Future Generation Computer Systems 109 (2020) 573–582

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A novel trust mechanism based on Fog Computing in Sensor–Cloud System

Tian Wang [a,b], Guangxue Zhang [a], MD Zakirul Alam Bhuiyan [c], Anfeng Liu [d], Weijia Jia [e], Mande Xie [f,*]

[a] College of Computer Science and Technology, Huaqiao University, Xiamen, Fujian, China
[b] School of Computer Science and Educational Software, Guangzhou University, Guangzhou, Guangdong, China
[c] Department of Computer and Information Sciences, Fordham University, NY, USA
[d] School of Information Science and Engineering, Central South University, Changsha, Hunan, China
[e] Data Science Center, University of Macau, Macau, China
[f] School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, Zhejiang, China

ARTICLE INFO

Article history:
Received 16 October 2017
Received in revised form 11 April 2018
Accepted 21 May 2018
Available online 26 June 2018

Keywords:
Sensor–Cloud
Fog Computing
Trust mechanism
Misjudgment node
Edge node

ABSTRACT

In recent years, Sensor–Cloud System (SCS) has become a hot research issue. In this system, there are some cyber security problems that can be well solved by the trust mechanism. However, there are still some deficiencies in existing trust mechanisms, especially for the SCS underlying structure. We proposed a fog-based hierarchical trust mechanism for these cyber security deficiencies. This hierarchical mechanism consists of two parts, trust in the underlying structure and trust between cloud service providers (CSPs) and sensor service providers (SSPs). For trust in the underlying structure, the behavior monitoring part is established and implemented in Wireless Sensor Networks (WSNs), and the fine-grained and complicated data analysis part is moved to the fog layer. For trust between CSPs and SSPs, it focuses more on the real-time comparison of service parameters, the gathering of exception information in WSNs, the targeted quantitative evaluation of entities and so on. The experimental results indicate that this fog-based hierarchical structure performs well in saving network energy, detecting malicious nodes rapidly and recovering misjudgment nodes in an acceptable delay. Furthermore, the reliability of edge nodes is well guaranteed by data analyses in the fog layer and an evaluation strategy based on similar service records is put forward.

1. Introduction

SCS is an effective combination of Wireless Sensor Networks (WSNs) and Cloud Computing, which bonds physical sensor nodes, applications and users together [1]. SCS solves some deficiencies which are in traditional wireless sensor networks, like the sensor-node-sharing problem that applications monopolize physical sensor nodes, the data-analyzing problem that WSNs cannot cope well with large amounts of data analysis tasks due to limitations in term of memory, energy, computing power and so on, the user-away-from problem that users cannot get heterogeneous network services as sensing-as-service, etc. [2,3]. Some researchers suggest that Cloud Computing can solve WSNs' deficiencies based on its advantages in terms of technology and economics, so the concept of SCS generates and becomes a hot issue [4]. It virtualizes physical sensor nodes to Cloud so that end users can obtain required sensor data by using virtual sensor services [5]. SCS not only provides users with customized services, but also improves the utilization of sensor nodes by way of making various services no longer occupy physical sensors alone.

However, there are some cyber security problems in SCS that need to be solved, like privacy, internal attacks and external attacks [6,7]. The traditional security mechanism, like encryption, authorization, authentication, does well in curbing external attacks [8,9]. However, it is lack of effective resistance on internal attacks due to malicious attackers entering the network by legal identities [10,11]. In SCS, some internal attacks become more serious, which occur in the wireless sensor networks layer and the junction between the wireless sensor networks layer and the cloud layer, for the immaturity technology of SCS and WSNs' limitations in computing power, storage capacity and energy [12–14].

A well designed trust mechanism can find malicious entities and eliminate security risks by monitoring behaviors of internal attack entities [15,16]. Moreover, the trust mechanism is more suitable for WSNs than the traditional security mechanism due to its light weight in cyber security [17,18]. However, the research

* Corresponding author.
E-mail addresses: mbhuiyan3@fordham.edu (M.Z.A. Bhuiyan), afengliu@mail.csu.edu.cn (A. Liu), xiemd@zjgsu.edu.cn (M. Xie).

https://doi.org/10.1016/j.future.2018.05.049
0167-739X/

on the trust mechanism of SCS is not perfect, especially in the Sensor–Cloud underlying structure. There are many aspects that should be concerned, such as energy consumption of establishing trust mechanism in WSNs, how to find hidden data attack, how to ensure outer nodes credible and how to recover misjudgment sensor nodes. In addition, there are two problems that need to be solved in the establishment of trusted third-part. (1) How to establish the trust relationship that is SSPs to CSPs. CSPs have some characteristics, such as the service similarity, the service dynamics, the changing service quality and so on [19,20]. (2) How to establish the trust relationship that is CSPs to SSPs. SSPs may provide tampered data, do not guarantee real-time data, submit low quality data and so on. The quality of service (QoS) of SCS is ensured by the quality of data that are from WSNs. To solve these problems, we adopt Fog Computing, which is closer to WSNs and can be designed as a trusted third party, for Fog Computing can extend Cloud Computing to the network edge [21]. In this paper, we proposed and designed a fog-based hierarchical trust mechanism for SCS in detail. In summary, we add the fog layer in Sensor–Cloud for following three objectives:

(1) The fog layer can reduce energy consumption, detect malicious nodes rapidly and recover misjudgment nodes in an acceptable delay.

(2) The fog layer can secure the whole WSNs from a global perspective and realize more functions, such as ensuring edge nodes credible and finding hidden data attacks.

(3) The fog layer can assist CSPs and SSPs in establishing trust relationship in a more comprehensive and credible means.

The rest of this paper is organized as follows. Section 2 introduces SCS structure, Fog Computing and development of trust mechanisms in SCS. Section 3 gives network model and design framework. Section 4 describes detailed design of trust mechanism. Section 5 provides analyses of experimental results. The last Section concludes this paper.

2. Related work

2.1. The general SCS structure

The general SCS structure consists of the user layer, the cloud layer and the wireless sensor networks layer as shown as Fig. 1. For the wireless sensor networks layer, researchers focus on the management strategy that aims at how to efficiently and optimally improve the utilization of physical sensor nodes, that is to say, a physical sensor node should service multiple sets of applications; for the cloud layer, the main work is establishing virtual sensor groups corresponding to physical sensor nodes and allocating virtual sensor nodes to users according to their requirement, in other words, users will not need to pay attention to the design and organization document of WSNs, which achieves great implementation efficiency and operation flexibility; for the user layer, the focus is on the design of application software and user interfaces that help users accurately publish requirements and find available resources. SCS is applied in environmental monitoring, disease surveillance, wildlife monitoring, remote sensing, agriculture and irrigation control, and so on [22,23].

