Open access • Proceedings Article • DOI:10.1109/ICACT.2005.245914

Trust evaluation model for wireless sensor networks — Source link

Junbeom Hur, Younho Lee, Hyunsoo Yoon, Daeseon Choi ...+1 more authors

Institutions: KAIST, Electronics and Telecommunications Research Institute

Published on: 11 Jul 2005 - International Conference on Advanced Communication Technology

Topics: Key distribution in wireless sensor networks, Mobile wireless sensor network, Wireless sensor network, Wireless network and Sensor web

Trust Evaluation Model for Wireless Sensor Networks

Junbeom Hur[†], Younho Lee[†], Hyunsoo Yoon[†], Daeseon Choi[‡] and Seunghun Jin[‡]

Division of Computer Science
[†]Korea Advanced Institute of Science and Technology,
373-1 Guseng-dong, Yuseong-gu, Daejeon 305-701, Rep. of Korea
[‡]ETRI(Electronics and Telecommunications Research Institute),
161 Gajeong-dong, Yusung-gu, Daejeon, 305-350, Rep. of Korea
Email: {jbhur,yhlee,hyoon}@camars.kaist.ac.kr
{sunchoi,jinsh}@etri.re.kr

***Abstract*—Wireless sensor networks offer many beneficial applications in various fields. However, because sensor devices are limited in their resources and susceptible to a variety of novel attacks, even a few malicious adversaries can easily spread deceitful data and make the networks be in confusion without great efforts. Therefore, it is essential to distinguish forged data of illegal nodes from innocent data of legal nodes in sensor networks. In this paper, to make resilient wireless sensor networks, we propose a trust evaluation model which can identify trustworthiness of sensor nodes in order to filter out malicious nodes' deceitful data.**

***Keywords*—security, trust evaluation, sensor network**

1. Introduction

Wireless sensor networks suggest potentially beneficial solutions for various applications [1]. A major feature of these systems is that sensor nodes in networks assist each other by passing data, in-network process and control packets from one node to another. It is often termed an infrastructure-less, self-organized, or spontaneous network [2].

Because wireless sensor networks pose some unique challenges, traditional security techniques cannot be applied directly to the sensor networks. First, each sensor node is limited in its memory, battery life, computation, and communication capabilities [3]. Therefore, computation-intensive techniques like public-key cryptography are not expected to be used in wireless sensor networks. Second, they are susceptible to a variety of attacks, for example node capture, eavesdropping, denial of services, wormhole, and sybil attack [4]. A major purpose of the active attackers is to make the entire or partial networks impractical or make the networks under the control of them. If the attacker can obtain their own commodity sensor nodes and induce the networks to accept them as legitimate nodes, it is hard to distinguish legitimate nodes from illegitimate ones just through the current network security policies [3]. In addition, such a distinction is also beyond the ability of the conventional key management scheme because we cannot guarantee the secrecy of each node's private key.

Therefore, some smart trust management schemes are needed to identify trustworthiness of sensor nodes in order to distinguish between malicious nodes and innocuous nodes, and to strengthen reliable nodes and weaken suspicious nodes. However, there have not been many of researches for trust evaluation models which are applicable to wireless sensor networks properly.

Here, we propose a trust evaluation model for resilient wireless sensor networks, which helps the networks to operate normally with high probability although some nodes or data would be compromised. General direction for resilience is to gather multiple and redundant sensing data and crosscheck them for consistency. Based on the result of that crosschecking, each node estimates its neighbor nodes' trust values.

The rest of the paper is organized as follows. Section 2 describes goals and assumptions of a proposed trust evaluation model. Section 3 describes specific approaches and, details an overall framework and protocol for a trust management scheme. Section 4 analyzes the performance evaluation. Section 5 describes some related work, and Section 6 remarks conclusion of the paper.

