Research Article

Event-Centric Situation Trust Data Aggregation Mechanism in Distributed Wireless Network

Fangwei Li, Yifang Nie, Fan Liu, Jiang Zhu, and Haibo Zhang

Chongqing Key Lab of Mobile Communications Technology, Chongqing University of Posts & Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Yifang Nie; nyf1990@yeah.net

Received 26 March 2014; Revised 3 July 2014; Accepted 9 July 2014; Published 17 August 2014

Academic Editor: Suat Ozdemir

Copyright © 2014 Fangwei Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In wireless sensor networks, the characteristics of the openness and vulnerability can lead to communication failure and node capture more easily. To cope with the deviation problem caused by the compromised nodes, this paper proposes a situation data fusion mechanism which can enhance reliability and integrality. It makes a reliability assurance scheme, which not only establishes a project distinguishing purposive and purposeless acts, but also builds a trust awareness rule based on historical trust and the correlations among current nodes. Furthermore, the data integrality is ensured in three stages, respectively. Firstly, in event detection stage, the proposed mechanism detects events and pretreats data to improve accuracy through most trust majority rule after collecting data. Moreover, during data fusion stage, data filtration is adopted to extract credible data. Additionally, the fusion weights of transactions depend on trust, so that they will change with the passage of time. As a result, the final fusion data will be closer to real value. Finally, in consistency detection stage, the detection rule with lower communication traffic is utilized to judge whether the centers are credible. Theory analyses and simulation results verify that the new mechanism can reduce the deviation. Meanwhile, reliability and stability have been validly enhanced.

1. Introduction

Owing to the wide coverage, the sharp increase of network flow, the huge amounts of data [1], and the frequent interactions and communications among network resources, future wireless sensor networks need situation data fusion mechanism to manage. Situation awareness [2, 3], such as security and survival situation awareness, regards data fusion as center, so that its model depends on fusion model [4, 5]. Its system structure can be classified into 5 sections: detection, perception, prediction, evaluation, and decision. Those networks with limited sources and store capacities, such as P2P, MANETs, and ad hoc, joining data fusion, can lead to the characteristics of openness, dynamics, and vulnerability, which can easily cause communication failure and node capture. The captured issues affecting the output of aggregators can be classified into two categories: (1) information is falsified in the process of transmission and (2) after information arrives at node, data is falsified by node. Encryption and authentication techniques can deal with the problem in transit, while this paper mainly considers how to fuse the situation data after nodes are compromised. The adversaries want to cause the deviation between fusion value and real value by 3 ways in [6]: (1) fabricating terminal data to misadvice higher center, (2) falsifying or tampering fusion data of center, and (3) hiding information and not telling other nodes related information.

It is an effective measurement building trust model to reduce the influence caused by capture attacks. As a result, many methods based on trust come into being, whereas there is a focus on finding a secure and credible aggregation model. At present, many data fusion researches based on trust [7, 8] hardly consider historical and time correlation. They fuse data with the same transaction weight or with fixed weight.

The aggregation method proposed in [9] regards MEDIAN function as reference to fuse data, which is more reliable than MAX, MIN, AVG [10], and SUM fusion function, because adversary can easily affect the SUM, MAX, and MIN value by changing some data reports. The network aggregation scheme in [11, 12] ensures data integrity and
reliability, but it depends on SUM, MAX, or MIN and fusion weights are fixed. It cannot cope with sortie attacks and needs to tackle the detectable evil nodes. The aggregation method based on reputation in [13] supposes true and false data number yield to beta distribution, assuming that only terminals can be captured, which is unpractical, because center also can be captured. The average fusion method (AFR) proposed in [10] refers to AVG as benchmark. After handling data, it uses AVG to fuse data. Heterogeneous neuron fusion (BPR) in [14] regards terminals as neurons and considers fusion as the processing of handling neuron data by primary function. But after fusion, the deviation is big. Above methods can respond known attack and hardly consider time and historical correlation, which also lack consistency rule to detect reliability of node.

This paper proposes a fusion mechanism MCMR based on trust, not only supposing that terminals can be captured, but also considering that centers can be compromised. It considers historical and time correlation and draws up situation trust awareness rule on historical trust and current data correlation. The mechanism contains 3 periods: when detecting events, relative nodes upload data reports to centers, and then, centers judge whether events have happened by most trust majority rule MMR. If they ensure that given events have happened, they will pretreat data before fusing data. Next, they fuse data with the transaction fusion weights related to trusts. Finally, during consistency detection, it utilizes consistence detection rule RSR to verify reliability of centers and issues the final value.

This paper is organized as follows. Section 2 introduces the system model. Then, Section 3 introduces situation data fusion mechanism MCMR. In Section 4, we introduce the reliability assurance scheme of mechanism. Section 5 analyzes simulation and evaluates performance. At last, we summarize the paper and point out future directions.