2.2. The trust mechanism in SCS

For security breaches, they can exists in every part of SCS, such as data generation, data transmission, in-network processing and the cloud [24]. In WSNs, captured nodes can lead to fake sensor data through producing fake sensor data in data generation and providing some fake sensor data in in-network processing. Moreover, captured nodes can also steal data through many routing attacks, such as Sybil attack, wormhole attack and so on. In cloud,

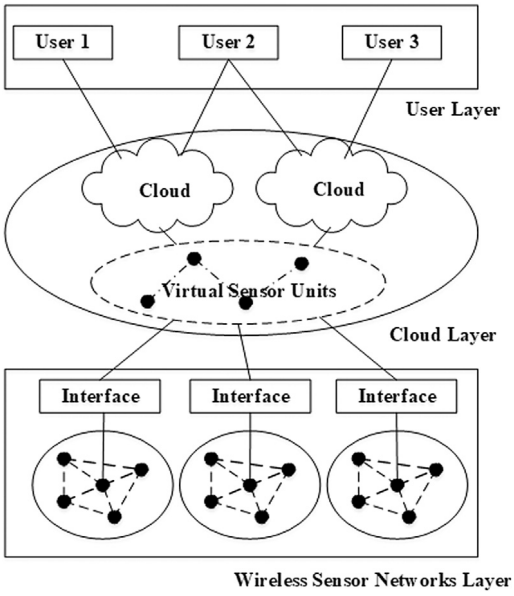


Fig. 1. The general SCS structure.

the data owner loses the direct control of sensor data, and the attacker can breaches data integrity and availability by malicious virtual machines (VMs). Trust management and trust dynamics are a decision aid complementary to cryptography, authentication, and stenography.

In SCS, there are two very critical and barely explored issues. The one is fake services among Cloud Service Users (CSUs), Cloud Service Providers (CSPs) and Sensor Network Providers (SNPs), which occurs in CSPs–CSUs and SNPs–CSPs. Another is that CSPs and CSUs likely select low trust and reputation service providers without the help of a trust and reputation calculation and management system. To solve these problems, Zhu et al. [25] proposed an authenticated trust and reputation calculation and management (ATRCM) system. ATRCM can avoid malicious impersonation attacks, moreover, it can help CSUs choose desirable CSPs and assist CSPs in selecting appropriate SNPs. To improve the quality of service (QoS) of Sensor–Cloud, Zhu et al. [26] proposed Trust-Assisted Sensor–Cloud (TASC), where trusted sensors and trusted data centers are selected to assist data transmission from sensors to users. However, there should be more service parameters that should be considered during the establishment of trust and reputation among CPUs, CSPs and SNPs, such as real-time, similarity and integrity.

To protect sensor data in the cloud, Henze et al. [27] proposed a trust point-based security architecture, which is a security-enhanced gateway against unauthorization access. In this trust model, there are three parts, which are the producer domain (sensor nodes and gateway devices, data owners), the storage domain (cloud, cloud providers) and the consumer domain (services, service providers). The trust point is an enhanced gateway device, which locates between data owner and cloud and guarantees the security of data transmission. However, it cannot guarantee that sensor data and cloud providers are credible.

In WSNs, trust models are divided into the node trust model and the data trust model [15,28,29]. In the node trust model, there are two types, the centralized trust model and the distributed trust model. For the centralized trust model, a base station or cluster head do trust value calculations for sensor nodes. It takes the base station or cluster head more burden in communication, computing, storage and so on. For the distributed trust model, sensor nodes do

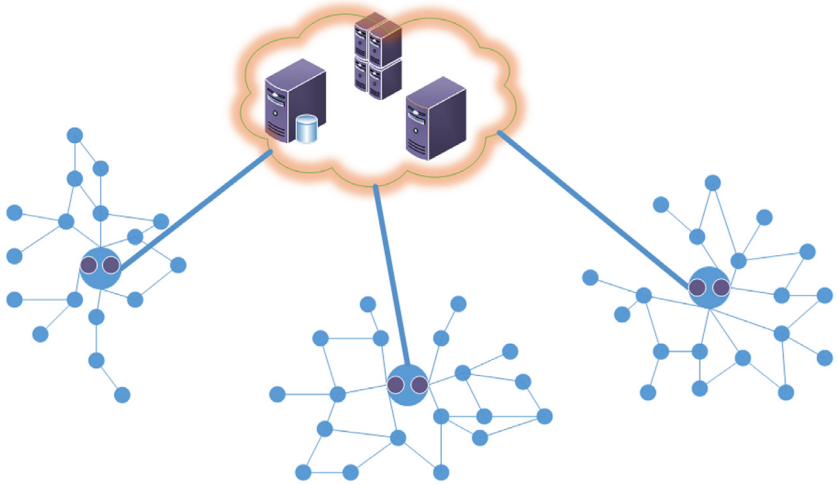


Fig. 2. The SCS underlying structure with a ladder diffusion algorithm.

trust value calculations by themselves. The more trust evidences are collected or the more complicated the theory for trust calculation is, the more resources of network are consumed, which is bad to resource-constrained WSNs. In the data trust model, the key is how to distinguish modified data and original data. Many researchers pay more attention in WSNs, such as sensor node identity, data similarity and data forwarding delay. However, it will consume more network resources when the data trust is established in WSNs.

There are many types of trust models in Cloud, such as feedback based, agent based, log based, authentication based and SLA (Service-Level Agreement) based [30]. For SCS, its three layers can be abstracted as three entities that are CSUs, CSPs and SSPs. Most researchers focus on how to establish trust on CSPs, but there are few studies on trust relationships of CSPs to SSPs or CSPs to CSUs. Moreover, the trust value has timeliness and locality. So, when updating trust values, there are more factors that should be considered, such as service dynamics, user cultural differences, demand diversification and evaluation criteria deviation. To some extent, the scalable evaluation standard and the detailed assessment criteria are two important directions of SCS trust mechanisms in the future.

Moreover, these systems are lack of concrete methods in detecting hidden data attacks, reducing energy consumption in WSNs, recovering misjudgment nodes, ensuring edge nodes credible and establishing a trusted and real-time third party among three entities of SCS.

2.3. Fog Computing

Fog Computing is proposed firstly by Cisco, which aims to extend Cloud Computing to the edge of network [31]. It is applied to many fields, such as vehicle networking, smart grid, smart city and WSNs. The definition of the fog [32]: *"fog computing is a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third parties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so"*. Fog Computing is located between Cloud Computing and edge networks, with low latency, location awareness, mobility, real-time, supporting heterogeneous devices and so on.

3. Preliminary

3.1. WSNs model

WSNs can be composed of multiple cluster structures with the efficient data collection and node management capability. The cluster structure is expressed as managing many adjacent sensor nodes that are in same geographic area or implementing monitoring functions by the mutual cooperation among many adjacent sensor nodes. A cluster structure consists of more than one cluster heads and many intra cluster nodes that cooperate with others to complete some tasks. Many cluster structures are combined together to realize complex network function with lower energy consumption. In the design, we adopt a multi-cluster-heads structure which is not only convenient to manage intra cluster nodes but also realizes the mutual supervision among cluster heads. Ho et al. in [33] proposed a ladder diffusion algorithm to reduce routing loops and extend network lifetime. Based on this routing algorithm, we do a change that nodes can transfer data to peer layer and upper layer. The sketch map of SCS underlying structure is shown as Fig. 2.