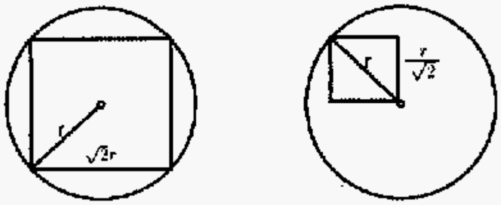
2. Goals and Assumptions

2.1 Threat Model

As a general wireless sensor network environment, sensor nodes in the network are deployed in open areas, so they are confronting the added risk of physical attacks. Because sensor networks have many opportunities to interact closely with anonymous adversaries, deceitful data from them can be easily accepted as legal data in the networks. In addition, because each sensor node is vulnerable to a node capture attack, some private keys used for secure communication in the networks can be snatched by active attackers.

2.2 Goals

We focus on making resilient wireless sensor networks which work normally even though some sensor nodes might be compromised. Without any trust evaluation mechanisms, we cannot guarantee the sensor networks to work appropriately even if the networks adopt cryptographic key management approaches. For the purpose of the resilience of sensor networks,



(a) Maximum grid size (b) Minimum grid size

Fig. 1. **Two Boundaries of a Grid Size**

we direct our approaches to evaluate trustworthiness of sensor nodes, and filter out inconsistent and deceitful data from the malicious or compromised nodes.

2.3 Assumption

We have some assumptions in our trust evaluation model as follows: (1) each sensor node has a knowledge of its own location information by using a location-detect system such as the Global Position System (GPS). (2) time is synchronized all through the networks. (3) sensor nodes are deployed densely enough to be able to sense some identical events redundantly with their own neighbor nodes, and (4) the adversaries try to flood inconsistent data in order to make the networks be in confusion and go wrong.

2.4 Our Contributions

We present a novel trust evaluation model for wireless sensor networks. Based on our survey, there are not clear trust evaluation models suitable for wireless sensor networks properly. Our approach is one of the incipient researches on trust evaluation model for wireless sensor networks that can handle and filter out the inconsistent sensing data of the malicious nodes. We expect our trust evaluation model to make a contribution to resilient wireless sensor networks.

3. Our Design: Trust Evaluation Model

We describe the protocol of our trust evaluation model. The protocol consists of four steps. First, we divide sensing areas into some logical grids and assign a unique identification to each grid (Section 3.1). Second, sensor nodes deployed in each grid verify location information of their neighbor nodes by ECHO protocol [6] (Section 3.2). Third, each node evaluates trustworthiness of its neighbor nodes by crosschecking the neighbor nodes' redundant sensing data with its own result. Inconsistent data from malicious or compromised nodes can be detected in this step (Section 3.3). Fourth, special nodes, aggregators, aggregate sensing data from their grids and transmit the computed results to the destination node, sink. Inconsistent data from malicious nodes can be excluded in this step (Section 3.4).

3.1 Step 1: Grid Definition

In this step, consider first sensing areas in which sensor nodes will be deployed and ready for some events. We can easily know the location information of the sensing areas before we deploy sensor nodes. Then, we divide the sensing areas into some logical square grids in proportion to the sensing range of a sensor device. We define r to be the sensing range of a sensor device. The main focus on dividing in this step is to set the size of a logical grid to the extent that one sensor device's sensing range can cover a grid entirely it belongs to. Two extreme deployment examples of a sensor node are shown pictorially in Figure 1.

Consider an ideal case that a sensor node is deployed at the center of a grid as in Figure 1(a). In that case, for the purpose of the whole coverage by r of the sensor node, a grid can expand its size to $\sqrt{2}r \times \sqrt{2}r$, which is the maximum size that a grid can extend to. On the other hand, in such a case that a sensor node is deployed at the apex of a grid as in Figure 1(b), a grid can expand its size only to $\frac{r}{\sqrt{2}} \times \frac{r}{\sqrt{2}}$, which is the minimum size that a grid can extend to.

In those cases, there is a tradeoff between correctness and economy. Although there can be so many choices between Figure 1(a) and (b), in our model, we intend to use as many redundant sensing data from multiple sensor nodes as possible to identify inconsistent data among them. So, for the higher correctness of the crosscheck, we choose the case of Figure 1(b), and one grid size is set to $\frac{r}{\sqrt{2}} \times \frac{r}{\sqrt{2}}$.

After dividing sensing areas into some logical grids, we assign a unique identification to each grid. In step 2, sensor nodes can be identified by their own locations and grid identifications which are assigned in this step.

