# A Novel Trust Establishment Method for Wireless Sensor Networks

**Farruh Ishmanov[1] and Sung Won Kim[2]**
[1] Deptartment of Electronics and Communication Engineering, Kwangwoon University
447-1 Wolgye-dong, Nowon-gu, Seoul, Republic of Korea
[e-mail: farruh.uzb@gmail.com]
[2] Department of Information and Communication Engineering, Yeungnam University
Deahakro 280, Gyeongsan-si, Gyeongsangbuk-do, 712-749, Republic of Korea
[e-mail:swon@yu.ac.kr]
*Corresponding author: Sung Won Kim

*Received October 14, 2014; revised February 14, 2015; accepted March 10, 2015; published April 30, 2015*

## Abstract

Establishment of trust is important in wireless sensor networks for security enhancement and successful collaboration. Basically, a node establishes trust with other nodes by estimating a trust value based on monitored behavior of the other nodes. Since a malicious/misbehaving node might launch different attack strategies and might demonstrate random misbehavior, a trust estimation method should be robust against such attacks and misbehavior. Otherwise, the operation of trust establishment will be meaningless, and performance of an application that runs on top of trust establishment will degrade. In this paper, we propose a robust and novel trust estimation method. Unlike traditional trust estimation methods, we consider not only the weight of misbehavior but also the frequency of misbehavior. The frequency-of-misbehavior component explicitly demonstrates how frequently a node misbehaves during a certain observed time period, and it tracks the behavior of nodes more efficiently, which is a main factor in deriving an accurate trust value. In addition, the weight of misbehavior is comprehensively measured to mitigate the effect of an on-off attack. Frequency and weight of misbehavior are comprehensively combined to obtain the trust value. Evaluation results show that the proposed method outperforms other trust estimation methods under different attacks and types of misbehavior.

***Keywords:*** Security, trust establishment, malicious node, misbehavior

**http://dx.doi.org/10.3837/tiis.2015.04.015** **ISSN : 1976-7277**

## 1. Introduction

Trust establishment is one of the recent research trends in many fields, such as e-commerce, web-based services, peer-to-peer networks, and wireless networks. Recently, different trust establishment (TE) methods, technologies and mechanisms, such as fuzzy logic [1-2], bio-inspired [3-4], deterministic- and probabilistic-based approaches [5-9] have been proposed for wireless sensor networks (WSNs). In general, TE can be used in WSNs for two purposes: cooperation improvement and security enhancement [5-8]. Cooperation of sensor nodes in WSNs is vital to maintaining operation of the network [10]. From this perspective, it is important to maintain successful collaboration among sensor nodes. Successful collaboration is assured only when all nodes operate in a trustworthy manner [5-7]. TE maintains successful collaboration by detecting trustworthy and untrustworthy nodes, and evaluating them based on their behavior/performance. Moreover, because WSNs are usually deployed in remote and unattended areas, and nodes often lack tamper-resistant hardware, they can be captured physically and easily compromised. Once a node is compromised, security techniques like cryptography and authentication fail to protect the network. Thus, TE can continuously monitor and evaluate node behavior and detect such compromised nodes.

The core of trust establishment is trust estimation. Basically, trust is periodically estimated based on the numbers of good and bad behavior recorded during a certain time interval in the WSN [3-9]. In addition, the number of good and bad behavior during previous time intervals are added, but with a forgetting factor [3-9]. The proportion of number of bad behaviors to the number of good and bad behavior determines the rate of misbehavior. Measured rates of misbehaviors in current and previous time periods are combined to obtain the weight of misbehavior. Hence, if the weight of misbehavior is high, then the trust value will be low; otherwise, it will be high. Thus, the trust value is derived solely based on the weight of misbehavior. This method cannot evaluate node misbehavior correctly, because the frequency of misbehavior is not considered in the trust estimation. Hence, according to traditional trust mechanisms, if the weight of misbehavior is low, even though the node misbehaves for a long time, its trust value will always be high. However, it is important to detect such misbehavior in WSNs because many nodes are stuck malfunctioning due to faults in software and hardware [8]. Moreover, considering only the weight of the misbehavior in trust estimation does not cope with a malicious node's different strategies to trick the trust establishment scheme. In order to detect persistent misbehavior, we previously proposed a trust estimation method that estimates a trust value based on aggregate misbehavior over time [11]. Even though it can efficiently track and evaluate persistent misbehavior, it cannot explicitly show how frequently a node is misbehaving. Another way to trick trust establishment is to use an on-off attack. Although this type of attack can efficiently compromise the operation of trust establishment, many trust establishment schemes do not consider it [1-4, 6, 8]. The nature and type of on-off attack make it difficult for a trust mechanism to detect the attack. In an on-off attack, there are two states: on and off. During the on state, the malicious node misbehaves; thus, the trust mechanism detects the number of bad behavior and records them. On the other hand, during the off state, the malicious node behaves well and the trust mechanism detects the good behavior and records it. Then, the trust mechanism takes the proportion of the observed good and bad behavior, and combines it with a previous proportion weighted by a forgetting factor. In this case, a smart attacker can manipulate the number of good and behavior; that is, it can keep the bad behavior low to get a higher trust value and not be detected. Moreover, an

attacker can change the on-off attack strategy by regulating the duration of on and off states and the number of bad and good behavior during the attack, based on weak points of the trust mechanism. To combat an on-off attack, one of the prominent methods is to regulate the forgetting factor [5, 9, 11, 12]. The idea behind such an approach is to slowly erase the previous bad behavior, compared to good behavior, so the malicious node gets fewer chances to misbehave/attack. Although this method can mitigate the effect of an on-off attack, it cannot detect it efficiently, as our evaluation results show.

In this paper, considering the aforementioned problems and shortcomings of our previous work, we propose a novel and lightweight solution. We propose considering frequency of misbehavior in the trust estimation as a new component. To the best of our knowledge, this is the first trust mechanism that considers frequency of misbehavior along with the weight of the misbehavior to estimate a trust value for a node. The frequency-of-misbehavior component can efficiently deal with persistent and random misbehavior. Frequency of misbehavior is measured during a certain time interval, $t_k$, which is further divided into several equal time units. Each time unit is defined either as an off or an on period, based on the rate of misbehavior. If the rate of misbehavior during time unit $j$ is above a certain threshold, then time unit $j$ is counted as an on period. Otherwise, it is counted as an off period. In this way, the number of on and off periods is determined during the $t_k$ period. Then, the frequency of misbehavior is estimated by dividing the number of on periods by the sum of on and off periods. In order to update the measured misbehavior frequency, a sliding time window is used. After each $\Delta$ time period, the time window slides to the right, dropping the first time unit and adding another time unit at the end of the window. Moreover, after each $\Delta$ time period, the weight of the misbehavior is estimated by combining measured misbehavior in the current and previous time units. Hence, after each $\Delta$ time period, the trust value is estimated based on the measured frequency and weight of the misbehavior. Employing a misbehavior-frequency component assists in explicitly stating how frequently a node is misbehaving and improves on-off attack detection. Moreover, it enables detection of persistent malicious nodes, regardless of their rate of misbehavior.

