in ICDT. Then the ICDT will be delivered to the base station of WSNs.

4.3.2. *Cooperative detection engine.* In the second part, we describe the Cooperative detection engine processed within the base station in detail. This includes two types detection, Fault decision policy and the last Compromise decision policy.

//Check Type 4 anomaly (cross-node measurement difference)

The Type 4 detection in this algorithm is the modification of the distributed fault detection of WSNs proposed by Chen et al [13]. The main difference is that all processes are processed in the base station to decrease the communication overhead. The Fig.5

Step 2：Generate a tendency value T_i of S_i based upon its neighboring sensors' test value：

IF $\sum_{S_j \in N(S_i)} c_{ij} < \lceil |N(S_i)|/2 \rceil$，where $|N(S_i)|$ is the number of the S_i's neighboring nodes THEN

T_i=LG；

ELSE T_i=LF；

Step 3：Compare the number of S_i's LG neighboring nodes with different test results to determine its status：

IF $\left(\sum_{S_j \in N(S_i) and T_j = LG} (1-2c_{ij}) \geq \lceil |N(S_i)|/2 \rceil\right)$ THEN

T_i=GD；

Step 4：For the remaining undetermined sensors：

FOR i= 1 to n

IF T_i=LG or T_i=LF THEN

IF T_j=GD $\forall S_j \in N(S_i)$，THEN

IF c_{ji}=0 THEN

T_i=GD；

ELSE T_i=FT；

ELSE repeat

IF T_i=FT THEN

CMD_i=1

Step 5：IF ambiguity occurs then the sensor's own tendency value determines its status：

FOR each S_i，IF T_j=T_k=GD $\forall S_j, S_k \in N(S_i)$，

Where $j \neq k$，

And IF $c_{ji} \neq c_{ki}$ THEN

IF T_i=LG,〈or LF〉 THEN

T_i=GD〈or FT〉

IF T_i=FT THEN

CMD_i=1

//Check Type 5 anomaly (event response)

Conduct the response checking ：

IF the response is not correct type THEN

ERC_i=1 ；

ELSE ERC_i=0 ；

//Intrusion decision policy

Intrusion Decision policy for S_i ：

IF (MAC$_i$ & $S_i \in ADT(S_j)$) , $\forall j$

THEN

Set “1” for S_i at IDR_i field in the Local Compromise Detection Table

ELSE

Set “0” for S_i at IDR_i field in the Local Compromise Detection Table

//Fault decision policy

Fault Decision policy for S_i ：

IF $\left(w_1 f_i^3 + w_2 CMD_i + w_3 ERC_i\right) \geq \theta_4$ THEN

FDR_i=1 ；

ELSE FDR_i=0 ；

//Compromise decision policy

Compromise Decision policy for S_i ：

IF IDR_i & FDR_i THEN

CDR_i=1 ；

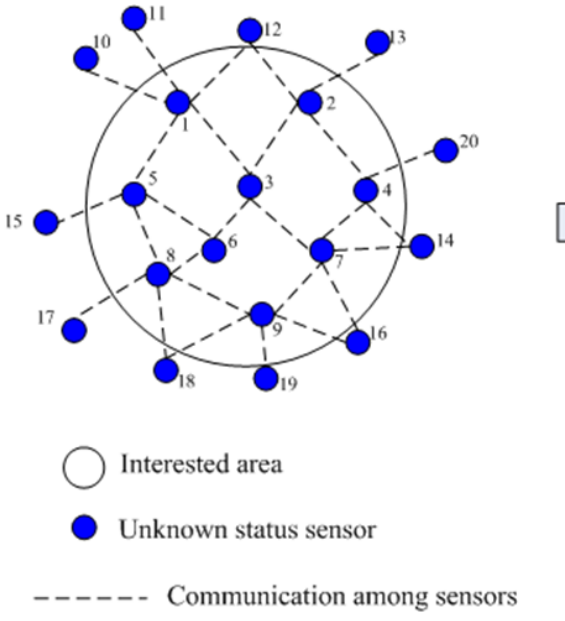
ELSE CDR_i=0 ；

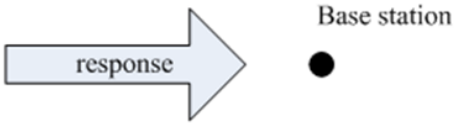
shows Type 4 detection diagram. Chen et al [13] have shown the example of Fig 5 in detail.