3.2. A novel framework for trust mechanism in SCS

We propose a new framework to solve some problems in trust mechanisms of SCS as shown as Fig. 3.

In this framework, the fog layer has three main functions. (1) The fog layer acts as a trust buffer zone between the cloud layer and the wireless sensor networks layer, which mainly focuses on three parts, collecting network state evidences, recording service details and monitoring service parameters. (2) The fog layer executes some tasks, such as data pre-processing, temporary storage, small-scale calculation and providing Cloud with services. (3) The fog layer has virtualization capability and node task allocation mechanisms, which transfer virtualization from the cloud layer to the fog layer. For the establishment of trust mechanism in SCS, the fog layer does well in three aspects. The first one is that the fog layer gets the whole trust state of WSNs. The second one is that the fog layer can deal with some data analysis tasks to find hidden data attacks, ensure edge nodes credible and recover misjudgment nodes. The third one is that the fog layer assists in establishing trust relationships between CSPs and SSPs through evidences collected and service records. There are three major database centers in the fog layer, which are the sensor database, the event control database and the service providing database. The sensor database mainly

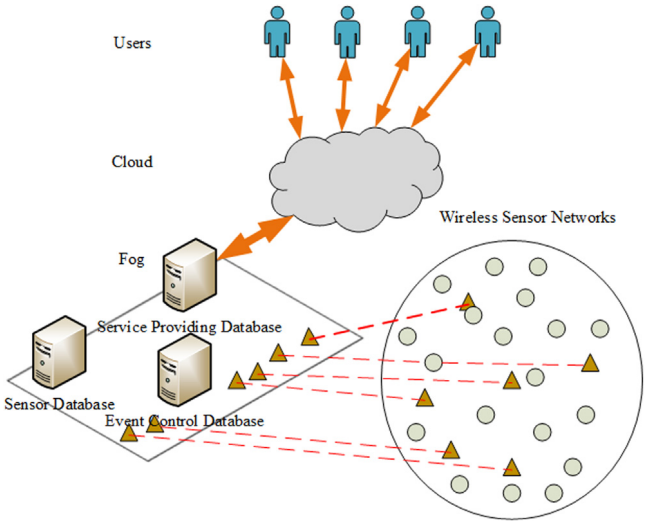


Fig. 3. A new trust mechanism framework with Fog Computing.

stores impermanent real-time data from WSNs, which can be used for some data analysis tasks; the event control database mainly does some judgments about exceptions uploaded from WSNs; the service providing database mainly provides CSPs with data services and does some monitoring tasks on service parameters.

3.3. Analysis of the novel trust mechanism

Fog Computing is the extension of Cloud Computing, so some security mechanisms that are in Cloud Computing can be used to ensure the security of Fog Computing, such as access control, audit, authentication and some complex trust mechanisms. In this paper, we mainly focus on two parts of this trust mechanism in SCS, which are among entities and in the underlying structure.

For the establishment of trust among three entities in the fog layer, there is a simple example. Before a service transaction, SSP_a and CSP_a negotiate service contents and store standard service parameters in the fog layer. In the course of service transaction, the fog layer monitors both SSP_a and CSP_a. On the one hand, the fog layer monitors SSP_a's service parameters and records exceptions happening in WSNs. The trust state of SSP_a is partially updated based on these monitoring values in real-time. On the other hand, the fog layer also partially updates trust state of CSP_a based on comparing real-time service parameters and standard service parameters. After the service transaction finishes, the detailed service record is stored in the fog layer. In other service transactions, this record will be used to calculate trust values of CSP_a and SSP_a.

For the establishment of trust in the underlying structure, the data processing capability of sensor nodes is limited, so the trust mechanism is designed into three levels with the advantage of Fog Computing. The first level is the gathering of observation values among physical sensor nodes and the calculation of direct trust, which just requires less computing workload; the second level reacts to some abnormal events, such as frequent routing failure, longer data forwarding time, larger difference between new and old trust value; the third level does data processing, like calculating global trust state of WSNs, discovering hidden data attacks, recovering misjudgment nodes and so on.

4. Design of trust mechanism

4.1. Direct trust among nodes

There are many behavior characteristics which can be observed to evaluate the trust state of nodes during node interaction process. However, the more characteristics are collected, the more difficult the system implementation becomes due to some restrictions should be followed, such as energy consumption, network load and others. We choose the packet loss rate, the route failure rate and the forwarding delay as evidences to assess the trust state of node.

The packet loss rate, $Trust_{packet}$, refers to the ratio that the number of data packets lost by recipient occupies the proportion of the total data packets in a communication cycle, which represents a type of evidence that can indicates node state or node whether is breached; The route failure rate refers to the ratio that the number of routing packets discarded by recipient occupies the proportion of the total routing packets sent by sender during an interval of time, which can be used to judge network state; The forwarding delay, $Delay_{forwarding}$, refers to the time interval from receiving data to forwarding data when relay node transfers data, which represents a type of evidence that node has been compromised or has serious fault. The source node can use these evidences to establish a direct trust relationship on cooperative nodes. Moreover, the observation value of node behavior may fluctuates with the change of environment and network load, so the history trust value, $Trust_{history}$, is added into direct trust calculation to reduce the misjudgment rate of normal nodes and the unnecessary waste of network resources. The formula of direct trust is shown as (1).

$$Trust_{direct} = (w_1 Trust_{packet} + w_2 Trust_{history}) \times Delay_{forwarding} \quad (1)$$

Where $Delay_{forwarding}$ is a very important evidence of serious security problem, which is considered that a relay node has modified data. When the time interval is greater than threshold, the value of $Delay_{forwarding}$ is set 0, otherwise 1. If $Delay_{forwarding}$ exception occurs, the value of $Trust_{direct}$ is 0. Otherwise, the value of $Trust_{direct}$ is determined by $Trust_{packet}$ and $Trust_{history}$ based on the weighted algorithm.

For weighted values, $w_1 + w_2 = 1$. To reduce node energy consumption in the data transmission, the trust detection period among nodes is maximized in a reasonable range. In this case, the trust value may become too old to really reflect the present trust state of node. So we cut down the weight of $Trust_{history}$ by formula (2).

$$w_2 = real_1 \times Period_{network} \times \exp(-real_2 \times Period_{network}) \quad (2)$$

Where $Period_{network}$ is the time interval from the last update to now. $real_1$ and $real_2$ are two real numbers that are set at initialization.