We prove the correctness and efficiency of our proposed method through comprehensive performance evaluations. Evaluation results show that the proposed method can detect many kinds of persistent misbehavior. Moreover, under different scenarios of the on-off attack, the proposed method demonstrates a more balanced and higher detection rate compared to previously proposed schemes.

The remainder of this paper is organized as follows. In Section 2, we present an overview of related work. Section 3 describes the proposed trust establishment method. Evaluation results of the proposed scheme are provided in Section 4, and finally, Section 5 concludes the paper.

## 2. Related Work

Trust establishment schemes for WSNs can be divided into the following groups based on trust estimation method [13]:

- Probabilistic
- Fuzzy logic
- Weighting
- Miscellaneous

Below, we present some representative TE schemes.

Shaikh et al. [6] proposed one of the earliest comprehensive TE schemes, called the group-based trust management scheme (GTMS) for clustered wireless sensor networks. The scheme works at three levels: the node level, the cluster head level, and the base station level.

At the node level, nodes estimate a trust value for other nodes using a time-window mechanism. The main objective of a time window is to record information and forget previous information. After each Δ period, node $x$ estimates the trust value of node $y$ based on recorded information in time window $t_k$. As the example in **Fig. 1** shows, after each Δ period, the time window slides to the right, recording recent information and forgetting information recorded earlier. The time window in Fig. 1 consists of three time units ($L$=3), with $S_{x,y}$ and $U_{x,y}$ being the good and bad behavior of node $y$ observed by node $x$ within time window $t_k$.

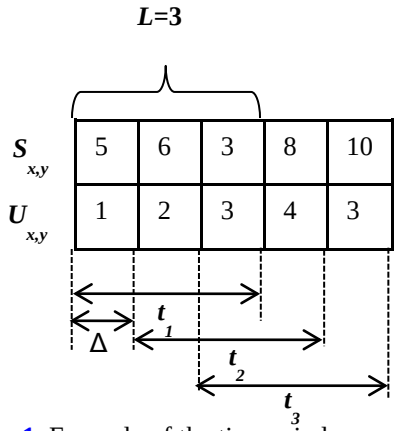



**Fig. 1.** Example of the time-window mechanism.

Using the information in the time window, the trust value of node $y$ per node $x$ is estimated as follows [6] :

$$T_{x,y} = \left[ 100 \times \left( \frac{(S_{x,y})^2}{(S_{x,y} + U_{x,y})(S_{x,y} + 1)} \right) \right] \tag{1}$$

where [] is the nearest integer function, $S_{x,y}$ is the total number of successful interactions by node $x$ with node $y$ during time $t_k$, and $U_{x,y}$ is the total number of unsuccessful interactions by node $x$ with node $y$ during time $t_k$. After estimation of the trust value, a node will quantize trust into three states in the proposed mechanism: trusted, uncertain, and untrusted.

Advantages of this scheme are that it is lightweight and energy-aware, both of which meet the requirements of WSNs. Furthermore, the authors proved that GTMS is resilient against cheating, bad behavior, and group attacks, under the assumption that the number of unsuccessful interactions is equal to, or more than, the number of successful interactions. However, this may not always be true, because an attacking node usually attempts to avoid detection as much as possible. Moreover, the time window is not resilient enough to counter an on-off attack.

A trust management scheme called the lightweight and dependable trust system (LDTS) for clustered wireless sensor networks was proposed [14], which is similar to GTMS [6]. The major difference between GTMS and LDTS lies in the trust estimation method and the feedback estimation and collection method. Specifically, trust is estimated as follows [14]:

$$T_{x,y} = \left[ \left( \frac{10 \times (s_{x,y})}{(s_{x,y} + u_{x,y})} \right) \left( \frac{1}{\sqrt{(u_{x,y})}} \right) \right] \quad (2)$$

where $S_{x,y}$ and $U_{x,y}$ are the numbers of good and bad behaviors observed by node $x$ about node $y$. $\sqrt{(u_{x,y})}$ is to strictly control increases in bad behavior. The trust value approaches 0 rapidly with an increase in the number of bad behavior. Thus, it is also used to defend against an on-off attack. Moreover, the authors proposed using a feedback aggregation method, which is robust against a bad-mouthing attack. However, the computational overhead of the feedback aggregation method was not considered.

Maturity-based trust management for mobile ad hoc networks was proposed by Velloso et al. [15]. The relationship maturity concept was introduced to improve the quality of trust evaluation in the presence of mobility. According to the concept, recommendations by long-term neighbors are given more weight than recommendations by short-term neighbors. The trust level of node $y$ is estimated by node $x$ by combining observation-based trust with recommendations, as follows [15]:

$$T_x(b) = (1-\alpha) \times Q_x(b) + \alpha \times R_x(b) \quad (3)$$

where $Q_x(b)$ is an observation-based trust value from node $x$ about node $y$, and $R_x(b)$ represents the aggregate value of recommendations from all neighbors. The variable $\alpha$ provides a relevant weight to each factor. $Q_x(b)$ is defined as follows [15]:

$$Q_x(b) = \beta \times E_x(b) + (1-\beta) \times T_x(b) \quad (4)$$

where $E_x$ and $T_x$ are currently and previously obtained trust values. The variable $\beta$ provides the necessary weight to each trust value.

Moreover, a recommendation exchange protocol to efficiently manage recommendation exchanges is proposed. It includes three messages: a Trust Request (TREQ) message, a Trust Reply (TREP) message, and a Trust Advertisement (TA) message. TREQ is used to request recommendations from neighbors about a target node. Neighbors of the target node reply with a TREP after waiting a random period of time, $tREP$. The goal of $tREP$ is to avoid collisions and to wait for other TREQs. TA is used to inform neighbors about a rapid change in the trust value of a certain node during a trust update.