2. System Model

This situation fusion mechanism has 3 steps: event detection, situation data fusion, and consistency detection. At first, the mechanism picks up the relative terminal sensor nodes and central nodes of event. Then, after all relative sensor nodes collect data, they will send data to their upper centers. Next, after centers have fused and tackled data, they will send data to base station. Finally, base station detects consistency of centers and issues the fusion value. The mechanism network model is presented in Figure 1. There are only 3 kinds of nodes in the network: terminal node, center, and base station. The detection coverage of terminal node is a circle with radius . The detection coverage of center is a circle with radius , and belong to , . The detection coverage of center is a circle with radius , and belong to , . All sensor nodes should cover the area across the board. Meanwhile, all centers should cover all terminal nodes. The base station can communicate with all nodes and tackle and store data. Centers can both gather and tackle data. Terminal nodes can just collect data. Sensor nodes can detect events in detection coverage.

Figure 1: Mechanism network model.

As is shown in Figure 1, events occurred simultaneously. The relative sensor nodes of event are , , , and . The relative sensor nodes of event are , and . Hence, node detects the events and simultaneously. are the abnormal nodes. If an event has happened in the coverage of a node, we define the node as relative node to this event. If there are centers among relative sensor nodes, the centers become relative centers, directly, for example, , to event . If there are no centers among relative sensor nodes, the centers having the relative terminal nodes will become relative centers, for example, , to event . The total collection of all nodes is , which contains all nodes in the network.

3. Situation Trust Data Aggregation Mechanism

3.1. Event Detection. Event detection mainly decides whether the contingent event uploaded by sensors has happened or not. Generally, if an event has happened, we can consider its state as 1, or we will consider its state 0. There are several methods to decide the state of event, such as “AND” rule, “OR” rule, “majority” rule. To “AND” rule, only if a sensor says the state is 1, it judges state as 1. To “OR” rule, only if a sensor says the state is 0, it judges state as 0. To common majority rule (CMR), if more than half of sensors say the state is 1, it judges state as 1. In persistent period , the arrival number of events yields to the random Poisson distribution, so that several events that occur at the same time can be considered as events occurring in slot time . The collection of contingent occurrence events is . The collection of probable relative nodes is
\( \mathbf{N}' = [\mathbf{n}_1^{N_1}, \ldots, \mathbf{n}_k^{N_k}, \ldots, \mathbf{n}_{N'}^{N'}] \), where the number of probable relative nodes to event \( e \) is \( N_{e}' \), and \( w' \) is the number of contingent concurrence events.

In this part, this mechanism judges the state of event by MMR, which is an improved rule of “majority” rule as follows.

First, trust filtration: the filtration function is \( \vartheta(\theta_{i,e}) \rightarrow [0,1], i \in \mathbf{N}', e \in \mathbf{E}' \). If \( \theta_{i,e} > \bar{\theta} \), then \( \vartheta(\theta_{i,e}) = 1 \), or \( \vartheta(\theta_{i,e}) = 0 \) and vice versa, where \( \theta_{i,e} \in [0,1] \) is the trust of node \( i \) to event \( e \) and \( \bar{\theta} \) is the threshold of trust filtration.

Second, individual detection: the individual detection function is \( f : \delta_{i,e} \rightarrow [0,1], i \in \mathbf{N}', e \in \mathbf{E}' \). If node \( i \) finds \( e \) have happened, then \( \delta_{i,e} = 1 \), or \( \delta_{i,e} = 0 \).

Finally, base station decision: decision function is \( f : \mathbf{E} \rightarrow [0,1] \).

The concurrence event collection is \( \mathbf{E} = [e_1, e_2, \ldots, e_w] \). The event detection is given in Procedure 1.

In system model, \( \mathbf{E} = [e_{x}, e_{y}] = [x, y] \), \( w = 2 \), because there are 2 events concurrently, event \( x \) and event \( y \). The relative node collection can be expressed as \( \mathbf{N} = [\mathbf{n}_1^{N_1}, \ldots, \mathbf{n}_k^{N_k}, \ldots, \mathbf{n}_{N'}^{N'}] \), where the total number of relative nodes to event \( e \) is \( N_{e}' \) and \( w' \) is the number of events. The concurrence event collection is \( \mathbf{E} = [e_1, e_2, \ldots, e_w] \). The event detection is given in Procedure 1.

Proposition 1. If the probability of evil nodes eliminated by MMR is \( I > 0 \), the average probability of attack success of MMR, \( P_{FM} \), is smaller than that of common majority rule (CMR), \( P_{FG} \); concisely, \( P_{FM} \leq P_{FG} \).