3.2 Step 2: Location Verification

In this step, each sensor node verifies its neighbor nodes' claimed locations. In order to verify location claims, we adopt the Echo protocol proposed by Sastry, et al [6].

We use s to represent the speed of sound, or 331 m/s. Likewise, we use c to represent the speed of light, or 3×10^8 m/s. We define $d(x, y)$ to be the distance between x and y. we use v to represent verifier which would like to verify the location of a prover p. We define l to be the p's claimed location and Δ_p to be the processing delay of p.

All sensor nodes are deployed in sensing areas which are logically divided into several grids in step 1. Then, each sensor node checks its own deployed grid and location with GPS and floods a HELLO message which containing a packet <Grid identification, Position> to announce itself to its neighbor nodes. Verifiers, v, receiving such packets verify each packet information of the neighbor provers, p, using Echo protocol.

According to [6], Echo protocol can be described as follows: the prover p first broadcasts its claimed location l and processing delay Δ_p to the verifier v. If the v can validate the claim, it broadcasts a nonce to the p; the p echoes the nonce back to the v over ultrasound. The v again computes the elapsed time of this communication: if it is no greater than the time for the signal to travel allowing for processing delay, the v accepts the claim.

However, by the Echo protocol, each sensor node can verify only that whether its neighbor nodes are within their claimed

ID_1	$D_{0,1}$	ss_1	sf_1	S_1	sr_1	st_1	R_1	cs_1	is_1	C_1	B_1	T_1
						⋮						
ID_k	$D_{0,k}$	ss_k	sf_k	S_k	sr_k	st_k	R_k	cs_k	is_k	C_k	B_k	T_k

Fig. 2. **Trust Evaluation Matrix**

distance, that is, in a specific range. Sensor nodes are incapable of verifying exact positions of their neighbor nodes. Instead, they can only convince themselves of the legitimate neighborhood of the provers. More than three neighboring verifiers can affirm that whether the prover is really located in its claimed grid. In our model, because we assume the networks to be dense enough to be able to sense an identical event redundantly from multiple neighbor nodes, we can expect an exact verification of the neighbor nodes' positions claimed in HELLO messages with high probability.

The purpose of the location verification in this step is to avoid several attacks which deceive every node in the networks into believing that an adversary is its neighbor or located in a specific position, such as sinkhole attack, wormholes, Sybil attack, and HELLO flood attack [4]. Because such attacks make use of false location information to make the networks impractical, location verification process in this step can protect the networks from such attacks.

3.3 Step 3: Trust Evaluation

In this step, sensor nodes evaluate trustworthiness of other nodes. However, each sensor node does not compute all the other nodes' trust values in the networks [5], but computes only its neighbor nodes' trust values accumulatively.

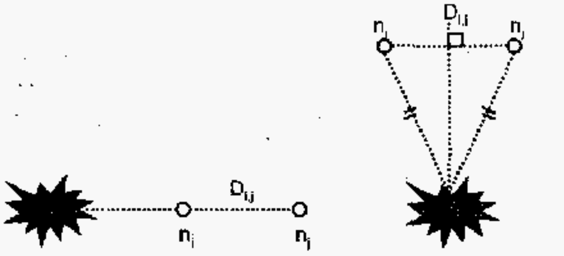
Each sensor node has a trust evaluation matrix which stores the trust evaluation factors for its neighbor nodes. The trust evaluation matrix is shown in Figure 2. The node 0 has k trust evaluation matrices for its k neighbor nodes as in Figure 2. The trust evaluation matrix consists of several trust evaluation factors as follows.