Even though this method has advantages, such as improving trust estimation in a mobile environment, the proposed scheme does not include protection against on-off and bad-mouthing attacks. Since these attacks have a direct influence on estimated trust values, not considering their influence leads to incorrect decisions.

One recent trust establishment scheme, attack-resistant and lightweight trust management for medical sensor networks (ReTrust), was proposed by He et al. [9]. Similar to work by Shaikh et al. [6], the He et al. proposal also works based on a clustered network. The entire network is divided into cells in which each cell has member nodes and one manager node. In a cell, node $x$ calculates a trust value for node $y$ using a time window as follows [9]:

$$T_{x,y} = \left[ \alpha \times \left( \frac{\sum_{j=1}^{m} \beta_j \times (1 - p_j) \times p_j}{\sum_{j=1}^{m} \beta_j \times (1 - p_j)} \right) \right] \tag{5}$$

where $\alpha$ scales the range of the trust value, and $L$ is the number of units in a time window. The authors introduced an aging-factor parameter, $\beta_j$, which is different for each time unit $j$ in the window. $\beta_j$ is defined as $\beta_j = \varphi^{L-j}$, where $0<\varphi<1$. $p_j$ is a successful interaction rate, which is estimated as follows [9]:

$$p_j = \frac{S_j + 1}{S_j + U_j + 2} \tag{6}$$

where $S_j$ and $U_j$ are the number of successful and unsuccessful interactions, respectively, during the $j^{th}$ unit of the time window.

Using the time-window mechanism along with the proposed comprehensive aging mechanism makes the trust estimation method robust against an on-off attack. However, like traditional trust estimation methods, ReTrust also does not consider continuity of misbehavior.

Another interesting trust model was proposed for peer-to-peer (P2P) networks by Han et al. [16]. This trust model is built by considering the status and reputation of a peer. The status of the peer is derived by weighting the topological potential of the peer. The details of the status derivation are given [16]. Trust of the peer is estimated as follows [16]:

$$T(u) = Cr(u) + Cr(C_u) + R(u) \tag{7}$$

where $Cr(u)$, $Cr(C_u)$, and $R(u)$ are accumulative credential, the credential of the community, and a recommendation credential for peer $u$, respectively. Although context of the trust model is different from wireless sensor networks, the idea of a community credential and its estimation method can be used for the sensors.

**Table 1** demonstrates a summary of the above-mentioned trust schemes in terms of trust estimation method. As we can see, all trust estimation methods deal with a proportion of the numbers of good and bad behavior, which is a weight of misbehavior. Hence, usually the numerator and denominator of the trust estimation equation represent the number of good behavior and the sum of the number of good and bad behaviors, respectively. Moreover, some trust establishment schemes [12, 16] incorporate a recommendation to estimate trust value.

**Table 1.** Summary of trust establishment schemes.

| Trust establishment method | Trust estimation equation | Comments |
|---|---|---|
| GTMS [6] | $T_{x,y} = \left[ 100 \times \left( \frac{(S_{x,y})^2}{(S_{x,y} + U_{x,y})(S_{x,y} + 1)} \right) \right]$ | $S_{x,y}$ - number of good behavior<br>$U_{x,y}$ - number of bad behavior |
| Velloso et al. [12] | $T_x(b) = (1-\alpha) \times Q_x(b) + \alpha \times R_x(b)$ | $Q_x(b)$ is an observation-based trust value.<br>$R_x(b)$ represents an aggregate value of the recommendations. |
| ReTrust [9] | $T_{x,y} = \left[ \alpha \times \left( \frac{\sum_{j=1}^{m} \beta_j \times (1-p_j) \times p_j}{\sum_{j=1}^{m} \beta_j \times (1-p_j)} \right) \right]$ | $\beta_j \times (1-p_j)$ is used as a forgetting factor.<br>$p_j = \frac{S_j + 1}{S_j + U_j + 2}$ |
| LDTS [15] | $T_{x,y} = \left[ \left( \frac{10 \times (s_{x,y})}{(s_{x,y} + u_{x,y})} \right) \left( \frac{1}{\sqrt{(u_{x,y})}} \right) \right]$ | $\sqrt{(u_{x,y})}$ is to strictly control increase in bad behavior. Trust value approaches 0 rapidly with an increase in the number of bad behavior. |
| Han et al. [16] | $T(u) = Cr(u) + Cr(C_u) + R(u)$ | $Cr(u)$ is a cumulative credential of peer *u*.<br>$Cr(C_u)$ is a credential of the community.<br>$R(u)$ is a recommendation credential for peer *u*. |

## 3. Novel Trust Establishment Method

### 3.1 Assumptions

We assume the following:

- All nodes have unique identities, and authentication methods are used to defend against using a fake ID.
- Nodes are static.
- Nodes can observe activities of other nodes within communication range. For example, a node can overhear its neighbors' transmissions, and in this way, can detect whether the node is forwarding or dropping packets.

- A malicious node misbehaves intelligently; that is, it tries to maintain its trust value in the trusted zone while attacking the network.

### 3.2 Trust estimation method

Trust is calculated based on either observations or recommendations. In order to calculate an observation-based trust for the node, two factors are considered: frequency and weight of the misbehavior. Frequency of misbehavior shows how frequently the node misbehaves during a certain time interval. We use a time-window mechanism to estimate frequency of misbehavior. It is measured based on the number of on and off periods during the $t_k$ period. Time window $t_k$ has several time units. Each time unit $j$ is defined as either an on or an off period based on the rate of misbehavior in time unit $j$ as follows:

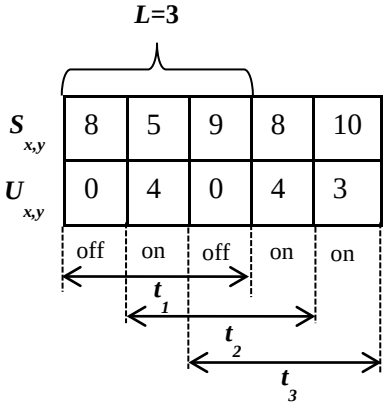
$$j = \begin{cases} \text{on-perod } if \;\; r_j > \theta \\ \text{off-period } otherwise \end{cases}$$

where $r_j$ is the rate of misbehavior in time unit $j$, which is $r_j = \frac{U_j}{S_j + U_j}$, $S_j$ and $U_j$ are good and bad behavior in time unit $j$, respectively, and $\theta$ is the threshold value. $\theta$ is an application-specific or network scenario–specific parameter. The purpose of this parameter is to avoid the effect of the network's condition on trust or on accommodating the application's needs. For example, if trust establishment is applied in routing, and trust is estimated based on the number of forwarded and dropped packets, then dropped packets due to channel condition or collisions should not affect the trust value. Hence, if the rate of misbehavior is greater than a certain threshold $\theta$, then time unit $j$ is defined as an on period. Otherwise, it is considered an off period.