In Cooperative detection engine, FDR is diagnosed by Type 3, 4 and 5 detection. Then, the engine detects the compromised sensor nodes by IDR and FDR further.

5. **Simulation.** In this section, we present the simulation results for the compromise detection mechanism proposed in this work.

5.1. **Simulation Environment.** In order to characterize the basic performance of our compromise detection mechanism, we used a popular simulator, TOSSIM [6], designed





S_i	S_j with c_{ij}=0	S_j with c_{ij}=1	T_i in Iterations		
			0	1	2
1	3,5,11,12	10	LG	GD	GD
2	4	3,12,13	LF	LF	FT
3	1,7	2,6	LG	GD	GD
4	2	7,14,20	LF	LF	FT
5	1,15	6,8	LG	GD	GD
6	8	3,5,7,9	LF	LF	FT
7	3,9,14	4,6,16	LG	GD	GD
8	6,17	5,9,18	LF	LF	FT
9	7,18,19	6,8,16	LG	GD	GD

FIGURE 5. ADT checking

especially for WSN. As a comparison, we conducted experiments on the detection mechanism in the TinyOS platform. In all experiments, we assumed that the number of sensor nodes is 100, and all sensor nodes sense the temperatures around it periodically. The temperature follows a normal distribution$N(\mu, \sigma^2)$, where μ is the actual temperature at the location. Taking into account the accuracy of typical thermistors and additional wireless interference between neighboring nodes, we set $\sigma = 0.5$. We also modeled the sensor deployment distribution as a random distribution. Besides, compromised nodes may inject the uniform distribution of data and send message to a normal sensor node by 40% and 10%, respectively. We further simulated our approach by different settings and compare with other compromise detection mechanisms.

5.2. **Experiment Results.** For the application above-mentioned, we set parameters,$\theta_1 = \theta_2 = \theta_3 = 1$, $w_1 = w_2 = 0.5$, $w_3 = 0$ and diagnosed the compromised nodes after 50 data sensed. We adjusted the parameter, θ_4. In Fig.6 and Fig.7, we showed the detection rate and false positive rate of the variance in θ_4. In Fig.6, we observed that our mechanism is very accurate for this application and the detection rate is deeply affected by the number of compromised nodes, the topology and where the compromised nodes are. The more compromised nodes appeared in WSNs, the less normal nodes will guard. So when the number of compromised nodes increases, the detection rate descends. When the number of type3 detection is set between 30 and 50, the detection rates may not follow aforesaid statement. By observing the Fig.6, the less θ_4 is, the better the detection rate is. We use θ_4 to be fault decision's threshold. CDR is 1 only when both FDR and IDR are 1. If θ_4 is too large, the probability that CDR is 1 is small. So when θ_4 increases, the detection rate also descends.

In Fig.7, we also observed that our mechanism has no false positives in this environment. Only the compromised sensors intrude the normal sensors, they are detected by Type 2 detection. If a sensor detected by Type 2 detection, it will be detected into a compromised sensor. So the false positive never occurs.