Proof. To \( w \) concurrence events, the false detection probability [15] can be expressed as

\[
P_f = P \left\{ \frac{1}{w} \sum_{k=1}^{w} N_k^2 > \eta \right\} = 1 - P \left\{ \sum_{k=1}^{w} N_k^2 \leq w \eta \right\},
\]

where \( N_k^2 \) is the noise variance, \( \eta \) is the threshold, and \( \sum_{k=1}^{w} N_k^2 \) obeys the Chi-square distribution with degree of freedom \( w \). The cumulative distribution function is \( \chi^2_{N'}(\cdot) \), and, therefore, we know the true detection probability is \( 1 - p_f > 0.5 \), \( p_f = 1 - \chi^2_{N'}(w \eta) \). If \( H \) is the number of nodes joining CMR and the number of malicious nodes is \( X \leq H \), we can know the number of nodes joining MMR is \( F = H - X \cdot l \),

\[
0 < F < H. \text{ To MMR, we get the average probability of attack success:}
\]

\[
P_{FM} = \sum_{r=F/2}^{F} \left( \frac{F}{r} \right) (1 - p)^{F-r}(p)^r, \quad p = p_f.
\]

\[
P_{FG} = \sum_{r=0}^{F/2} \left( \frac{H}{r} \right) (\lambda^2 w \eta)^{r} (1 - \lambda^2 w \eta)^{H-r}.
\]

To CMR, we can also obtain the average probability of attack success:

\[
P_{FG} = \sum_{r=0}^{F/2} \left( \frac{H}{r} \right) (\lambda^2 w \eta)^{r} (1 - \lambda^2 w \eta)^{H-r}.
\]

Therefore, \( P_{FM} \leq P_{FG} \).

3.2. Data Resilient Aggregation. Data fusion is intended to know the concrete data of event. If the event is fire emergency, the data may be the temperature and smoke scope. Fusion methods in [9, 10, 13, 14, 16] presented how to fuse data but did not pretreat data before data fusion. Situation data fusion proposed in this paper considers how to pretreat data, which contains 3 stages: prior judgment, data filtration, and data fusion.

The first stage is prior judgment: when there are several events that happened simultaneously. It processes event data in chronological order. If several events have occurred, the base station will preferentially tackle event with biggest prior. Other events with lower prior will wait until that has been finished. If only one event has happened, it can enter data filtration directly.

Definition 2 (prior judgment rule). To concurrence events in \( \mathbf{E} \), there is priority \( \rho = [\rho_1, \ldots, \rho_{k}, \ldots, \rho_{w}] \), where \( \rho_k \in (0, 1) \). The mechanism selects and handles the event with biggest prior weight preferentially, according to formula \( \max_{e \in \mathbf{E}} I_{e} \), where \( I_e = \sum_{c \in e} \rho_c \).
In system model, to events $x$ and $y$, if $I_x < I_y$, the mechanism will handle event $y$, preferentially.

The second stage is data filtration: pretreatment of event data.

**Definition 3** (data filtration rule). To current event $e$, if all relative nodes in $n_{N_x,e}$ belong to $L$ relative centers, $L$ centers will constitute collection $C_e = \{A_1, A_2, \ldots, A_L\}$. If node $i \in n_{N_x,e}$ belongs to relative center $A_v \in C_e$, $v = 1, \ldots, L$, it can come into collection $O_{i,e} \subset V_e$, etc. Meanwhile, the center $A_v \in C_e$ rejects these nodes not meeting “condition 1,” with the probability $1 - \epsilon$. In other words, these nodes not meeting “condition 1” can enter data fusion, with the probability $\epsilon$.

To the center $A_v \in C_e$, after data filtration, the nodes in $O_{i,e}$ meeting “condition 1” constitute node collection $F_{i,e}$, and those nodes in $O_{i,e}$ which can enter data fusion will constitute node collection $H_{i,e}$. We record the size of $O_{i,e}$ as $m_{i,e}$. To the relative center $A_v \in C_e$, if there are $M_{i,e}$ nodes meeting “condition 1,” $F_{i,e}$ will have $M_{i,e}$ elements. We round $M_{i,e} + (m_{i,e} - M_{i,e})\epsilon$ to the nearest integer, and $H_{i,e}$ will have round $(M_{i,e} + (m_{i,e} - M_{i,e})\epsilon)$ elements. The error frequency of node $i$ in collection $O_{i,e}$ is $num_i, i \in O_{i,e}$. To event $e$, the data filtration procedure of center $A_v \in C_e$ is given in Procedure 2.

In system model, to event $y$, we can get $C_e = \{c_1, c_2\}$ is $\{A_1, A_2\}, L = 2, O_{1,y} = \{c_1, r_3\}, O_{2,y} = \{c_2, r_8\}$. To current event $e$, the center $A_v \in C_e$ can get the size of $O_{i,e}$, and data $\text{data}_{c_i,\text{med}} = (\text{data}_{c_i} + \text{data}_{r_3})/2$, $\text{data}_{c_i,\text{med}} = (\text{data}_{c_i} + \text{data}_{r_3})/2$. To event $y$, there is no abnormal node, and we suppose $H_{1,y} = O_{1,y}$, $H_{2,y} = O_{2,y}$.