1) Identification: This factor contains identification information of a node. It consists of a node's position and grid identification in which it deployed.
   - $ID_i =< GridID, Position_i >$, where $1 \leq i \leq k$
2) Distance: This factor contains distance information between two nodes. x_i means x coordinate and y_i means y coordinate of node i.
   - $D_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, where $0 \leq i, j \leq k$ and $i \neq j$
3) Sensing communication: This factor contains communication ratio information. When a node detects a certain event, if its neighbor nodes also detect the same event and broadcast the sensing results, communication ratio values for those neighbor nodes go up. If they do not communicate, communication ratio values for those nodes go down. This factor represents the level of selfishness and normality of a node. If a node does not participate in communication in the networks continuously for its battery saving or some other troubles, its trust value will be degraded.
   - S_i: sensing communication value of node i, where $1 \leq i \leq k$
   - ss_i: sensing success count of node i
   - sf_i: sensing failure count of node i
4) Sensing result: This factor represents sensing result information for detected events. This factor consists of sensing data and sensing time for the events. The information of this factor is used to check a consistency of each sensor node and to detect illegal or compromised nodes in the networks.
   - $R_i =< sr_i, st_i >$: sensing result value of node i, where $1 \leq i \leq k$
   - sr_i: sensing data of node i
   - st_i: sensing time of node i
5) Consistency: This factor represents a level of consistency of a node. Based on this factor, we can identify malicious or compromised nodes, and filter out their data in the networks.
   - C_i: Consistency value of node i, where $1 \leq i \leq k$
   - cs_i: consistent sensing count of node i
   - is_i: inconsistent sensing count of node i
6) Battery: This factor represents remained lifetime of a sensor node. As we compute trust values in consideration of this factor, we can reduce additional processes which would be necessary to handle some power-managing policies. In addition, some nodes which have high trust values are likely to process more jobs than the other nodes which have low trust values. In that case, the higher trust value a node has, the earlier the node meets its end. According to the adoption of this battery factor, we can prevent such a biased battery exhaustion.
   - B_i: Battery value of node i, where $1 \leq i \leq k$
7) Trust value: This factor represents a total trustworthiness of a node, which is evaluated based on the other trust evaluation factors. Trust value of a node is dynamic because the values of each trust evaluation factor change with the lapse of time.
   - T_i: Trust value of node i, where $1 \leq i \leq k$

Next, we propose a novel inconsistency check mechanism, trust quantification method, and trust computation method.

1) Inconsistency Check: Here, we introduce a general inconsistency check mechanism in detail. The mechanism uses sensing results in R_i. The inconsistency check result affects the value of consistency factor, C_i.

When node j checks the inconsistency of its neighbor node i's sensing results, if the results are out of relatively standard bound of node j, node j estimates the results to be inconsistent or deceitful data. Such an estimation for its neighbor, node



(a) Maximum sensing difference (b) Minimum sensing difference

Fig. 3. **Two Boundaries of Sensing Difference**

i, affects the value of the consistency factor, C_i, in trust evaluation matrix.

Two extremely opposite boundaries of sensing difference are represented in Figure 3. If the distance between node i and node j, $D_{i,j}$, is fixed, the case in which difference of the sensing results from node i and node j will be maximum is that one node is located on the extended straight line from an event to the other node. We define $\theta_{i,j}$ as the upper boundary of the difference which can be acceptable as consistent sensing result between node i and node j. Figure 3(a) shows the case that sensing difference can be maximized.

On the other hand, the case in which difference of the sensing results from node i and node j will be minimum is that the distance between an event and node i and the distance between the same event and node j are equal. Figure 3(b) shows that case. In that case, we can expect no discrepancy between two sensing results of node i and node j under ideal conditions. Therefore, the lower boundary of the difference of sensing results is 0.

We define a maximum difference function, f, which offers the upper boundary of the difference between two nodes' sensing results for a same event on condition of preserving consistency. In our model, every sensor node knows this maximum difference function and makes use of it to check legality of its neighbor nodes' sensing data. Proposed maximum difference function is the following:

$$\theta_{i,j} = f(D_{i,j}, prior(R_i, R_j)), \tag{1}$$

where $prior(R_i, R_j)$ returns sensing result of a node which sense an identical event prior to the other node. For example, if $st_i < st_j$, then $prior(R_i, R_j)$ returns a sensing result value of node i, sr_i.

When a sensor node senses an event, it broadcasts its identification and sensing data, $< ID, R >$, to its neighbor nodes. If a node receives information of the sensing data and sensing time from its neighbor nodes, it can check whether the received data from neighbor nodes can be acceptable as consistent data or unacceptable as inconsistent data by comparing them with its own sensing data for the same event.