Theorem 1. *The larger the value of $Period_{network}$ is, the faster it converges to $Trust_{packet}$.*

Proof. Firstly, we take derivative of the function to check its changing trend. Then, we find out the descent region of this function. Finally, an appropriate descent region is selected for the weight setting of $Period_{network}$.

$$\begin{aligned}(w_2)' &= (real_1 \times Period_{network} \times \exp(-real_2 \times Period_{network}))' \\ &= real_1 \times (1 - real_2 \times Period_{network}) \\ &\quad \times \exp(-real_2 \times Period_{network})\end{aligned} \quad (3)$$

$$(w_2)' = 0 \quad then \quad Period_{network} = \frac{1}{real_2} \quad (4)$$

We can find that the curve goes down with $Period_{network}$ larger than $\frac{1}{real2}$. The curve drops dramatically in the front part and declines stably in the latter part. To achieve an ideal result, the value of $real_2$ can be set in [0.7,1] and the value of $real_1$ is set according to $real_2$, which can assign different weighted values to $trust_{history}$ according to different $Period_{network}$. □

4.2. Comprehensive trust among nodes

In this level, the source node requests recommendation values from its trust adjacent nodes when it finds some exceptions of adjacent nodes. Meanwhile, the source node also sends these exceptions to the fog layer to analyze the trust state of every node in this area. If these abnormal nodes are identified as malicious, the fog layer would inform cluster head to isolate these malicious nodes.

Exceptions in WSNs are divided into three categories, which are the route failure rate exception, the forwarding delay exception and the difference value exception. The routing failure is a normal phenomenon in WSNs, but it is considered as an exception when the route failure rate is up to threshold in a certain period of time. When the forwarding delay surpasses threshold, a forwarding delay exception occurs. The difference value exception is that the difference value between new trust value and historical trust value is beyond the reasonable range. The general formula of recommendation trust calculation is shown as (5).

$$Trust_{recommendation} = \sum_{i \in set(neighbor)} w_{i(i,j)} \times Trust_{(j,k)} \tag{5}$$

Where $set(neighbor)$ is a trust node set of the source node. $Trust_{(j,k)}$ is the trust value of node j to node k. However, the source node has different trust values to different adjacent nodes. In this case, there should be some mechanisms to properly decrease the impact of low performance nodes. Here, we sort trust table of the source node from small to large by trust values, then calculate the weighted value of every adjacent node by an arithmetic progression, as formula (6).

$$w_{i(i,j)} = \frac{i}{\sum_1^n i} = 2 \times \frac{i}{n(n+1)} \tag{6}$$

Where parameter i is the location value of nodes in the ordered trust table and parameter n is the node number in $set(neighbor)$.

To be specific, $Trust_{recommendation}$ gives the source node an advisory opinion, and the final decision of the source node is based on $Trust_{direct}$ and $Trust_{recommendation}$ as shown as (7). The weighted value of $Trust_{direct}$ is larger than $Trust_{recommendation}$, and $w_3 + w_4 = 1$.

$$Trust_{synthesis} = w_3 \times Trust_{direct} + w_4 \times Trust_{recommendation} \tag{7}$$

Theorem 2. *A small proportion of malicious recommendation nodes cannot decide* $Trust_{recommendation}$.

Proof. The most trusted node of the source node has the greatest weighted value, which is $\frac{2}{n+1}$. The difference value between weighted values of two adjacent nodes in trust table is $\frac{2}{n(n+1)}$. When n is in $[2, 3, \ldots, n]$, the corresponding greatest weighted value is $\left[\frac{2}{2+1}, \frac{2}{3+1}, \ldots, \frac{2}{n+1}\right]$. The larger n is, the smaller the weighted value of every node has. □

4.3. Data analysis in the fog layer

There are three types of data analysis in the fog layer. The first type recovers misjudgment nodes and detects hidden data attacks, which is based on trust tables, historical sensor data and network topology. The second type inspects whether there are some malicious nodes or malicious recommendation nodes after receiving exceptions from WSNs, which is based on trust tables, recommendation tables, historical sensor data and network topology. The third type is concerned about the credibility of edge nodes, which is based on trust tables and sensor data correlation.

All sensor nodes send change values of trust table along with sensor data to the fog layer during a certain period, and the source node sends the recommendation table along with sensor data to the fog layer after finishing the recommendation trust calculation. The fog layer periodically analyzes the global trust state of every node and verdicts whether there are misjudgment nodes and hidden data attacks. Moreover, based on these global trust state of nodes, we can predict some network status, like network load, residual energy of nodes, and so on.

Some malicious nodes provide wrong sensor data to lead users make wrong decisions. These nodes are more difficult to find, for they behave normally when they communicate with other nodes. Within the same area or the same cluster, there are some data correlation phenomena, such as sensor data from several nodes in the same geographical position are similar, sensor data from several nodes in different geographical positions show gradualness, sensor data from several nodes moving together have trajectory correlation. The fog layer can process simultaneously sensor data from several nodes and analyze whether there are malicious nodes through some data correlation phenomenon indicators, such as variation trend, fault-tolerant interval, similar trajectory, and so on. The fog layer is closer to WSNs, so the delay of detecting hidden data attacks is acceptable. We carry on a multi-path operation to analyze sensor data from different nodes. The process structure is shown as Fig. 4.

Here, we mainly consider nodes that realize the same function in the same geographical position. The formula is show as (8).

$$Array = \begin{cases} Count_{crest} \cup degree, \dfrac{X2_i - X1_i}{Y2_i - Y1_i} > 0 \\ Count_{trough} \cup degree, \dfrac{X2_i - X1_i}{Y2_i - Y1_i} < 0 \end{cases} \tag{8}$$

Where we use *Array* to store the crest, the trough, and the degree. $Count_{crest}$ indicates the crest of sensor data curve, which is recorded as 1. $Count_{trough}$ indicates the trough of sensor data curve, which is recorded as -1. *degree* is the difference value between two adjacent sensor data. The judgment of crest/trough is continuous two negative/positive values after a positive/negative value, and the continuous crest/trough occurs when the change value of sensor data is zero after a crest/trough record. At every time point, *Array* records state value $(1, -1, 0)$ and degree value in *Array*.

For edge nodes, they have less communications with other nodes comparing to inner nodes. We set shorter period detect for edge nodes in WSNs. Moreover, the fog layer will scan and analyze trust state of edge nodes in a shorter period.

4.4. Establishment of trust relationship between SSPs and CSPs

The trust relationship among CSPs and SSPs is divided into two parts. One is the trust relationship of CSPs to SSPs, and the other is the trust relationship of SSPs to CSPs.

4.4.1. Trust relationship of CSPs to SSPs

For CSPs, they expect that data from SSPs should meet some requirements, such as no tampering, integrality, timeliness and precision. However, a service user may does not require service providers to meet all needs, that is to say, service providers only need to meet the specific requirements of the service user. So, there should be some special recommendation mechanisms to find CSPs that offer good services in special aspects. Fog Computing can deal well with these problems. The trusted third party based on Fog Computing can ensure SSPs reliability through three parts as shown as formula (9).

$$Trust_{SSPs} = w_5 Trust_{service} + w_6 Trust_{WSNs} + w_7 Trust_{CSPs} \tag{9}$$

Where $Trust_{service}$ is the trust value for service parameters. Before a service transaction, SSPs and CSPs negotiate service parameter