The third stage is data fusion: the data fusion rule is as follows.

**Definition 4** (data fusion rule). The data yields to uniform distribution, according to equilibrium theory, multiplying data inverse function, $\gamma : f(x) = \sqrt{2\pi}\sigma \exp(-(x - u)^2/(2\sigma^2))$, where the real value is $u$. This can get the data in unified form. To current event $e$, according to the data, and the trust $\theta_{i,e}$ of node $i \in H_{i,e}$, relative center $A_v \in C_e$ can get fusion value:

$$d_{A_v,e} = \frac{\sum_{i \in H_{i,e}} \theta_{i,e} \text{data}_{c_i}}{\sum_{i \in H_{i,e}} \theta_{i,e}}.$$  

(6)

Generally, based on the trusts $\theta_{A_v,e}$, $A_v \in C_e$, the mechanism can compute the center fusion value:

$$BAd_c = \frac{\sum_{A_v \in C_e} \theta_{A_v,c} d_{A_v,e}}{\sum_{A_v \in C_e} \theta_{A_v,c}}.$$  

(7)

3.3. Consistency Detection

**Definition 5** (consistency detection rule (RSR)). After base station accepts the fusion value of relative centers in $C_e$, it will begin to check consistency of centers, by selecting $\lambda$ terminal nodes belonging to relative centers in $C_e$ to send...
data, randomly. $\lambda$ nodes compose collection $S_v$. They shape the fusion data $BSd_e = \text{med}([data_i], i \in S_v)$. If $|BSd_e - BAd_e| < \Delta data$, the station will think that all centers in $C_e$ are reliable, where $\Delta data$ is the correlation threshold. And then, it regards $BAd_e$ as final fusion value. Otherwise, the consistency detection is failed, and the base station will repeat the operation to detect consistency. In excess of the maximum number $\kappa$ of replication, it will cut down trusts of centers not meeting condition 2: $|BAd_e - \text{med}(d_{A_v})| < \Delta data, A_v \in C_e$, and regards $BSd_e$ as final fusion value, where $d_{A_v}$ is the data of all $d_{A_v}, A_v \in O_{\alpha e}$, and $\text{med}(d_{A_v})$ is the median of $A_v \in C_e$. The error frequency of each center $A_v$ is $\text{num}_{A_v}$. To event $e$, the consistency detection procedure is as Procedure 3.

In system model, to event $e$, at the consistency detection stage, the base station will select $\lambda$ terminal nodes from the lower level nodes $r_5, r_4, r_6, r_8, r_11$, and $r_{12}$ to detect the consistency of $c_2, c_4$.

Presently, most researches hardly have considered consistency detection. This mechanism has used consistency detection RSR to detect the consistency of center. Compared with general overall search detection (GAS), RSR has lower complexity and workload. If the number of nodes joining data fusion is $S$ and the sampling number of nodes in consistency detection is $I$, we can see the complexity of GAS is $O(S^2 + S/2)$ and that of RSR is $O(IS + S/2)$. The ratio of the former to the latter is $S/I$. We know the reliability of GAS is close and lesser than 1 and that of RSR is $1 - \binom{k}{i} \cdot \left(\frac{n-k}{m-1}\right) / \binom{n}{m}$. The mechanism flow chart is like Figure 2.

### 4. Reliability Assurance Scheme of Mechanism

In order to ensure its reliability, this mechanism especially builds reliability assurance scheme from two aspects as follows:

(i) establishing trust awareness rule on historical value and current data correlation;

---

**Procedure 2: Data filtration ($O_{\alpha e}$, $\text{med}(d_{A_v})$, $\Delta d, \epsilon$).**

**Input:** $\Delta$ data, $\lambda, \kappa, \text{med}(d_{A_v}), BAd_e$, and the collection $C_v$  

**Output:** adjust the trust of relative center $A_v \in C_v$

1. for $j = 1, \ldots, \kappa$ do
2. base station selects $\lambda$ terminal nodes of centers in $C_v$ to send data randomly
3. $\lambda$ nodes send data to base station
4. station computes $BSd_e = \text{median}(data, i \in S_v)$
5. if $|BSd_e - BAd_e| < \Delta data$ then
   6. base station considers all centers in $C_v$ are reliable and regards $BAd_e$ as final fusion value
7. end if
8. if $k = \kappa$ then
9. station regards $BSd_e$ as final fusion value
10. for $k = 1, \ldots, \text{size}(C_v)$ do
11. if $|BAd_e - \text{med}(d_{A_v})| < \Delta data, A_v \in C_v$ then
12. station reduces the trust of center $A_v \in C_v$
13. end if
14. end for
15. end if
16. end for
(ii) building project distinguishing purposive acts and nonpurposive acts.

