2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing

A Dynamic Trust Establishment and Management Framework for Wireless Sensor Networks

Junqi Zhang[1,2], Rajan Shankaran[1], Mehmet A. Orgun[1], Vijay Varadharajan[1] and Abdul Sattar[2]

[1]Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
E-mail: {janson,mehmet,rshankar,vijay}@science.mq.edu.au

[2]School of Information and Communication Technology, Griffith University, Brisbane, Queensland, Australia
E-mail: A.Sattar@griffith.edu.au

***Abstract*—In this paper, we present a trust establishment and management framework for hierarchical wireless sensor networks. The wireless sensor network architecture we consider consists of a collection of sensor nodes, cluster heads and a base station arranged hierarchically. The framework encompasses schemes for establishing and managing trust between these different entities. We demonstrate that the proposed framework helps to minimize the memory, computation and communication overheads involved in trust management in wireless sensor networks. Our framework takes into account direct and indirect (group) trust in trust evaluation as well as the energy associated with sensor nodes in service selection. It also considers the dynamic aspect of trust by introducing a trust varying function which could be adjusted to give greater weight to the most recently obtained trust values in the trust calculation. The architecture also has the ability to deal with the inter-cluster movement of sensor nodes using a combination of certificate based trust and behaviour based trust.**

I. Introduction

Wireless sensor networks help to accurately gather information, monitor and react to events from the physical world. A sensor node consists of sensor(s), wireless communication device, small microcontroller and energy source. WSNs have certain unique characteristics at both the sensor node level and the sensor network level. At the sensor node level, each sensor node has constraints on resources such as energy, memory, computation speed and bandwidth. At the sensor network level, WSNs may inherit the infrastructureless nature of the wireless ad hoc networks, and they can have dynamic network topology and membership, without the support of a management authority. In addition, WSNs may be deployed in large scale and can be mobile but can suffer from the lack of physical protection, as well as general failures that relate at the node and communication level.

WSNs have many applications such as surveillance of infrastructure, habitat monitoring, health care and traffic control. Many applications of the WSNs require secure communications and quality of service [1]. But in practice, wireless sensor networks are prone to different types of malicious attacks, such as denial of service, routing protocol attacks as well as replay attacks, sybil attacks, traffic analysis and physical attacks on nodes. Traditional cryptoschemes may not prevent such types of malicious attacks. Moreover, traditional trust management schemes developed for wired and wireless networks may not be suitable for networks with small sensor nodes due to limited bandwidth and stringent node constraints in terms of power and memory. Therefore, it is important to develop trust management schemes and protocols that take into account the intrinsic features of wireless sensor networks mentioned above. There are several proposals for trust management for WSNs [2], [3], [4], [5], [6], [7]. None of those proposals consider the requirements of trust management for WSNs all at once, such as memory constraints, computation and communication overheads and energy levels of individual sensor nodes.

Recently, Shaikh *et al* [8] have proposed a group-based trust management scheme (GTMS) for clustered wireless sensor networks. It is aimed to detect and prevent selfish, faulty and malicious nodes. However, it does not take into account the dynamic aspects of trust and predeployment knowledge of sensor nodes. Moreover, GTMS has significant communication overhead in terms of calculations needed to determine a node's trust value. We have recently proposed a trust management architecture for hierarchical WSNs in order to remedy some of the shortcomings of GTMS [9]. This paper builds on the new trust management framework for WSNs proposed in [9], aiming to improve the trust evaluation process by taking into account the dynamic aspects of trust.

Our new trust management framework makes use of the hierarchical wireless sensor network architecture to minimize the memory overhead by using the cluster head in the management of trust information. Our approach also reduces the communication overhead by making the nodes only communicate with the cluster head. We propose a novel trust value calculation scheme with a decaying technique, so that recent trust values could be given more (or less) weight in the overall trust calculation, thereby taking into account the dynamic nature of trust. The bad behavior of a node will reduce its trust value greatly. In addition, the weighting is parameterized to make it flexible enough to suit various applications. We also take into account the energy level of sensor nodes to avoid the short life time of highly trustworthy nodes. Moreover, we envisage that the nodes may move from one cluster to another, while maintaining their trust records. Furthermore, we strengthen the cluster head security. Finally, we combine the behaviour based trust and certificate based trust using pre-deployment

978-0-7695-4322-2/10 $26.00 © 2010 IEEE
DOI 10.1109/EUC.2010.80

knowledge in the establishment of trust relationships.

This paper is organized as follows. Section 2 briefly summarizes the GTMS trust scheme as well as discussing the related work in this area. In Section 3, we propose our new trust scheme for hierarchical ad hoc wireless sensor networks. Section 4 provides a comparison of our proposed trust scheme with the existing trust management schemes. Finally, Section 5 concludes the paper with a brief summary.