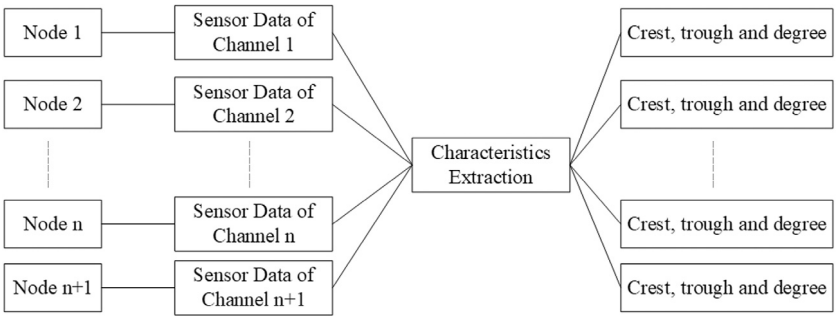


Fig. 4. The structure of data analysis.

standards. Then, the fog layer monitors these service parameters during a transaction in real-time, and contrasts these service parameters with standard values. If the value of a monitored service parameter is in the reasonable range, the record of this parameter is 1, else 0. Finally, $Trust_{service}$ is calculated through different weighted values of service parameters. $Trust_{WSNs}$ is the trust value for WSNs, and it is based on exception information records of WSNs. If a WSN has more exceptions in a transaction, it would be given a lower value. $Trust_{CSPs}$ is a type of trust value that is calculated through the record information of other CSPs in the fog layer. There are two steps that are used to decide the selection of CSPs, the general recommendation calculation $R_{general}$ and the similar recommendation calculation $R_{similar}$. Some CSPs are selected into a candidate list, whose service records contain all requested service parameters. $R_{general}$ assigns all selected CSPs to different sets which are classified by the number of redundant parameters. Then, the trust value of every SSP is separately calculated in different sets. Finally, some abnormal CSPs are excluded from candidate list according the change rule of trust value among different sets. $R_{similar}$ is an optimal selection strategy, which has many principles that can be selected by service users. $Trust_{CSPs}$ is calculated with these selected CSPs. w_5, w_6 and w_7 are three weighted values and set at initialization depending on users' different requirements, and $w_5 + w_6 + w_7 = 1$. The value of $Trust_{service}$, $Trust_{WSNs}$ and $Trust_{CSPs}$ is in range 0 to 1.

4.4.2. Trust relationship of SSPs to CSPs

For SSPs, they expect services that are provided by CSPs should meet some criteria, such as reliability, security, friendliness, controllability, stability and so on. These indicators are important for SSPs to establish trust on CSPs. The fog layer can monitor these indicators in real-time. The trust relationship of SSPs to CSPs consists of two parts as shown as formula (10).

$$Trust_{SSPs} = w_8 Trust_{service1} + w_9 Trust_{SSPs} \tag{10}$$

Where $Trust_{service1}$ is the trust value of CSPs' service parameters that is similar to $Trust_{service}$. $Trust_{SSPs}$ is calculated with some selected SSPs, whose selection process is similar to $Trust_{CSPs}$. In the fog layer, there are some databases that are used to store service records of a certain time. w_8 and w_9 are two weight values and set at initialization depending on users' different requirements, and $w_8 + w_9 = 1$. The value of $Trust_{service1}$ and $Trust_{SSPs}$ is in range 0 to 1.

5. Evaluation

The experiment platform is MATLAB R2016b. There are six cluster structures with more than three hundred nodes which are randomly deployed in the wireless sensor networks layer. Every cluster is divided into three levels, where the outer layer has more nodes than the inner layer. In every cluster structure, cluster heads can receive sensor data packets from six nodes simultaneously. The maximum delay time from WSNs to the fog layer is set as seven communication cycle. These parameters are shown in Table 1. In the fog layer, there are some record that are close to real records.

Table 1
Simulation parameters.

Parameters	Values
Network protocol	The Ladder Diffusion Algorithm
The number of Clusters	6
The number of Cluster Heads	36
The number of Cluster Nodes	300
The number of levels	3
The maximum delay	7

5.1. The trust update status of nodes

There are two types of trust mechanisms, which are the periodic update and the aperiodic update. For the aperiodic update, nodes update trust states of their adjacent nodes when detecting abnormal behaviors. There are some flaws in the aperiodic update, such as too little attention to edge nodes and older trust states of nodes, the experiment result is shown as Fig. 5(a). The aperiodic update cannot detect malicious nodes in time. For the periodic update, nodes update trust values of their neighbor nodes when the period time is over. There are also some disadvantages in the periodic update, like taking up too much storage and computing resources, lowering network communication performance and so on, whose experiment result is shown in Fig. 5(b). Our design is based on the periodic update. We set that the trust updating cycle at outermost layer is the same to the periodic update, which can be found obviously in Fig. 5(b) and 5(c). We can lengthen the trust updating cycle at inner layer with the help of Fog Computing, which can avoid more waste of resources in the periodic detection as shown as Fig. 5(c). Three experimental results above consider the number of trust updating times in every level of WSN, and the network load increases based on augmenting the number of data-generation nodes.

Fig. 5(d) shows a total number of trust updating times in a shorter test time. From these experimental results, we can get three pieces of information. (1) For the aperiodic update, the number of trust update times gradually increases with more random nodes selected to transmit data. (2) For the periodic update, it maintains a steady state. However, there is a slight reduction from 6 to 24 on the x-axis, the reason is that the direct trust update reduces the number of periodic update times. (3) For our design, we can get more biggish advantage when the updating cycle is lengthened. When the WSN gets congested, the network transmission capacity decreases and the number of trust updating times increases due to frequent routing failures. Compared with the periodic update, our design can save network energy and maintain network performance by reducing the number of periodic update times.