After defining all the time units as either an on or off period within the $t_k$ period, based on the number of on and off periods, the frequency of misbehavior is defined as follows:

$$f_{t_k}^{\,m} = \frac{o_{t_k}}{o_{t_k} + p_{t_k}} \tag{8}$$

where $o_{t_k}$ and $p_{t_k}$ are the number of on and off periods during $t_k$. In order to update the frequency of misbehavior after each $\Delta$ time period, the time window slides to the right, forgetting information in the first time unit and adding information in the last time unit. A sample scenario of time-window usage to estimate frequency of misbehavior is illustrated in **Fig. 2**. According to **Fig. 2**, node $x$ records observations and estimates the frequency of misbehavior for node $y$. For the sake of simplicity, the threshold value is set to $\theta$=0, and the number of time units in the time window ($L$) is 3. As seen in **Fig. 2**, each time unit is defined as either an on or an off period, depending on the misbehavior rate. For example, in the first time period, there is no misbehavior (that is, the misbehavior rate is zero), so it is defined as an off period. Based on the number of on and off periods in each time window, the frequency of misbehavior is estimated for each, $t_1$, $t_2$, and $t_3$.




$t_1 = (o_{t_1} = 1, p_{t_1} = 2), \quad f_{t_1}^m = \frac{1}{3}$

$t_2 = (o_{t_2} = 2, p_{t_2} = 1), \quad f_{t_2}^m = \frac{2}{3}$

$t_3 = (o_{t_3} = 1, p_{t_3} = 2), \quad f_{t_3}^m = \frac{1}{3}$

**Fig. 2.** Misbehavior frequency estimation using a time-window mechanism.

Then, the node's status is determined according to $f_{t_k}^m$ as follows for time window $t_k$:

$$S\left(f_{t_k}^m\right) = \begin{cases} 1 & \text{persistent malicious node} \\ (0;\theta) & \text{legitimate node} \\ (\theta;1) & \text{malicious or on-off attacking node} \end{cases} \tag{9}$$

A node is considered a persistent malicious node if all time units are found to be an on period; that is, $f_{t_k}^m$ =1. On the other hand, if all time units are off periods, $f_{t_k}^m \in (0;\theta)$, then a node is considered legitimate, or a good node. If the value of $f_{t_k}^m$ is between $(1;\theta)$, then the node is malicious or an on-off attacking node.

Next, the weight of the misbehavior is estimated, based on the rate of misbehavior in the last and previous time units as follows:

$$w_{t_k}^m = \begin{cases} \alpha \times r_L + (1-\alpha) \times r_{L-1} & \text{if } r_L > r_{L-1} \\ (1-\alpha) \times r_L + \alpha \times r_{L-1} & \text{if } r_L < r_{L-1} \\ r_L \end{cases} \tag{10}$$

where $r_L$ and $r_{L-1}$ are rates of misbehavior in the last and previous time units, and $L$ is the number of time units in the time window. The rate of misbehavior for time unit $j$ is estimated as $r_j = \frac{U_j}{S_j + U_j}$, and $S_j$ and $U_j$ are the number of good and bad behavior, respectively, in time unit $j$. The variable $\alpha$, within the range (0.5, 1), represents the forgetting factor. It assigns more weight to the greater measured misbehavior, so bad behavior is remembered longer. The objective of such a method is to defend against an on-off attack because, in an on-off attack, the malicious node alternates its behavior from bad to good (or vice versa) so it is not detected. Hence, when a malicious node intends to misbehave in one time unit and to show good behavior in another time unit, the proposed method assists in mitigating the effect of the on-off attack.

Measured frequency and weight of misbehavior are combined to obtain trust value $T_{t_k}$ for time window $t_k$ as follows:

$$T_{t_k} = \frac{\beta \times \left(1 - f_{t_k}^{m}\right) + \left(1 - \beta\right) \times \left(1 - w_{t_k}^{m}\right)}{1 + a_{fw}} \tag{11}$$

where $\beta$ is a parameter to give a weight to the frequency and weight of misbehavior, which varies within the interval [0.5;1]. Depending on the application or performance requirements, different $\beta$ values are assigned to each factor. For example, if the frequency of misbehavior is more important than the weight of misbehavior (instantaneous misbehavior) for some applications, then more weight is given to frequency of misbehavior. Hence, our scheme provides room for adaptability. $a_{fw}$ is the average of the measured frequency and weight of misbehavior, which is defined as $a_{fw} = \frac{f_{t_k}^{m} + w_{t_k}^{m}}{2}$. The objective of this equation is to emphasize both factors equally. Since the numerator of trust-estimation equation (9) allows control of the weight given to each factor, the denominator ensures that the effect of each factor is kept equal. After each Δ period, each node estimates three components: frequency and weight of misbehavior, and trust. Moreover, after each Δ period, the number of on and off periods is updated using the sliding time window.

## 4. Performance Evaluation

In this section, we evaluate and compare our proposed trust mechanism against earlier proposed schemes. The proposed scheme is evaluated and compared in terms of persistent malicious behavior and on-off attack detection. We compared our scheme with GTMS [6] and Retrust [9]. The former is one of the earliest comprehensive trust schemes for WSNs. The latter is one of the more recent comprehensive trust schemes.

In all evaluations, values for system parameters such as trust threshold, forgetting factor, and number of time units in the time window were selected based on heuristics and values previously defined in the literature. For example, trust threshold is set to about half of the maximum trust value in the literature [6-7, 17-22]. Hence, in these references, the defined trust threshold is between 0.4 and 0.8. Yu et al. [17] suggested that the most intuitive trust threshold is 0.5 when the maximum trust value is 1. The optimal trust threshold according to a scenario defined by Bao et al. [22] is 0.6. The choice of value for the forgetting factor remains largely heuristic and depends on the strategy of trust establishment [17]. Since the forgetting factor is used mainly to combat an on-off attack, authors have used different values and different mechanisms to derive the value of the forgetting factor according to their own trust estimations and considerations [5-6, 18, 20]. Following the guidelines and suggestions from Sun et al. [5], we intuitively used a forgetting factor of 0.6.