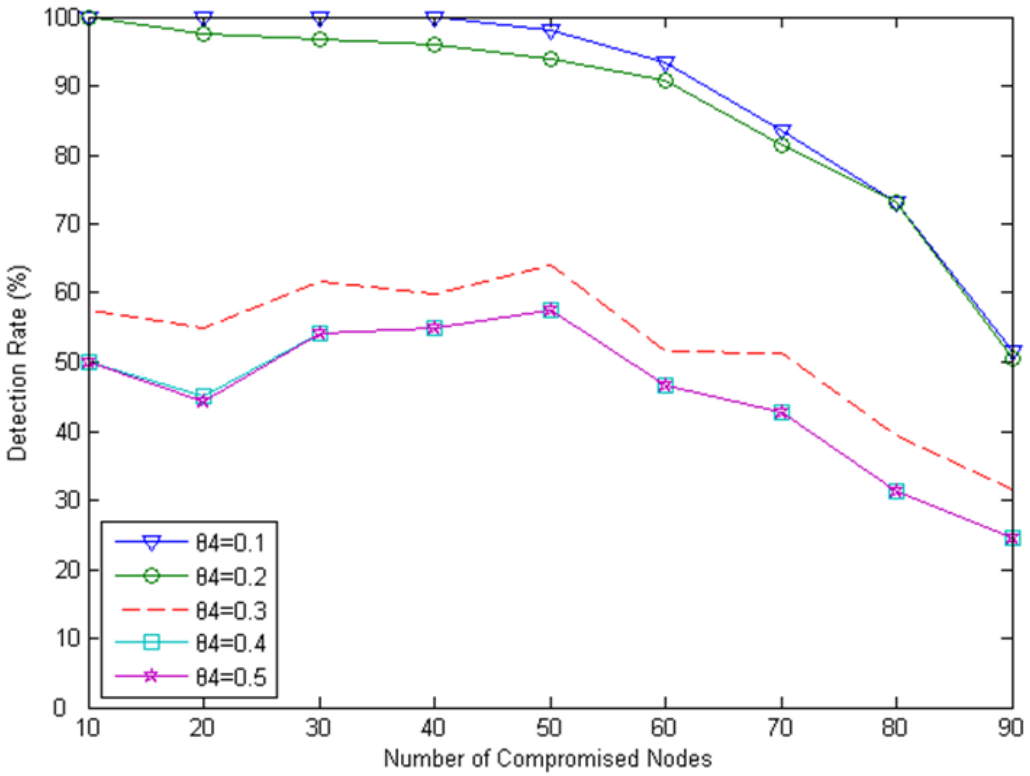


FIGURE 6. The detection rate of the variance in parameter (θ_4).

TABLE 3. My caption

Item	ZYNs algo.	CLCD
Mechanism type	Centralized	Distributed
Detection range	Non-Cross-layer detection	Cross-layer detection
Network environment	Static sensor network	All WSNs
Needed information	Topology of WSN	Not need topology information
Simulation tool	Unknown development tool	TOSSIM

We also changed the number of Type3 detection to calculate the Type3 anomaly occurred ratio. In simulation, the number of Type3 detection is 50. In Fig.8, we set θ_4=0.2 and showed the detection rate of the variance in number of Type3 detection. The number of Type3 detection makes a great impact on the total time for detecting compromised sensors. Besides, the number of Type3 detection also impacts on the local measurement difference accuracy. So, we can see the higher the number of Type3 detection is, the higher detection rate is.

Table. 3 compares CLCD with ZYN's algorithm [12]. The distributed cross-layer mechanism we propose is more efficient than ZYN's algorithm. In the third item, the neighbor table in a sensor of our mechanism is adjusted with the site of a sensor, so our mechanism will be practicable in all kinds of WSNs. And our mechanism does not need other information except for sensor's neighbor table. We also simulated in a popular simulator for WSNs, TOSSIM. Overall, our algorithm outperforms previous compromise detection algorithm and achieves high detection accuracy and no false positive rate even with a large set of compromised nodes.