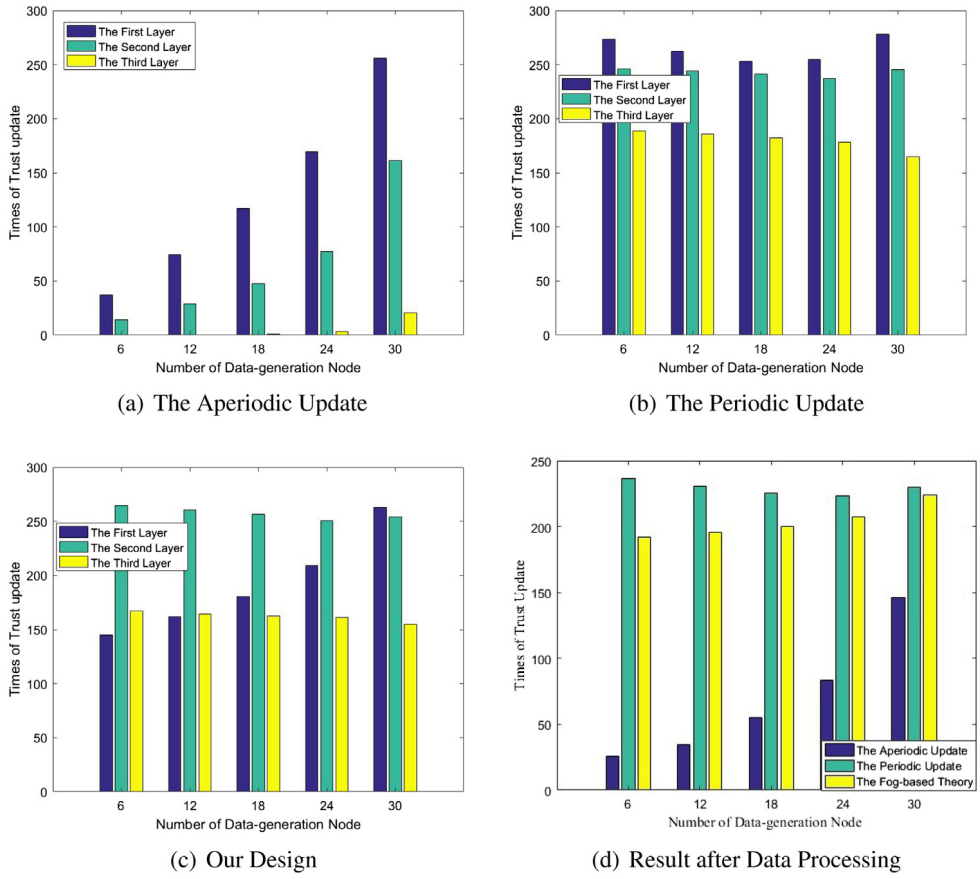


Fig. 5. The comparison of three schemes.

5.2. The detect speed of malicious nodes

There is no need for too frequent trust updating, for internal attacks occurs in a special time and frequent updating will occupy more transmission and computing resources. We compare the detect speed of malicious nodes between our design and the periodic update.

Malicious nodes can be detected from two parts, the wireless sensor network layer and the fog layer. Because the delay time in the fog layer is slightly longer than that in the wireless network layer, the malicious nodes detection in the fog layer is introduced as an aided detection. The detect speed of the periodic update is shown in Fig. 6(a), and Fig. 6(b) shows the detect speed of our design. In the experiment, we separately placed malicious nodes in different levels of network when initializing, which spend more time than in the process of mechanism running. Experiment results show that the detect speed of malicious nodes gets faster with the increasing of network load except for the outermost layer, because the trust state of nodes is updated more frequently when the network load gets larger. In Fig. 6(c), we randomly place malicious nodes in three levels that indicates a more intuitive downward trend. Even though there are some delay problems in detect speed, we can take advantage of Fog Computing to get the whole trust state of WSNs through some data analyses, such as the hidden data attack detection and the anomaly detection of monitor values.

5.3. The recovery of misjudgment nodes

The service capabilities of nodes may fluctuate with the change of environment and themself electricity consumption. So some nodes show unusual behaviors in some cases, such as the biggest change of environment, low battery and transient fault. Almost all trust mechanisms do not consider these cases, we solve these problems in the fog layer by data analysis.

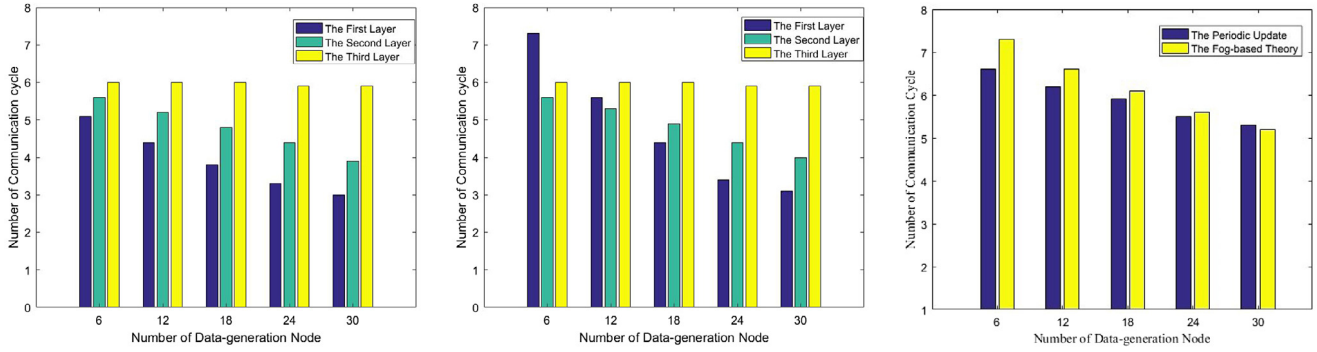
This experiment scene is that the number of data-generation nodes is 18 in one communication cycle, and we add one kind of condition every five communication cycles after the network is initialized. As shown as Fig. 7, there are some malicious nodes added to network when initializing at (1). After six communication cycles, the periodic update and our design find malicious nodes at the same time. We add general malicious nodes at (2). After (2), malicious nodes are detected by both trust mechanisms after a fewer of communication cycles. The cleanup program of malicious nodes is executed at (3) and the environment is changed at (4). After the change of environment, nodes that are sensitive to environment show abnormal behaviors which can be misjudged as malicious nodes. During this period, there are no malicious nodes actually. For these misjudgment nodes, the fog layer analyzes whether these nodes are true malicious nodes by trust tables, recommendation trust tables, network topology and long-term sensor data. After an acceptable delay, our scheme can recover misjudgment nodes before the cleanup program at (5).

5.4. The selection for SSPs

For $Trust_{service}$, it can be get by comparing real-time monitor values and standard negotiation values. For $Trust_{WSNs}$, it can be calculated based on some exception records that is discovered and recorded by the fog layer. For the calculation of $Trust_{SSPs}$, we will introduce it with a concrete example, such as one CSP wants to select SSPs that can provide with integrality and precision. Before the calculation of $Trust_{SSPs}$, there is some scoring mechanisms, such

Table 2
The information record of SSPs in the fog layer.

	Service requirement	Interaction record	Recommendation record	Number of accepted	Check-in time
CSP1	Integrality precision	SSP1 (60) SSP2 (91) SSP4 (98)	CSP2 CSP3 CSP4	56/58	10
CSP2	Integrality precision	SSP2 (90) SSP4 (95) SSP5 (92)	CSP1 CSP3 CSP5	46/60	8
CSP3	No tampering integrality precision	SSP3 (72) SSP4 (93) SSP5 (82)	CSP1 CSP2 CSP4 CSP5	88/92	15
CSP4	No tampering integrality precision	SSP1 (84) SSP3 (70)	CSP1 CSP2 CSP5	26/29	4
CSP5	No tampering integrality timeliness precision	SSP1 (71) SSP2 (70) SSP3 (70) SSP4 (82)	CSP3 CSP4	32/33	1



(a) The Periodic Update (b) Our Design (c) Result after Data Processing

Fig. 6. The comparison of two schemes.