4.1. Trust Awareness Rule of Mechanism. According to historical trust and current data correlation, the mechanism adjusts trust. If several events occur at the same time, the trust renewal will refer to prior judgment. If there is a new event, it will wait until current event has finished. Owing to evil acts of terminal node, the trust will be reduced by 0.1, but for that of center the trust will be decreased by 0.2. The initial trust of terminal sensor node is 0.8, but that of central node is 1. If the current event is not in the coverage of nodes, the mechanism will retain their trusts. All trusts return initial states, every executing fusion 20 times. In several stages, trust updating is as follows.

(i) To event $e$, owing to the existence of concealed attack, in event detection, to all relative nodes, only relative nodes in $N$ having the same event detection result as their center will retain their trusts; however, others will reduce by $\Delta \theta$. The other irrelative nodes will retain their intrinsic states.

(ii) To event $e$, owing to the existence of concealed attack, in event detection, only relative nodes in $N$ having the same event detection result as their center will retain their trusts; however, others will reduce by $\Delta \theta$. The other irrelative nodes retain their intrinsic states. To event $e$, according to the error frequencies, the mechanism will reduce trusts of relative nodes not meeting “condition 1” in data fusion. Because of the error frequency $num_i, i \in O_{\epsilon,x}$ of each node $i$ in collection $O_{\epsilon,x}$, the trust of node $i$ will be updated as

$$\theta_{i,e} = \theta_{i,e} - 0.1 \cdot num_i.$$  

(iii) To event $e$, in consistency detection, the base station compares $BAd_{e}$ with $BSd_{e}$. If $|BSd_{e} - BAd_{e}| < \Delta data$, it will retain the trusts of relative centers in $C_e$. If $BAd_{e}$ and $BSd_{e}$ do not always meet the condition $|BSd_{e} - BAd_{e}| < \Delta data$ and the repeat count exceeds the maximum number $\kappa$, the system will cut down trusts of relative centers in $C_e$ not meeting condition $2:|BAd_{e} - \text{med}(d_{A_{\epsilon,x}})| < \Delta data, A_{\epsilon,x} \subset C_e$. Owing to the error frequency $num_{A_{\epsilon,x}}$ of each center $A_{\epsilon,x} \subset C_e$, the trust of $A_{\epsilon,x}$ will be adjusted to $\theta_{A_{\epsilon,x}} = \theta_{A_{\epsilon,x}} - 0.2 \cdot num_{A_{\epsilon,x}}$.

(iv) After the mechanism has released the final fusion value, it begins to adjust trust. To all nodes having joined the fusion of current event $e$, the association degree of arbitrary two nodes between $i, i \in H_{\epsilon,x}$ and $q, q \in H_{\epsilon,x}$ is $\xi_{iq}$. If $|data_i - data_q| < \Delta data$, $\xi_{iq} = 1$, or $\xi_{iq} = 0$. To arbitrary node $i, i \in H_{\epsilon,x}$, its historical trust is $\bar{\theta}_i$. Time decay factor is $\mu$. The trust of node $i$ will be updated as

$$\theta_{i,e} = \mu \bar{\theta}_i + \frac{(1 - \mu)}{M_{\epsilon,x} + (m_{\epsilon,x} - M_{\epsilon,x}) \epsilon} \sum_{q=1}^{M_{\epsilon,x}} \xi_{iq}.$$  

4.2. Distinguishing the Purposive Acts and Nonpurposive Acts. We call the accidental false actions nonpurposive act, while
Table 1: Parameter setting.

| Name | Description | Default |
|------|-------------|---------|
| \( N \) | The whole number of nodes | 100 |
| \( C \) | The number of centers | 4 |
| \( R_1 \) | The sensing radius of terminal | 1 m |
| \( R_2 \) | The sensing radius of center | \( 2\sqrt{2} \) m |
| \( ma_{per} \) | The percentage of time when a malicious node starts the attack | 100% |
| \( w \) | The number of concurrence events | 0 |
| \( (-\varepsilon, \varepsilon) \) | The error range of each node | \((-0.05, 0.05)\) |
| \( q \) | The percentage of compromised reports to total reports | 0% |
| \( \hat{\theta} \) | The lowest trust of system | 0.3 |
| \( \mu \) | Time decay factor | 1 |
| \( \Delta d \) | Data filtration threshold | 5 |
| \( \Delta data \) | Correlation threshold in consistency detection | 5 |
| \( \kappa \) | The maximum number of replication in consistency detection | 10 |
| \( \lambda \) | The percentage of attacks to total nodes | 20% |

the false reports based on the equipment failure and compromising nodes we call them purposive act. The probability of normal node appearing \( m \) consecutive errors is \((1 - \varepsilon)^m\), but that of malicious node is \( \varepsilon^m \), because of the manipulated data. Hence, the ratio of the former to the latter is about \((\varepsilon/(1 - \varepsilon))^m\). The probability of malicious node far outweighed that of normal node.