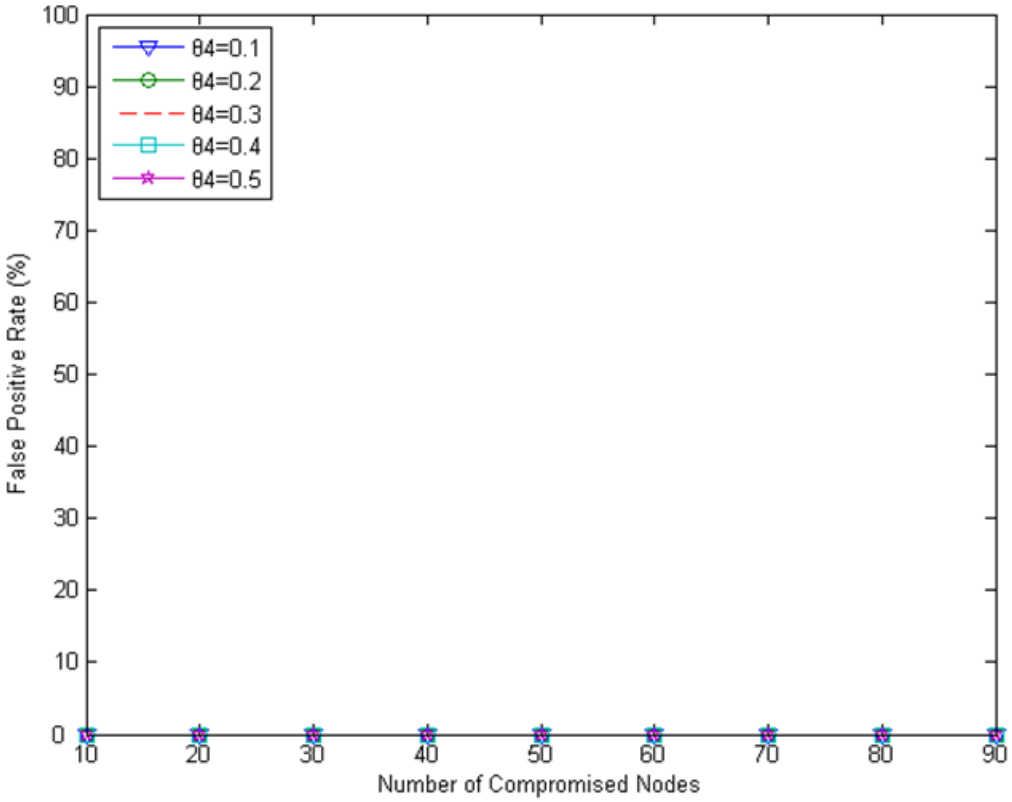


FIGURE 7. The false positive rate of the variance in parameter (θ_4).

6. **Conclusion.** Few of researches on compromise detection are proposed. So we proposed an adaptive compromise detection mechanism for WSNs. In simulation result, CLCD mechanism is very effective, accurate, scalable and practical for WSN. Despite the fact that some nodes may be compromised, the attackers do not know what the detailed meanings of messages in the compromised sensor nodes. In accordance with this point, we also proposed a compromise detection mechanism has no false positives by using CLCD mechanism.

In this paper, we use some variant detection method and form a cross-layer compromise detection mechanism by our system model. Therefore, even if there are many variant applications applied in a WSN, CLCD model is still practical with some modification.

REFERENCES

[1] F. Stajano, and R. Anderson, The resurrecting duckling: Security issues for adhoc wireless networks, the 7th International Workshop on Security Protocols, *Lecture Notes in Computer Science*, vol. 1796, 1999.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A Survey on Sensor Networks, IEEE Communication Magazine, vol. 40, no. 8, pp. 102-14, 2002.

[3] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, D. Culler, SPINS: Security protocols for sensor networks, *ACM Wireless Network*, vol. 8, no. 5, pp. 521-534, 2002.

[4] D. Liu and P. Ning, Efficient distribution of key chain com-mitments for broadcast authentication in distributed sensornetworks, *the 10th Annual Network and Distributed System Security Symposium (NDSS'03)*, February 2003, pp. 263-276.

[5] G. Gupta and M. Younis, Fault-Tolerant Clustering of Wireless Sensor Networks, *the IEEE Wireless Communication and Networks Conference (WCNC)*, 2003.