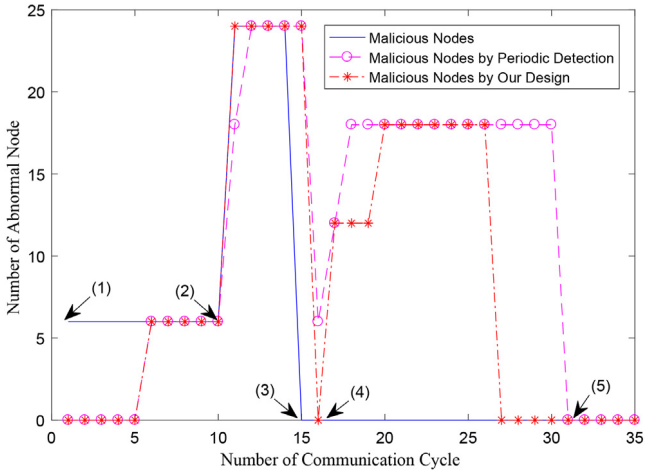


Fig. 7. The recovery of misjudgment nodes.

as score drops one grade when service parameters are one less than requirement service parameters. In Table 2, we only display some important parameters that is shown as an example. The service requirement stores service parameter records of CSP who has required these service parameters in a past transaction. The interaction record stores SSPs and their trust values, which have provided data to the CSP. The recommendation record stores CSPs that have received recommendation services from the CSP. The number of accepted stores the number of accepted recommendations and total recommendations. The check-in time stores the existence time of the CSP in the fog layer.

Firstly, the fog layer selects service records that are in the acceptable time range as shown as Table 2, where integrality and timeliness are in different sizes of service requirement sets, such as $set1 : (CSP1, CSP2)$, $set2 : (CSP3, CSP4)$, $set3 : (CSP5)$. There are some potential matchup relationship that is SSPs and their advantages in the bracket. (1) SSP1 (no tampering, integrality). (2) SSP2 (integrality, timeliness). (3) SSP3 (no tampering, timeliness). (4) SSP4 (no tampering, integrality, precision). (5) SSP5 (integrality, timeliness). Secondly, $R_{general}$ performs. In the calculation of $R_{general}$, every SSPs' average trust value is calculated in different sets, such as [SSP1(60), SSP2(90.5), SSP4(96.5), SSP5(92)] in set1, [SSP1(84), SSP3(71), SSP4(93), SSP5(82)] in set2 and [SSP1(77.5), SSP2(70), SSP3(70), SSP4(82)] in set3. The trust value of one SSP in smaller set should be great than or equal to that in larger set. So, we can find some abnormal evaluation by this rule, such as CSP1 may is a wrong selection. Thirdly, $R_{similar}$ performs. After abnormal recommender are removed, the optimal choice is considered, such as familiarity, popularity and risk. The familiarity is influenced by recommendation records, location of SSPs and reputation. The popularity refers to the accepted number of one CSP's recommendations. The risk focuses on whether it takes some losses if a young recommender is selected. For an example, CSP3 has more recommendation services than CSP2, and its number of accepted recommendations is larger than CSP2. Moreover, CSP3 stays in the fog layer for a more long time that show it more credible than CSP2. Finally, final recommenders are chosen.

5.5. The selection for CSPs

For the selection for CSPs, its information record table is similar to the selection for SSPs, which also contains the service requirement, the interaction record, the recommendation record, the number of accepted and check-in time. For $Trust_{service1}$, it also is generated by the fog layer based on the comparison between monitored service parameters and standard service parameters. For $Trust_{SSPs}$, the process of selection is the similar to that of SSPs.

6. Conclusion

SCS becomes more and more popular among many places, and its security issues have been concerned all the time. The traditional security mechanism cannot effectively deal with internal attacks, especially in the SCS underlying structure. There is also lack of a sufficiently trusted third party to establish the trust

relationship between SSPs and CSPs. We design a fog-based hierarchical trust mechanism to make well up for these deficiencies and solve resource consumption problems in WSNs, whose the behavior trust among nodes is established in the wireless sensor networks layer and the data trust of nodes and entities is established in the fog layer. Through a more granular data analysis in the fog layer, we can monitor the whole network trust state, detect data attacks and recover misjudgment nodes. Moreover, the fog layer can be built as a credible third party. The experiment results show that our trust mechanism has some advantages in some respects, such as reducing energy consumption, ensuring the trust state of edge nodes and network, detecting some hidden data attacks and recovering misjudgment nodes.

Acknowledgments

Above work was supported in part by grants from the National Natural Science Foundation of China (NSFC) under Grant No. 61772148 and No. 61672441 and Natural Science Foundation of Fujian Province of China (No. 2018J01092, No. 2016J013021 and No. 2016J05158) and the Fujian Provincial Outstanding Youth Scientific Research Personnel Training Program, China and Chinese National Research Fund (NSFC) Key Project No. 61532013 and National China 973 Project, China No. 2015CB352401 and Shanghai Scientific Innovation Act, China of STCSM No. 15JC140-2400 and 985 Project of Shanghai Jiao Tong University, China with No. WF220103001 and Subsidized Project for Cultivating Postgraduates' Innovative Ability in Scientific Research of Huaqiao University, China No. 1611314018.