Therefore, it can distinguish the two kinds of acts based on the consecutive error number. Generally, it defines \( s \) consecutive errors as the purposive act. \( s \) can be adjusted.

5. Performance Evaluation and Simulation Analysis

In order to evaluate the mechanism performance and prove the applicability and reliability, we simulate the situation data fusion mechanism relying on the above theoretical frame. We consider shadow fading and free path loss when communicating in wireless environment. The hardware experiment environment is Intel Core (TM) Duo 2.66 GHz CPU, 2 GB Memory, Windows XP operating system, MATLAB 7.0 simulation platform. Nodes distribute in 110 m*110 m area symmetrically, with cluster-class method in [18]. Nodes distribution is approximate to uniform distribution. The parameter setting is as in Table 1.

We analyze overall and local performance. Overall performance: damage degree, stability, and trust situation. Local performance: probability of attack success of MMR in event detection, complexity, and reliability of RSR in consistency detection.

5.1. Overall Performance Evaluation of Mechanism. We chose algorithms which are as follows: average function fusion rule (AFR) [10] and heterogeneous situation neuron fusion rule (BPR) [14]. Comparing MCMR with above two rules, we describe deviation degree and stability in Figures 3 and 4, respectively. Moreover, we operate them to compute data with recommended parameters and primary function.

5.1.1. Average Function Fusion Rule (AFR). AFR [10] considers average as primary value when fusing data. After that, it computes fusion value by AVG function. AFR may eliminate some malicious nodes but fail to detect adversaries with abnormal data, which can cause reference average increasing, finally provoking sharp deviation. Meanwhile, the fluctuation of error happens. MCMR can reject abnormal data to improve stability.

5.1.2. Heterogeneous Situation Neuron Fusion Rule (BPR). BPR [14] regards lower nodes as neurons, defining fusion computing as a process from input to output. The input is processed by primary function, geometric mean function. The weights of neurons adjustment and thresholds adjustment based on last value and error both continuously change. Moreover, there is no strategy to cope with sudden data change due to the lack of data filtration and preprocessing. The deviation also uncontrollably fluctuates.

5.1.3. Damage Degree Evaluation. Owing to many uncertain factors, the mechanism may have damage error inevitably. We define damage degree as the deviation between fusion value and real data. The bigger the damage degree is, the bigger the deviation is. The measure norm is mean absolute degree of damage percentage (MADDP). Consider

\[
\text{MADDP} = \frac{1}{N} \sum_{n=1}^{N} \left| \frac{\text{fusion}_n - \text{real}}{\max(\text{real}, \text{fusion}_n)} \right|, \tag{9}
\]

where fusion\(_n\) is \( n \)th fusion value, real is real value, and \( N \) is fusion frequency.

The simulation result is like Figure 3, after simulating MCMR, MFR [9], AFR [10], and BPR [14] for 25 times under attack and no attack environment, respectively. From Figure 3(a), we know when there is no attack, MADDP of MFR will be lowest, being close to 0. There are heaves at 13th and 19th events, because terminals without communication have been selected in consistency detection, thus causing damage degree increasing. From Figure 3(b), we can see when
there are attacks, MADDP of MCMR will be lowest. AFR is from 0.2 to 0.5. MFR is from 0 to 0.5, but BPR is from 0.6 to 0.95. Because attacks lie in 3rd, 8th, 13th, 15th, and 18th events, MADDP of these four methods increases. Therefore, when there are attacks, MADDP of MCMR is the lowest. When there is no attack, MADDP of MFR is the lowest.

5.1.4. Stability Evaluation. The mechanism regards the minimum mean square error (MMSE) [19, 20] as norm to evaluate stability. Adversaries want to capture nodes to cause MMSE increasing. MCMR can filter out abnormal ones to reduce MMSE of mechanism. According to MADDP of MCMR, MFR, AFR, and BPR, we can obtain variances under attack environment, respectively, as follows: 0.026, 0.073, 0.144, and 0.652. In summary, we can get the performance comparison as in Table 2.

According to the detection error of per node $e_i \sim U(-\epsilon, \epsilon)$, the mean $\epsilon^2/3$, the variance $4\epsilon^2/45$, and the mean square error $\frac{c^2}{n} = \frac{1}{n} \sum_{i=1}^{n} e_i^2$, if the number of nodes joining fusion is $n$ and the capture ratio is $\alpha$, we can know $\lim_{n \to \infty} P(\epsilon^2 - u)/\sigma \leq \lim_{n \to \infty} F_n(u)$ yields to normal Gaussian distribution, where $\epsilon_0 = k \epsilon, u = \sum_{i=1}^{n} u_i^2$, and $\sigma = \sqrt{\sum_{i=1}^{n} \sigma_i^2}$, so $\lim_{n \to \infty} F_n = \Phi(\sqrt{5n/2} * (3k^2 - 1 - n\alpha))$. Owing to the fluctuation and uncertainty of BPR, the experiment compares MMSE of AFR and MCMR on the condition of $n = 50, \alpha = 0.2$, and $n = 50, \alpha = 0.9$, respectively. Then, we gain the average performance after operating it 25 times as in Figure 4. As is shown in Figure 4, MMSE of MCMR is relative concentration, and it is from 0.5$\epsilon$ to 1.2$\epsilon$. We can know it rises with the increasing of compromised ratio. However, MMSE of AFR is at the range of 3$\epsilon$ to 4.6$\epsilon$.