### 4.1 Impact of $\beta$ parameter on misbehavior detection

In this section, we demonstrate the impact of the $\beta$ parameter on persistent and frequent misbehavior detection. Specifically, **Fig. 3** shows estimated trust values with different $\beta$ values under persistent misbehavior; that is, $f_{t_k}^{m}$ =1. As we can see, when $\beta$ is between 0.5 and 0.7, all estimated trust values fall under the trust threshold, s=0.6, when the weight of misbehavior is between [0.1; 1]. It means that when $\beta$ is between 0.5 and 0.7, the trust estimation method can detect almost all kinds of persistent malicious behavior.

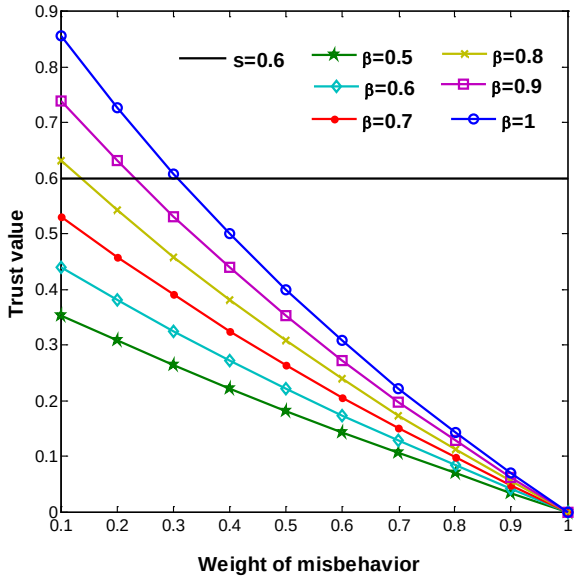



**Fig. 3.** Impact of $\beta$ on detecting persistent malicious behavior ( $f_{t_k}^m$ =1) with different misbehavior weights.

**Fig. 4** demonstrates estimated trust values under a fixed misbehavior weight; that is, $w_{t_k}^m$ =0.1, with different misbehavior frequencies. The reason for setting misbehavior weight at 0.1 is that a weight smaller than 0.1 can be negligible, and for values bigger than 0.1, detection will be obvious. As **Fig. 4** shows, when the $\beta$ value is between 0.5 and 0.7 and the frequency of misbehavior is between 0.5 and 1, the estimated trust values fall under the trust threshold. From **Fig. 3** and **Fig. 4**, we conclude that the optimal $\beta$ value is between 0.5 and 0.7, because the misbehavior detection rate is the highest in that case.

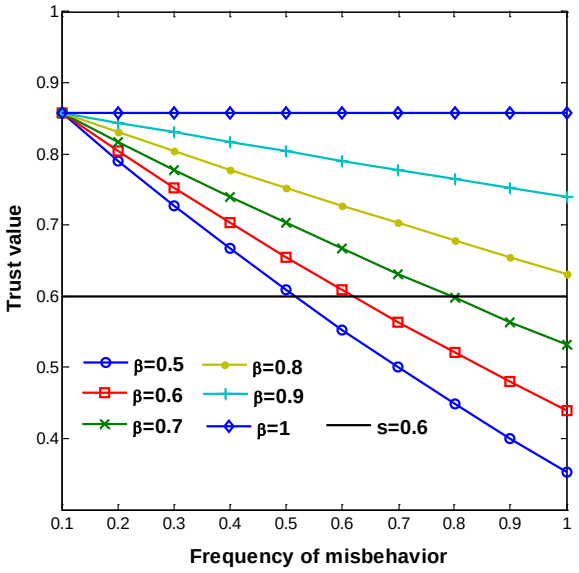



**Fig. 4.** Impact of $\beta$ on detecting frequent misbehavior with a fixed misbehavior weight ( $w_{t_k}^m$ =0.1).

### 4.2 Persistent misbehavior detection

The misbehavior rate of a node is measured based on the proportion of the number of bad behavior to the total number of behaviors, $r = \frac{U}{S+U}$, where $U$ is the number of bad behaviors and $S$ is the number of good behaviors. Thus, persistent malicious behavior means that measured misbehavior rate is always greater than the predefined threshold. The goal of the threshold is to avoid the effect of other factors on the estimated trust value. For example, if trust establishment is applied in routing, and trust is estimated based on the number of forwarded and dropped packets, then dropped packets due to channel condition or collisions should not be considered bad behavior. To emulate persistent malicious behavior and to demonstrate its detection, the parameters in **Table 2** are used.

**Table 2.** Parameters to emulate persistent misbehavior.

| Parameter | Value |
|---|---|
| Rate of misbehavior for each time unit | Fixed from 0.2 to 0.3 |
| | Random between 0.1 and 0.3 |
| Number of time units | $L$=3<br>L= 10 (for the proposed trust scheme) |
| Trust and misbehavior frequency and weight estimation period | Δ=1 |
| Trust threshold | s=0.6 |
| Experiment time | 50Δ |
| Beta value | $\beta$=0.7 |
| Forgetting factor | $\alpha$=0.7 (for all trust schemes) |
| Threshold for rate of misbehavior | $\theta$=0 |

For each time unit, the numbers of misbehavior and good behavior are generated in a random or fixed manner, and trust was estimated based on the numbers generated for misbehavior and good behavior. We compared our trust estimation mechanism with GTMS [6] and ReTrust [9]. Fig. 5 shows estimated trust values over time under persistent malicious behavior. For each time unit, the numbers of good and bad behavior are generated randomly at between 5 and 10 and between 1 and 3, respectively. As we can see from Fig. 5, the trust value abruptly falls below the trust threshold at time 10 because, at the very beginning, misbehavior frequency ($f_{t_k}^m$) is not included in the trust estimation because the length of the sliding time window equals 10Δ. Hence, trust values go down suddenly at time 10 when misbehavior-frequency information is added to estimate the trust value. Trust values fluctuate because of the different weights for misbehavior in each time period. Dynamicity of the trust values shows that our trust scheme also efficiently considers current status of the node. As Fig. 5 demonstrates, trust values in other schemes remain above the trust threshold, even though a node persistently misbehaves.