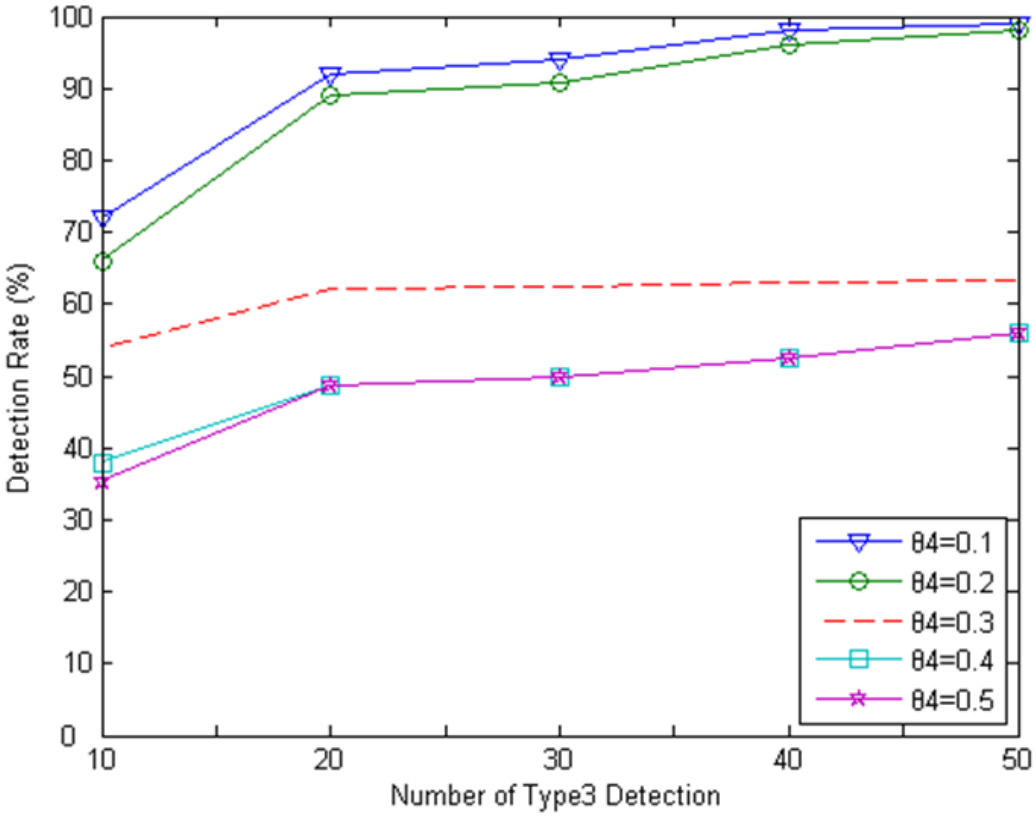


FIGURE 8. The detection rate of the variance in number of Type3 detection

[6] P. Levis, N. Lee, M. Welsh, and D. Culler, TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications, the 1st International Conference on Embedded Networked Sensor Systems, pp. 126-137, 2003.

[7] W. Du, L. Fang, P. Ning, LAD: Localization Anomaly Detection for Wireless Sensor Networks, *the 19th IEEE International Parallel and Distributed Processing Symposium*, 2005.

[8] I. Onat, A. Miri, An intrusion detection system for wireless sensor networks, *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, vol. 3, pp. 253-259, 2005.

[9] Michael Manzo, Tanya Roosta, Time Synchronization Attacks in Sensor Networks, *ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2005.

[10] A. P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. C. Wong, Decentralized intrusion detection in wireless sensor networks, *the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, ACM Press, pp. 16-23, 2005.

[11] V. Bhuse, A. Gupta, Anomaly intrusion detection in wireless sensor networks, *Journal of High Speed Networks*, vol. 15, no. 1, 2006.

[12] Q. Zhang, T. Yu, P. Ning, A Framework for Identifying Compromised Nodes in Sensor Networks, *2nd IEEE Communications Society/CreateNet International Conference on Security and Privacy in Communication Networks* , 2006.

[13] J. Chen, S. Kher, A. K. Somani, Distributed Fault Detection of Wireless Sensor Networks, *DIWANS*, 2006.

[14] R. Muraleedharan, L. A. Osadciw, Cross Layer Attacks using Swarm Intelligence, *IEEE INFOCOM-M*, 2006.

[15] W. Gu, X. Wang, S. Chellappan, and D. Xuan, Defending against Physical Attacks in Sensor Networks, *OSU-CISRC-8/05-TR56*, pp. 14.

[16] G. J. Horng, Opportunistic content sharing scheme for distributed network in city environments, *Wireless Personal Communications*, vol. 84, no. 4, pp. 2327-2350, 2015.

[17] G. J. Horng, H. T. Wu, The Adaptive Path Selection Mechanism for Solar-Powered Wireless Sensor Networks, *Wireless Personal Communications*, , vol. 81, no. 3, pp. 1289-1301, 2015.

[18] T. T. Nguyen, J. S. Pan, S. C. Chu, J. F. Roddick, and T. K Dao, Optimization Localization in Wireless Sensor Network Based on Multi-Objective Firefly Algorithm, *Journal of Network Intelligence*, vol. 1, no. 4, pp. 130-138, 2016.