5.1.5. Trust Situation of Mechanism. The mechanism depends on trust to assign weight during data fusion, and it evaluates trust by historical trust and data correlation. It separates trusts into ABCD levels like Table 3.

The simulation visualizes the misrepresentation and data outlier as in Figure 5. Figure 5(a) shows trust situation after simulating 25 events without attack by MCMR. The gray areas

![Figure 3: Situation of damage degree (MADDP).](image)

![Figure 4: The mean square error, nodes, and capture ratio.](image)

![Table 2: Performance comparison.](image)

- **Table 2: Performance comparison.**
  | Algorithm | MCMR | AFR | BPR | MFR |
  |-----------|------|-----|-----|-----|
  | MADDP with attacks | 0–0.1 | 0.2–0.5 | 0.6–0.9 | 0–0.5 |
  | MMSE of simulation | 0.026 | 0.144 | 0.652 | 0.073 |
in the upper left are the learning process. Figure 5(b) is trust situation after simulating 25 events with misrepresentations by MCMR. There are always misrepresentations in 14th and 21st nodes, so their trusts reduce from 0.8 to 0.3, gradually. And finally, after their trusts are lower than $\tilde{\theta}$, they will be out. Figure 5(c) displays data outlier distribution situation of 25 events, orderly. The gray areas represent attacks that are not serious in these areas, but the black areas represent attacks that are serious. Figure 5(d) is trust situation after simulating 25 events with outlier distribution in Figure 5(c) by MCMR. Because there are several consecutive errors in the 7th and 10th nodes and sporadic errors in others, such as 23rd, 30th, 31st, and 32nd nodes, trusts of the former nodes obviously reduce to 0.3, so they will be removed. Whereas trusts of the latter nodes decrease, they are preserved.

Hence, trust situation of the mechanism is the representation of node action. It may skip over some outliers but can visualize trusts of nodes with misrepresentations and data outliers, accurately. The intendant can handle the evil actions and encourage honest actions with the trust situation.

5.2. Local Performance of Mechanism

5.2.1. Probability of Attack Success of MMR in Event Detection Stage. If parameters are $H = 30$, $p_f = 0.05$, and $w = 10$, we can get Figure 6. Because of the influence of majority rule, we suppose the percentage of compromised reports to total reports is $q$, and after $q$ is more than 50%, the attack success probability will obviously increase. From Figure 6, we can see

**Table 3: Trust level.**

| Trust | Trust level | Remark      |
|-------|-------------|-------------|
| 0–0.3 | A           | Unlikelihood|
| 0.3–0.5| B           | Not very likelihood |
| 0.5–0.8| C           | Probable    |
| 0.8–1.0| D           | Reliable    |

*If the trust level of node is A, it will be eliminated. Only nodes trust level higher than B can access.*
the error probability of attack success of \( l = 0.5 \) is bigger than that of \( l = 0.75 \). Therefore, the probability of attack success increases with the increasing of \( q \). The bigger the ratio \( l \) is, the smaller the probability of attack success is. During event detection, because of the application of MMR, the probability of misrepresentation attack success reduces.

5.2.2. Complexity and Reliability Evaluation of RSR in Consistency Detection. If the maximum number of nodes joining data fusion is \( S = 100 \) and the sampling number of nodes in consistency detection \( I \) is from 0 to 100, we can get complexity as in Figure 7(a). Because partial nodes join in consistency detection, RSR has lower complexity. Figure 7(b) is the reliability with \( \lambda = 20\% \) and \( t \) the sampling number of nodes in consistency detection \( I = 20 \). Compared with GAS, RSR has lower reliability with no or few attacks, which is the cost of reducing complexity. However, when the sampling number of attacks \( I \) is bigger than 5 in consistency detection, the reliability of RSR is higher than that of GAS. Therefore, RSR is more appropriate to the condition with more attacks.

6. Conclusion
To cope with the deviation problem caused by compromised nodes in distributed network, this paper proposes situation data fusion mechanism based on trust, making trust awareness rule to improve reliability and processing data from 3 stages: event detection, data fusion, and consistency detection. The deviation between fusion value and real value can be reduced when there are attacks. Because of the application of MMR and the additional consideration about RSR, the reliability can be enhanced. In general, if combining with encryption techniques [6], it can improve performance, especially in big data domain, such as environmental monitoring. How to take strategy to handle abnormal actions may be the research of future work.