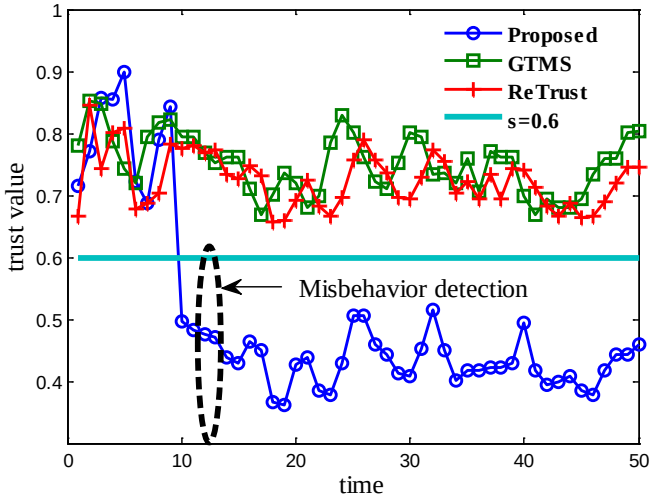



**Fig. 5.** Persistent malicious behavior detection when misbehavior is random.

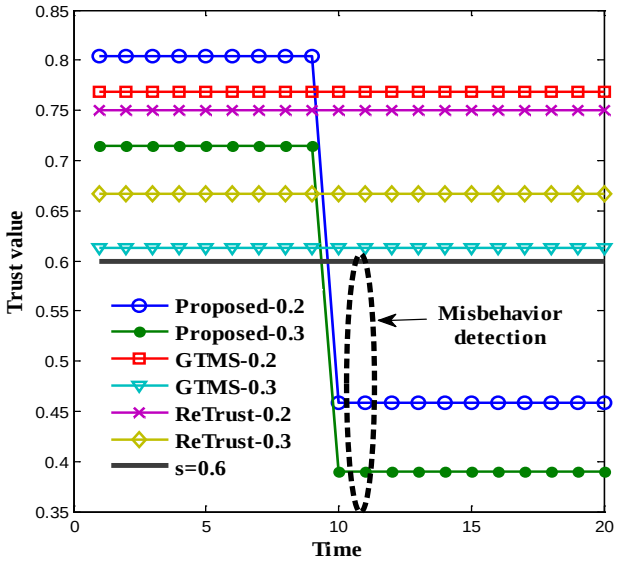



**Fig. 6.** Persistent malicious behavior detection when misbehavior is constant.

**Fig. 6** shows persistent malicious behavior detection under fixed, measured misbehavior. Thus, the rate of misbehavior in each $\Delta$ period is fixed from 0.2 to 0.3. In **Fig. 6**, Proposed-0.2 means the performance of the proposed scheme under a fixed misbehavior rate of 0.2. Hence, the numbers of good and bad behaviors are generated as 8 and 2, respectively, for each time unit in the time window. As we can also see from **Fig. 6**, once misbehavior frequency information is added to trust estimation at time 10, trust values fall below the trust threshold and remain steady. As **Fig. 6** demonstrates, trust values remain constant over time as the numbers of good and bad behaviors remain constant in each time unit.

### 4.3 On-off attack resilience

In this section, we evaluate and compare our trust scheme under different on-off attack strategies. In an on-off attack, the goal of a malicious node is to remain undetected while attacking. Specifically, a malicious node attempts to ensure its trust value remains within the trustable level while attacking or misbehaving. Thus, sometimes it behaves well to increase the trust value and sometimes it attacks. Hence, an on-off attack consists of two periods: on and off. In an on period, it misbehaves, or attacks, and during the off period, it either stops doing anything or only behaves well.

To emulate the behavior of an on-off attack node and to evaluate the proposed trust scheme under an on-off attack, we used the parameters in **Table 3**.

**Table 3.** Parameters to emulate an on-off attack.

| **Parameter** | **Value** | |
|---|---|---|
| Probability of an on period | 0.6, 0.4, and 0.2 | |
| Probability of an off period | 0.4, 0.6 and 0.8 | |
| Good behavior | On period: | Randomly generated between: |
| | 0.6 | 8 to 10 |
| | 0.4 | 8 to 10 |
| | 0.2 | 8 to 10 |
| | Off period: | Randomly generated between: |
| | 0.4 | 8 to 10 |
| | 0.6 | 8 to 10 |
| | 0.8 | 8 to 10 |
| Bad behavior | On period | Randomly generated between: |
| | 0.6 | 4 to 6 |
| | 0.4 | 6 to 9 |
| | 0.2 | 12 to 18 |
| | Off period | In all cases, zero |
| Number of time units | $L$= 3 (for other trust schemes); L= 10 (for the proposed trust scheme) | |
| Trust and misbehavior frequency and weight estimation period | $\Delta$=1 | |
| Trust threshold | $s$=0.6 | |
| Experiment time | 100$\Delta$ | |
| Weight parameter | $\beta$=0.7 | |
| Forgetting factor | $\alpha$=0.7 | |
| Threshold for rate of misbehavior | $\theta$=0 | |

To make the emulation more realistic and fair, we use three different strategies for on-off attacks. In the first strategy, a malicious node intends to attack more but decreases the weight of the misbehavior so it is not detected while attacking. In other words, frequency of the misbehavior increases but the weight of the misbehavior decreases. So, in this strategy the probability of an on period is set to 0.6, and the good and bad behavior incidents number between 8 and 10 and 4 and 6, respectively, during on periods. Moreover, on and off periods

are randomly distributed over time. In the second strategy, a malicious node intends to attack fewer times, but it increases the weight of the misbehavior during an on period. So, in this strategy, the probability of an on period is set to 0.4, and the numbers of good and bad behaviors are generated between 8 and 10 and 6 and 9, respectively, during an on period. Finally, in the third strategy, a malicious node increases the weight of the misbehavior during an on period, but decreases the number of on periods. Hence, the probability of an on period is set to 0.2, and the numbers of good and bad behaviors are generated between 8 and 10 and 12 and 18, respectively, during an on period. In all three strategies, the number of good behavior is generated randomly at between 8 and 10, and the number of bad behavior is always 0 during an off period. The trust value is estimated after each time unit, and if an estimated trust value falls below the trust threshold, the node is considered untrustworthy for that period. To find the average detection rate of the attack, the sum of the number of untrustworthy time is divided by the total experiment time. As **Fig. 7** shows, the detection rate is the highest in our proposed scheme, because frequency of the misbehavior is the highest among the three strategies.