We consider two main cases in sensing environment from a local point of view. First case is that two neighbor nodes, n_i and n_j, succeed in sensing an identical event simultaneously. In this case, from a n_i's point of view, n_i increases sensing success count value for n_j by 1, that is $ss_j = ss_j + 1$, and checks consistency of the n_j's sensing data and assigns the checking result to its corresponding trust evaluation factor for n_j. The consistency check processes are as follows:

1) If $0 \leq sr_i - sr_j \leq \theta_{i,j}$, then $cs_j = cs_j + 1$, where $st_i \leq st_j$.
2) If $sr_i - sr_j < 0$ or $sr_i - sr_j > \theta_{i,j}$, then $ic_j = ic_j + 1$.

Second case is that two neighbor nodes fail to sense an identical event simultaneously. In this case, on equal terms, n_i and n_j increase sensing failure count value for n_j and n_i by 1 respectively, that is $sf_j = sf_j + 1$ and $sf_i = sf_i + 1$.

In Equation (1), our proposed function, f, remains to be black box. It is because the maximum difference function may have to be dynamic for the applications it is adopted. Of course, parameters of the function can be changed dynamically according to its application. Due to such an application-dependent feature, we propose a maximum difference function as black box.

2) Trust Quantification: Here, discrete values of each trust evaluation factor are transformed into continuous values from -1 to +1. -1 and +1 mean complete distrust and complete trust respectively. As a node communicates and revalues trust factor values for their neighbor nodes continuously, trust quantification process is imperative for impartial comparison among each node's trust values. Trust quantification processes for each trust evaluation factor are as follows:

1) Consistency value

$$C_i = \frac{cs_i - is_i}{cs_i + is_i}, \quad \text{where } -1 \leq C_i \leq 1. \tag{2}$$

2) Sensing communication value

$$S_i = \frac{ss_i - sf_i}{ss_i + sf_i}, \quad \text{where } -1 \leq S_i \leq 1. \tag{3}$$

3) Battery value

$$B_i : -1 \leq B_i \leq 1,$$

where each sensor node broadcasts quantification value of its own B_i.

3) Trust Computation: Trust computation involves an assignment of weights to the trust factors that are evaluated and quantified in trust quantification step. We define W_i as a weight which represents importance of a particular factor from 0, unimportant, to +1, most important. The weight is dynamic and dependent on the application.

Trust value for node i is computed by the following equation:

$$\text{If } B_i \neq -1, \quad T_i = \frac{W_1 C_i + W_2 S_i + W_3 B_i}{\sum_{i=1}^{3} W_i}, \tag{4}$$

where $0 < W_i \leq 1$. In case of $H_i = -1$, we just assign -1 to T_i and exclude the node from the networks because it totally cannot work in the networks.

Because each sensor node uses histograms for the accumulative trust evaluation, some malicious or compromised nodes that broadcast inconsistent or deceitful data continuously can be detected and classified in this step.

3.4 Step 4: Data Aggregation

In this step, we propose a data aggregation scheme based on trust value of each node evaluated in step 3. Sensing data of multiple nodes are aggregated per grid. To aggregate data, we elect one node as an aggregator per each grid. Then, the aggregator obtains sensing data from the other member nodes in its grid and aggregates them to a representative value in consideration of the trust values of member nodes. Detailed data aggregation processes are as follows:

1) Aggregator Selection: Prior to a data aggregation, sensor nodes elect an aggregator node in their own grid, which has the highest trust value among all the nodes in an identical grid by the majority of vote. Aggregators are electd periodically with some application-dependent time interval and changed dynamically. The roles of an aggregator are to get sensing data from member nodes together, output a representative sensing result, and transmit it to the sink node.

After selected as an aggregator, the aggregator, node a, sends its own identification, $ID_a =< GridID, Position >$, to the sink node.

2) Trust Agreement: Because the trust value of a node is evaluated by its neighbor nodes, a trust agreement process is necessary prior to the data aggregation.