Appendix
Simulation figures: from Figure 3 to Figure 7.

Conflict of Interests
The authors declare that there is no conflict of interests regarding the publication of this paper.
Acknowledgments

This work is supported by the National Nature Science Foundation of China (no. 61272620, no. 61102062, no. 61301122), NSF of Chongqing (no. cstc2014jcyjA40052), the Research Program of Chongqing Municipal Education Commission (no. KJ1400405), Program for Changjiang Scholars and Innovative Research Team in University (IRT1299), the special fund of Chongqing key laboratory (CSTC), NSF of CQUPT (no. A2013-30), and the Doctor Science Research Starting Foundation of CQUPT (no. A2013-23).

References

[1] L. D. Xin and S. Divesh, "Big data integration," in Proceedings of the 39th International Conference on Very Large Data Bases, vol. 6, pp. 1188–1189, Trento, Italy, 2013.

[2] A. Coronato and G. de Pietro, "Situation awareness in applications of ambient assisted living for cognitive impaired people," Mobile Networks and Applications, vol. 18, no. 3, pp. 444–453, 2013.

[3] J. Holsopple, M. Sudit, M. Nusinov, D. F. Liu, H. Du, and S. J. Yang, "Enhancing situation awareness via automated situation assessment," IEEE Communications Magazine, vol. 48, no. 3, pp. 146–152, 2010.

[4] M. M. Kokar and M. R. Endsley, "Situation awareness and cognitive modeling," IEEE Intelligent Systems, vol. 27, no. 3, pp. 91–96, 2012.

[5] H. Parvar, M. N. Fesharaki, and B. Moship, "Shared situation awareness system architecture for network centric environment decision making," in Proceedings of the 2nd International Conference on Computer and Network Technology (ICCNT ’10), pp. 372–376, Bangkok, Thailand, April 2010.

[6] S. Dietzel, E. Schoch, B. König, M. Weber, and F. Kargl, "Resilient secure aggregation for vehicular networks," IEEE Network, vol. 24, no. 1, pp. 26–31, 2010.

[7] A. Das and M. M. Islam, "SecuredTrust: a dynamic trust computation model for secured communication in multiagent systems," IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 2, pp. 261–274, 2012.

[8] F. Bao, I. Chen, M. Chang, and J. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," IEEE Transactions on Network and Service Management, vol. 9, no. 2, pp. 169–183, 2012.

[9] W. David, "Resilient aggregation in sensor networks," in Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN ’04), pp. 78–87, Washington, DC, USA, October 2004.

[10] J. Bohli, P. Papadimitratos, D. Verardi, and D. Westhoff, "Resilient data aggregation for unattended WSNs," in Proceedings of the 6th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp), pp. 994–1002, Bonn, Germany, October 2011.

[11] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," ACM Transactions on Sensor Networks, vol. 4, article 15, no. 3, 2008.

[12] P. Stavros, K. Aggelos, and P. Dimitris, "Exact in-network aggregation with integrity and confidentiality," Computational Information Systems, vol. 24, no. 10, pp. 1760–1773, 2012.

[13] C. R. Perez-Toro, R. K. Panta, and S. Bagchi, “RDAS: reputation-based resilient data aggregation in sensor network,” in Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON ’10), Massachusetts, Boston, Mass, USA, June 2010.

[14] X. W. Liu, J. G. Yu, and M. L. Wang, "Network security situation generation and evaluation based on heterogeneous sensor fusion," in Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM ’09), pp. 1–4, Beijing, China, September 2009.

[15] T. Clouqueur, K. K. Saluja, and P. Ramanathan, “Fault tolerance in collaborative sensor networks for target detection,” IEEE Transactions on Computers, vol. 53, no. 3, pp. 320–333, 2004.

[16] E. Elbaşi and S. Özdemir, "Secure data aggregation in wireless multimedia sensor networks via watermarking," in Proceeding of the 6th International Conference on Application of Information and Communication Technologies (AITC ’12), pp. 1–6, Tbilisi, Georgia, October 2012.

[17] K. M. Alam, J. Kamruzzaman, G. Karmakar, and M. Murshed, “Priority sensitive event detection in hybrid wireless sensor networks,” in Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN ’12), pp. 1–7, Munich, Germany, August 2012.

[18] R. A. Shaikh, H. Jameel, B. J. d’Auriol, H. Lee, S. Lee, and Y. Song, “Group-based trust management scheme for clustered wireless sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 11, pp. 1698–1712, 2009.

[19] A. Srinath, Resilient Data Aggregation in Wireless Sensor Networks, Department of Computer Science, North Carolina State University, Raleigh, NC, USA, 2005.

[20] T. Yajima, H. Ushikubo, and H. Shigeno, “Reputation aggregation method considering reliability of newly joining peers in unstructured p2p networks,” in Proceedings of the 7th International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA ’12), pp. 74–81, Victoria, Canada, November 2012.