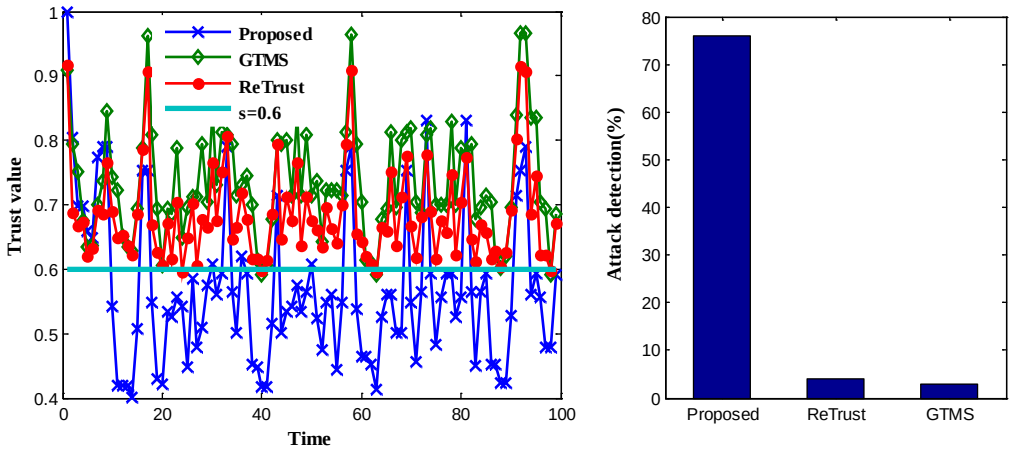



**Fig. 7.** On-off attack detection (probability of an on period is 0.6).

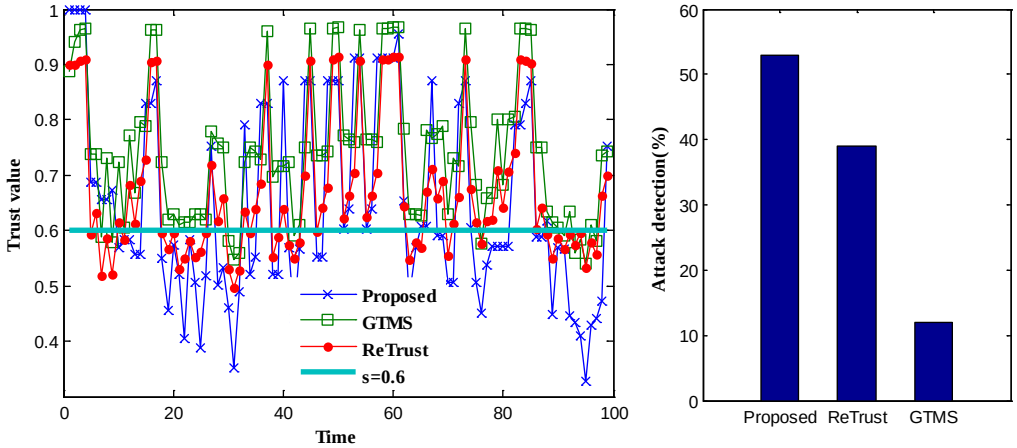



**Fig. 8.** On-off attack detection (probability of an on period is 0.4).

In **Fig. 8**, when the frequency of misbehavior decreases to 0.4, our scheme outperforms the other two schemes. However, **Fig. 9** shows that ReTrust outperforms the others when the number of bad behaviors is the highest and the number of on periods is the least among the

three strategies. An important observation from these three types of evaluation is that, even though the total number of bad behavior in the three strategies is similar (that is, they are about 300), detection rates differ much more in the traditional schemes. For example, in **Fig. 8** the detection rate is very low, whereas in **Fig. 9**, it is medium under the other two schemes. With our scheme, the detection rate remains between 76 and 31.

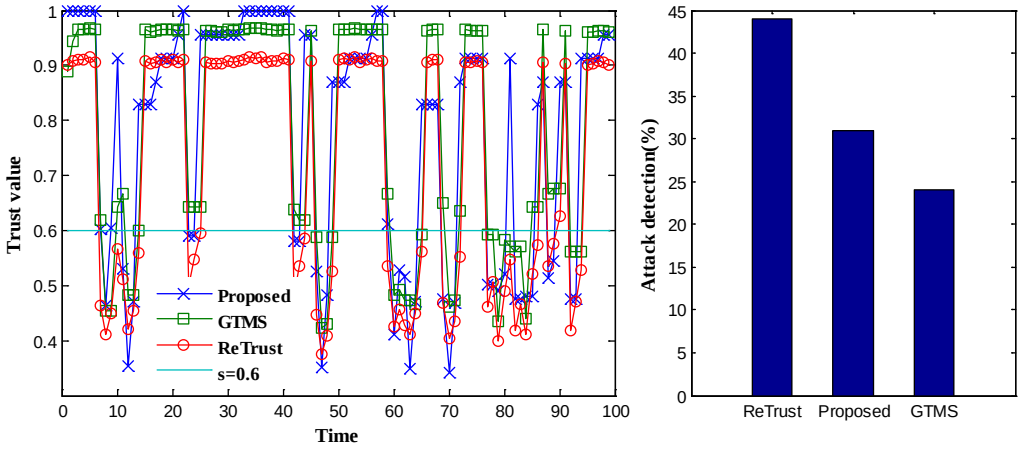



**Fig. 9.** On-off attack detection (probability of an on period is 0.2).

## 5. Conclusion

In this paper, we propose a novel trust establishment scheme. Unlike traditional trust schemes, the proposed trust scheme considers frequency of misbehavior to estimate trust values, which allows efficient tracking of node behavior. To the best of our knowledge, this is the first trust establishment scheme that considers misbehavior frequency in trust estimation. It can explicitly demonstrate how frequently a node misbehaves within an observed period. Such a property is important in wireless sensor networks, because sensor nodes are often stuck malfunctioning, which can be efficiently detected by the proposed trust scheme. Evaluation results demonstrate that the proposed scheme can detect all kinds of persistent misbehavior. Moreover, under different on-off attack strategies, the proposed scheme demonstrates stable and higher detection rates compared to other trust mechanisms.

## Acknowledgement

The present research has been conducted by the Research Grant of Kwangwoon University in 2015. Moreover, it was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2012R1A1B4000536) .