An aggregator requests its neighbor nodes in and out of its grid to notify the aggregator itself of its member nodes' trust values. Next, request-received neighbor nodes, for example node j, who have a knowledge of trust values for those member nodes, node i, reply to the aggregator in the form of $< ID_j, T_i >$. Then, the aggregator gathers up all information for its member nodes' trust values from the repliers and evaluate member nodes' trust values in proportion to the trust values of repliers themselves that the aggregator knows by this equation:

$$T_i = \frac{\sum_{j=1}^{k}(T_j + 1) \times T_{i,j}}{\sum_{j=1}^{k}(T_j + 1)}, \qquad (5)$$

where k means the number of repliers, and $T_{i,j}$ means a trust value for node i received from node j.

3) Data Aggregation: Sensing data from multiple nodes are aggregated in consideration of the agreed trust values of member nodes per each grid. Because the data aggregation is based on trust values, deceitful data from malicious or compromised nodes whose trust values are lower than those of the other legal nodes can be excluded in this step naturally. An aggregator of each grid aggregates sensing data from its member nodes by this equation:

$$SR_{GridID} = \frac{\sum_{i=1}^{m}(T_i + 1)sr_i}{\sum_{i=1}^{m}(T_i + 1)}, \qquad (6)$$

where m is the number of nodes in a grid including the aggregator itself.

This data aggregation process is executed only when more than half member nodes of a grid sense an identical event simultaneously. It is for reducing some redundant communications between an aggregator and the sink.

4) Data Transmission: After aggregating sensing data from its member nodes, each aggregator sends the aggregated data to the sink with its identification, $< ID_a, SR_{GridID} >$.

4. Analysis

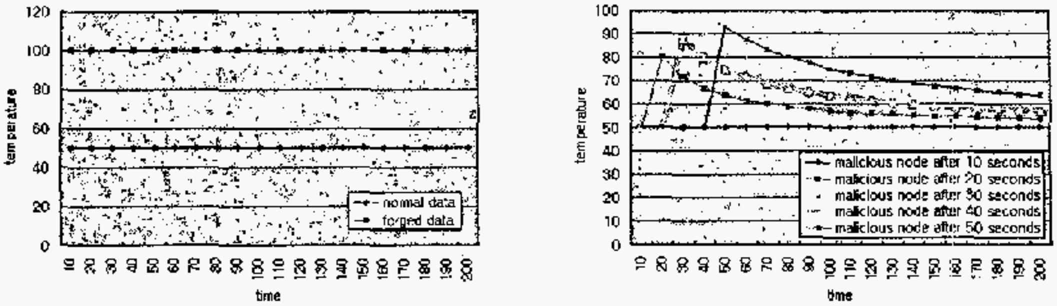
We evaluate the trust evaluation model in terms of its efficiency of excluding forged data in the network. This analysis shows how much time it needs to filter inconsistent data out in the network. We implement and simulate a temperature sensing system. The environments of the system are as follows. 300 sensor nodes are uniformly distributed at the sensing area whose size is $500m \times 500m$. Sensing range of a sensor device is $70m$ and a grid size is $50m \times 50m$. In this analysis, we assume that all sensor nodes have a same amount of battery power and participate in communication positively regardless of their roles. So, we consider only a consistency evaluation factor. The results are shown in Figure 4.

In this simulation, a same event occurs every 10 seconds in an identical grid and we focus on that grid. As we can see Figure 4(a), normal aggregated data of the grid is 50°C, but a forged aggregated result is 100°C by a single malicious node which broadcasts four times as high as a normal sensing result. This indicates the vulnerability of a system without a trust evaluation scheme. Figure 4(b) shows the process of filtering inconsistent data of a malicious node which acts inconsistently after certain seconds with a proposed trust evaluation scheme. The earlier the system detects a malicious node, the lower the forged data of it can effect the aggregated result.

5. Related Work

5.1 Trust Evaluation Model

Current security researches for trust management schemes mainly focus on more powerful ad hoc networks than sensor networks. Z. Yan, P. Zhang, and T. Virtanen proposed a trust evaluation model [5]. In this model, each node should evaluate trust values of all the other nodes in the networks. Such a global computation cannot be accomplished in practical resource-constraint sensor networks. In addition, trust evaluation factors used in that model cannot reflect malice of the illegal nodes, rather just check experience statistics such as communication success, reference count, and personal preference. So, the previous trust evaluation model cannot filter out maliciously forged inconsistent data in the networks.