References

[1] A. Alamri, W.S. Ansari, M.M. Hassan, M.S. Hossain, A survey on sensor-cloud: architecture, applications, and approaches, Int. J. Distrib. Sens. Netw. 2013 (6) (2013) 18.
[2] M. Yuriyama, T. Kushida, Sensor-cloud infrastructure - physical sensor management with virtualized sensors on cloud computing, in: International Conference on Network-Based Information Systems, 2010, pp. 1–8.
[3] M.Z.A. Bhuiyan, G. Wang, J. Wu, J. Cao, X. Liu, T. Wang, Dependable structural health monitoring using wireless sensor networks, IEEE Trans. Dependable Secure Comput. 14 (4) (2017) 363–376.
[4] S. Misra, S. Chatterjee, M.S. Obaidat, On theoretical modeling of sensor cloud: a paradigm shift from wireless sensor network, IEEE Syst. J. 11 (2) (2017) 1084–1093.
[5] S. Madria, V. Kumar, R. Dalvi, Sensor cloud: a cloud of virtual sensors, IEEE Softw. 31 (2) (2014) 70–77.
[6] T. Wang, J. Zhou, M. Huang, M. Bhuiyan, A. Liu, W. Xu, M. Xie, Fog-based storage technology to fight with cyber threat, Future Gener. Comput. Syst. 83 (2018) 208–218.
[7] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, Y. Tang, Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks, J. Netw. Comput. Appl. 106 (2018) 117–123.
[8] C.Z. Gao, Q. Cheng, X. Li, S.B. Xia, Cloud-assisted privacy-preserving profile-matching scheme under multiple keys in mobile social network, Cluster Comput. (2018) 1–9.
[9] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, C.Z. Gao, Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures, J. Netw. Comput. Appl. 107 (2018) 113–124.
[10] Z. Sun, L. Li, X. Li, X. Xing, Y. Li, Optimization coverage conserving protocol with authentication in wireless sensor networks, Int. J. Distrib. Sens. Netw. 13 (3) (2017) 155014771769556.
[11] J. Granjal, E. Monteiro, J.S. Silva, Security in the integration of low-power wireless sensor networks with the internet: a survey, Ad Hoc Netw. 24 (2015) 264–287.
[12] T. Wang, M.Z.A. Bhuiyan, G. Wang, M.A. Rahman, J. Wu, J. Cao, Data reduction for a smart citys critical infrastructural health monitoring, IEEE Commun. Mag. 56 (3) (2018) 128–133.
[13] T. Wang, J. Zeng, Y. Lai, Y. Cai, H. Tian, Y. Chen, B. Wang, Data collection from WSNs to the cloud based on mobile Fog elements, Future Gener. Comput. Syst. (2017).
[14] A. Ahmed, K.A. Bakar, M.I. Channa, K. Haseeb, A.W. Khan, A trust aware routing protocol for energy constrained wireless sensor network, Telecommun. Syst. 61 (1) (2016) 123–140.
[15] G. Han, J. Jiang, L. Shu, J. Niu, H.C. Chao, Management and applications of trust in wireless sensor networks: a survey, J. Comput. System Sci. 80 (3) (2014) 602–617.
[16] J. Jiang, G. Han, F. Wang, L. Shu, An efficient distributed trust model for wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 26 (5) (2015) 1228–1237.
[17] T. Wang, Y. Li, W. Fang, W. Xu, J. Liang, Y. Chen, X. Liu, Comprehensive trustworthy data collection approach in sensor-cloud system, IEEE Transactions on Big Data, 2018.
[18] P. Li, J. Li, Z. Huang, T. Li, C.Z. Gao, S.M. Yiu, K. Chen, Multi-key privacy-preserving deep learning in cloud computing, Future Gener. Comput. Syst. 74 (C) (2017) 76–85.
[19] J. Sidhu, S. Singh, Improved TOPSIS method based trust evaluation framework for determining trustworthiness of cloud service providers, J. Grid Comput. 15 (1) (2016) 1–25.
[20] V.V. Rajendran, S. Swamynathan, Hybrid model for dynamic evaluation of trust in cloud services, Wirel. Netw. 22 (6) (2016) 1–12.
[21] R.N.P.Z.J. Bonomi, Flavio Milito, Fog computing: a platform for internet of things and analytics, in: Big data and internet of things: A roadmap for smart environments, 2014, pp. 169–186.
[22] M.J. Guezguez, S. Rekhis, N. Boudriga, A sensor cloud architecture for healthcare applications, in: ACM Symposium on Applied Computing, 2016, pp. 612–617.
[23] T. Ojha, S. Misra, N.S. Raghuwanshi, Sensing-cloud: leveraging the benefits for agricultural applications, Comput. Electron. Agric. 135 (2017) 96–107.
[24] O. Savas, G. Jin, J. Deng, Trust management in cloud-integrated wireless sensor networks, in: International Conference on Collaboration Technologies and Systems, 2013, pp. 334–341.
[25] C. Zhu, H. Nicanfar, V.C.M. Leung, L.T. Yang, An authenticated trust and reputation calculation and management system for cloud and sensor networks integration, IEEE Trans. Inf. Forensics Secur. 10 (1) (2014) 118–131.
[26] C. Zhu, V.C.M. Leung, L.T. Yang, L. Shu, J.J.P.C. Rodrigues, X. Li, Trust assistance in sensor-cloud, in: Computer Communications Workshops, 2015.
[27] M. Henze, R. Hummen, R. Matzutt, K. Wehrle, A trust point-based security architecture for sensor data in the cloud, Trusted Cloud Comput. (2014) 77–106.
[28] F. Bao, I.R. Chen, M.J. Chang, J.H. Cho, Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection, IEEE Trans. Netw. Serv. Manag. 9 (2) (2012) 169–183.
[29] V.R. Prabha, P. Latha, Enhanced multi-attribute trust protocol for malicious node detection in wireless sensor networks, Sdhan 42 (2) (2017) 1–9.
[30] K. Gokulnath, R. Uthariaraj, A survey on trust models in cloud computing, Indian J. Sci. Technol. 9 (47) (2016).
[31] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Edition of the Mcc Workshop on Mobile Cloud Computing, 2012, pp. 13–16.
[32] L. Rodero-Merino, L. Rodero-Merino, Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing, ACM, 2014, pp. 27–32.
[33] J.H. Ho, H.C. Shih, B.Y. Liao, S.C. Chu, A ladder diffusion algorithm using ant colony optimization for wireless sensor networks, Inform. Sci. 192 (6) (2012) 204–212.

Tian Wang received his B.Sc. and M.Sc. degrees in Computer Science from the Central South University in 2004 and 2007, respectively. He received his Ph.D. degree in City University of Hong Kong in 2011. Currently, he is a professor in the Huaqiao University of China. His research interests include wireless sensor networks, fog computing and mobile computing.

Guangxue Zhang received his B.S. degree in Liaoning University of Technology in 2016. Currently, he is a master candidate in the National Huaqiao University of China. His research interests include wireless sensor networks, mobile computing and Fog Computing.

Md Zakirul Alam Bhuiyan received the Ph.D. degree and the M. Eng. degree from Central South University, China, in 2009 and 2013 respectively, and the B.Sc. degree from International Islamic University Chittagong, Bangladesh, in 2005, all in Computer Science and Technology. He is currently an assistant professor (research) in the Department of Computer and Information Sciences at Fordham University. He is a member of the Center for Networked Computing (CNC). Earlier, he worked as a post-doctoral fellow at the Central South University, China, a research assistant at the Hong Kong PolyU, and a software engineer in industries. His research focuses on dependable cyber–physical systems, wireless sensor network applications, network security, and sensor–cloud computing. He has served as a managing guest editor, program chair, workshop chair, publicity chair, TPC member, and reviewer of international journals/conferences. He is a member of IEEE and a member of ACM.

Anfeng Liu is a Professor of School of Information Science and Engineering, Central South University, China. He is also a Member (E200012141M) of China Computer Federation (CCF). He received the M.Sc. and Ph.D. degrees from Central South University, China, 2002 and 2005 respectively, both majored in computer science. His major research interests are Cyber–Physical Systems, Service network, wireless sensor network.

Weijia Jia is currently a Chair Professor at University of Macau while he is taking no-pay leave from the Department of Computer Science & Technology, Shanghai Jiao Tong University, China. He received the B.Sc. and the M.Sc. degrees in computer science from Central South University, Changsha, China, in 1982 and 1984, and Master of Applied Sci. and Ph.D. degrees from Polytechnic Faculty of Mons, Belgium, in 1992 and 1993, respectively. He joined the German National Research Center for Information Science (GMD) in Bonn (St. Augustine) from 1993 to 1995 as a research fellow. From 1995 to 2013, he has worked in Department of Computer Science, City University of Hong Kong as a full professor. His research interests include smart cities, wireless communication and networks, next-generation Internet of Things, cyberspace sensing, smart cities, distributed systems, QoS and routing protocols for the Internet. He has published more than 400 papers in various IEEE Transactions and prestige international conference proceedings. He has served as an editor and guest editor for international journals and as a PC chair and PC member/keynote speaker for several international conferences.

Mande Xie was born in 1977 and is currently a Professor in the Zhejiang Gongshang University. He received the PhD degree in Circuit & System from Zhejiang University in 2006. His research interests include Wireless Sensor Networks (WSNs), Social Network and Privacy Preservation.