## Appendix

**Table 1.** Definitions of notations.

| Notation | Definition |
|---|---|
| $t_k$ | Time window $k$ |
| $L$ | Number of units in the time window |
| $S_{x,y}$ | Number of good behavior in node $y$ found by node $x$ |
| $U_{x,y}$ | Number of bad behavior in node $y$ found by node $x$ |
| $\beta$ | Weight parameter given to frequency and weight of misbehavior |
| $T_{t_k}$ | Trust value in time window $t_k$ |
| $\alpha$ | Forgetting factor |
| $f_{t_k}^m$ | Weight of misbehavior for time window $t_k$ |
| $w_{t_k}^m$ | Frequency of misbehavior for time window $t_k$ |
| $r_j$ | Rate of misbehavior for time unit $j$ |
| $\Delta$ | Duration of one time unit |
| s | Trust threshold |
| $\theta$ | Threshold for rate of misbehavior |

## References


[1] Falcone, Rino, Giovanni Pezzulo, and Cristiano Castelfranchi, "A fuzzy approach to a belief-based trust computation," *Trust, reputation, and security: theories and practice*, pp. 73-86, 2003. Article (CrossRef Link)

[2] Kim, Tae Kyung, and Hee Suk Seo, "A trust model using fuzzy logic in wireless sensor network," *World academy of science, engineering and technology*, vol. 42, pp. 63-66, 2008.

[3] Manoj, V et al. "A Novel Security Framework Using Trust and Fuzzy Logic in MANET," *International Journal of Distributed and Parallel Systems*, vol. 3, 2012. Article (CrossRef Link)

[4] Mármol, Félix Gómez, and Gregorio Martínez Pérez, "Providing trust in wireless sensor networks using a bio-inspired technique," *Telecommunication systems*, vol. 46, no. 2, pp. 163-180, 2011. Article (CrossRef Link)

[5] Sun Yan Lindsay, Zhu Han, and KJ Ray Liu, "Defense of trust management vulnerabilities in distributed networks," *Communications Magazine, IEEE*, vol. 46, no. 2, pp. 112-119, 2008. Article (CrossRef Link)

[6] Shaikh, Riaz Ahmed et al, "Group-based trust management scheme for clustered wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions*, vol. 20, no. 11, pp. 1698-1712, 2009. Article (CrossRef Link)

[7] Govindan, Kannan, and Prasant Mohapatra, "Trust computations and trust dynamics in mobile adhoc networks: a survey," *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 2, pp. 279-298, 2012. Article (CrossRef Link)

[8] Ganeriwal, Saurabh, Laura K Balzano, and Mani B Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, pp. 15, 2008. Article (CrossRef Link)

[9] He, Daojing et al, "ReTrust: Attack-resistant and lightweight trust management for medical sensor networks," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no.4, pp. 623-632, 2012. Article (CrossRef Link)

[10] Akyildiz, Ian F et al, "A survey on sensor networks," *Communications magazine, IEEE*, vol. 40, no. 8, pp. 102-114, 2002. Article (CrossRef Link)

[11] Ishmanov Farruh, Sung Won Kim, and Seung Yeob Nam, "A Secure Trust Establishment Scheme for Wireless Sensor Networks," *Sensors*, vol. 14, no. 1, pp.1877-1897, 2014. Article (CrossRef Link)

[12] Zhan, Guoxing, Weisong Shi, and Julia Deng, "Design and implementation of TARF: a trust-aware routing framework for WSNs," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, no. 2, pp. 184-197, 2012. Article (CrossRef Link)

[13] Ishmanov Farruh et al, "Trust management system in wireless sensor networks: Design considerations and research challenges," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 2, pp. 107-130, 2015. Article (CrossRef Link)

[14] Li, Xiaoyong, Feng Zhou, and Junping Du, "LDTS: A Lightweight and Dependable Trust System for Clustered Wireless Sensor Networks," *IEEE transactions on information forensics and security*, vol. 8, no. 6, pp. 924-935, 2013. Article (CrossRef Link)

[15] Velloso, Pedro B et al, "Trust management in mobile ad hoc networks using a scalable maturity-based model," *IEEE Transactions on Network and Service Management*, vol. 7, no.3, pp. 172-185, 2010. Article (CrossRef Link)

[16] Qiyi Han et al. "A topological potential weighted community-based recommendation trust model for P2P networks," *Peer-to-Peer Networking and Applications.* DOI 10.1007/s12083-014-0288-9. Article (CrossRef Link)

[17] Yu, Han et al, "A survey of trust and reputation management systems in wireless communications," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1755-1772, 2010. Article (CrossRef Link)

[18] Jøsang, Audun, Roslan Ismail, and Colin Boyd, "A survey of trust and reputation systems for online service provision," *Decision support systems*, vol. 43, no.2, pp. 618-644, 2007. Article (CrossRef Link)

[19] He Daojing et al, "A distributed trust evaluation model and its application scenarios for medical sensor networks," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no. 6, pp. 1164-1175, 2012. Article (CrossRef Link).

[20] Chae, Younghun, Lisa Cingiser DiPippo, and Yan Lindsay Sun, "Predictability trust for Wireless Sensor Networks to provide a defense against On/off attack," *Collaborate Com*, pp. 406-415, 2012. Article (CrossRef Link).

[21] Fung, Carol J et al, "Design of a simulation framework to evaluate trust models for collaborative intrusion detection," *Network and Service Security, 2009. N2S'09. International Conference*, pp. 1-5, 2009.

[22] Bao, Fenye et al, "Trust-based intrusion detection in wireless sensor networks," *Communications (ICC), 2011 IEEE International Conference*, pp. 1-6, 2011. Article (CrossRef Link)

**Farruh Ishmanov** received his BS in Information Systems in 2007 from Tashkent State University of Economics, Uzbekistan. At this university, he studied and worked in the Multimedia Lab during his undergraduate years. He received his MS and PhD from the Department of Information and Communication Engineering, Yeungnam University, Korea, in 2009 and 2014, respectively. He was awarded a Korean Government IITA Scholarship for his MS. He is currently Asistant Professor in the Department of Electornics and Communication Engineering, Kwangwoon University. His research interests include resource management and security in wireless sensor networks.

**Sung Won Kim** received his BS and MS from the Department of Control and Instrumentation Engineering, Seoul National University, Korea, in 1990 and 1992, respectively, and his PhD from the School of Electrical Engineering and Computer Sciences, Seoul National University, Korea, in August 2002. From January 1992 to August 2001, he was a Researcher at the Research and Development Center of LG Electronics, Korea. From August 2001 to August 2003, he was a Researcher at the Research and Development Center of AL Tech, Korea. From August 2003 to February 2005, he was a Postdoctoral Researcher in the Department of Electrical and Computer Engineering, University of Florida, Gainesville, USA. In March 2005, he joined the Department of Information and Communication Engineering, Yeungnam University, Gyeongsangbuk-do, Korea, where he is currently a Full Professor. His research interests include resource management, wireless networks, mobile networks, performance evaluation, and embedded systems.