(a) Without trust evaluation

(b) With trust evaluation

Fig. 4. **Sensing Results of a Single Grid**

5.2 Inconsistency Check

Efficient inconsistency check mechanisms are mainly researched in intrusion detection research areas. Generally, intrusion detection systems consider unexpected results or events which are out of their learned pattern as intrusions. In order to train the intrusion detection system, some machine-learning models, for example hidden Markov model, are adopted to the system and the system is trained by a large number of training data [10]. Such an anomaly detection scheme is necessary in wireless sensor networks to find out malicious or compromised sensor nodes which act inconsistently. However, how to define such an anomaly model based on which training data is still a main challenge.

5.3 Key Management

Because of the infrastructure-less and resource-constrained features of sensor networks, traditional asymmetric key mechanisms, such as digital signature and public key encryption, are seldom applied to sensor networks. So, key management schemes using a small number of symmetric keys, while security level of the system is still remained high, are studied and proposed for wireless sensor networks [11],[12]. However, such a key management scheme alone cannot help legal nodes in the networks to identify legitimacy of neighbor nodes which they communicate with. Moreover, it is a reasonable assumption that some nodes are likely to deprived of secret keys by physical attacks [3]. So, a novel trust management scheme is necessary for secure and resilient wireless sensor networks.

6. Conclusion

We proposed a trust evaluation model for wireless sensor networks. As we referred, the security for wireless sensor networks is still in its infancy and there are not clear trust evaluation models which can be applied to sensor networks properly. Our model does not employ cryptographic approaches or certification mechanisms, so it is light enough to fit well with wireless sensor networks without great overheads. To the best of our knowledge, our approach is one of the incipient researches on trust evaluation model for wireless sensor networks that can detect malicious and compromised sensor nodes, and filter out the inconsistent sensing data of them. We expect that our trust evaluation model can help to make resilient wireless sensor networks.

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the advanced Information Technology Research Center(AITrc) and University IT Research Center Project.

References

[1] H. Chan and A. Perrig, *Security and Privacy in Sensor Networks*, IEEE Computer 2003.

[2] A. Pirzada, C. McDonald, *Establishing Trust In Pure Ad-hoc Networks*, Proceedings of the 27th conference on Australasian computer science, 2004.

[3] A. Perrig, J. Stankovic, D. Wagner, *Security in Wireless Sensor Networks*, Communication of the ACM, June 2004.

[4] C. Karlof, D. Wagner, *Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures*, NEST 2003.

[5] Z. Yan, P. Zhang, T. Virtanen, *Trust Evaluation Based Security Solution in Ad Hoc Networks*, NordSec 2003, Proceedings of the Seventh Nordic Workshop on Secure IT Systems, 15th-17th October 2003.

[6] N. Sastry, U. Shankar, D. Wagner, *Secure Verification of Location Claims*, Proceedings of the 2003 ACM workshop on Wireless security.

[7] B. Przydatek, D. Song, A. Perrig, *SIA: Secure Information Aggregation in Sensor Networks*, SenSys 2003.

[8] B. Krishnamachari, D. Estrin, S. Wicker, *The Impact of Data Aggregation in Wireless Sensor Networks*, ICDCSW, Proceedings of the 22nd International Conference on Distributed Computing Systems, pp. 575-578, 2002.

[9] J. Deng, R. Han, S. Mishra, *Security Support for In-Network Processing in Wireless Sensor Networks*, Conference on Computer and Communications Security, Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, pp. 83–93, 2003.

[10] S.S. Doumit, D.P. Agrawal, *Self-Organized Criticality and Stochastic learning based intrusion detection system for wireless sensor networks*, Military Communications Conference, 2003. MILCOM '03. 2003 IEEE, pp.609–614.

[11] H. Chan, A. Perrig, D. Song, *Random key predistribution schemes for sensor networks*, IEEE Symposium on Security and Privacy, Berkely, California, May 11-14 2003, pp. 197-213.

[12] W. Du, J. Deng, Y.S. Han, P.K. Varshney, *A pairwise key pre-distribution scheme for wireless sensor networks*, Proceeings of the 10th ACM Conference on Computer and Communications Security (CCS), Washington, DC, USA, October 27-31 2003, pp. 42-51.