II. Trust Management for WSNs

The notion of trust has been studied extensively in various disciplines. In the area of secure computing, different aspects of trust have been discussed in different contexts such as trusted processes, trusted platforms and trusted computing, trusted code, trust management and trust negotiation. In mobile ad hoc networks (MANETs) and WSNs, many trust models determine the trust values associated with the nodes using a continuous monitoring of the nodes' behavior. A node's misbehavior can be divided into two types: (1) the selfish or greedy behavior and (2) malicious behavior [10]. A selfish node usually wants to save its own resources such as power, CPU cycles, and memory. The selfish behavior itself can be divided into two types: the self exclusion and non-forwarding. Such behaviour can occur in two stages of the network deployment, namely the routing phase and packet transfer phase. The malicious behaviour associated with non-forwarding include packet dropping, packet modification, packet fabrication, timing attacks, and silent route change. During the route discovery phase, misbehavior includes attacks such as black hole attacks, gray hole attacks, and worm hole attacks.

There have been many recent proposals for trust management for ad hoc networks in the literature, such as [11], [12], [13], [14]. According to their scope, purpose and type of evidence the trust value is based on, they can be categorized into two groups: the certificate-based framework and the behavior-based framework [15]. The certificated based frameworks usually use certificates as the pre-deployment knowledge to establish a trust relationship. A valid certificate can be used to prove that the target node is trusted by a Certification Authority (CA) or trusted third parties or other nodes. With a hierarchical structure, the certificates are signed by trusted parties and arranged in a hierarchy, and the trustor can verify the signature with the public keys in the associated trusted path. This approach is usually used to authenticate the target nodes and determine whether they are legimate members of the network. In behavior based frameworks, each node continually monitors the behavior of target nodes and builds the trust value based on how cooperative they are. This can then be used to determine whether a node is selfish or malicious.

In general, the trust management schemes for mobile ad hoc networks make certain assumptions with respect to the capabilities of individual nodes that are not realistic in WSN environment. Therefore several trust management schemes have also been proposed for wireless sensor networks, taking into consideration the inherent characteristics of WSNs. Broadly, we can classify these schemes into three categories: sensor node-based, super-sensor node-based and base station-based trust management schemes. In our framework, we use the super node based trust management approach.

a) Node based schemes: Because of the limited resources, the nodes in WSNs can only monitor, store and use little trust information of their neighbouring nodes. Several trust management schemes belong to this category [16], [7].

b) Super nodes based schemes: In order to establish a trust relationship in large scale sensor networks, several super nodes based schemes have been proposed [2], [3], [4], [5], [6], [8]. In such schemes, some nodes referred to as super nodes are assumed have more computation power, storage, and power for communication.

c) Base station based schemes: In this type of schemes, the trust value calculations are managed by base stations. Base station based schemes assume that all the nodes in WSNs have the same power and resources. An example of such a scheme is presented in [17].

A. Group-based Trust Management Scheme

A type of scheme that use super nodes is called a group-based trust management scheme (GTMS) proposed in [8] for clustered (hierarchical) WSNs. As our scheme is also based on a similar wireless sensor network architecture, we first describe this scheme in some detail.

Hierarchical (clustered) wireless sensor network model was proposed in [18], [19] and has been subsequently used in other works such as [20], [21], [22]. In this model, a wireless sensor network consists of a command node (base station), cluster heads and numerous sensor nodes grouped into clusters [23], see Figure 1. The clusters of sensors can be formed based on various criteria such as capabilities, location and communication range, and using different cluster algorithms such as in [24], [25]. It is assumed that all sensors and cluster head nodes are stationary and the physical location and communication range of all nodes in the network are known.

The scheme makes the following assumptions: All the sensor nodes have unique identities and are organized into clusters. The base station is a central command authority and virtually has no resource constraints and is fully trusted. Clus-

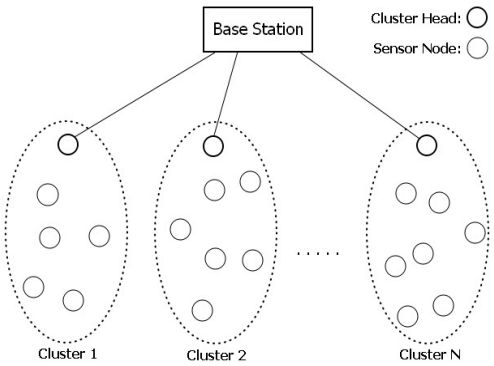

Fig. 1. Hierarchical WSN Architecture

ter heads have more computational power and more memory compared to other sensor nodes in the network. The base station communicates with the cluster head and each cluster head manages the nodes in its group.

The trust model works with two topologies: the intra-group topology that uses distributed trust management and inter-group topology that deploys centralized trust management. For the intra-group network, each sensor calculates the individual trust values for all of its group members. Based on these trust values, a node assigns one of the three possible states to the other member nodes, namely *trusted*, *untrusted*, or *uncertain*. Each node then forwards the trust states of all the group member nodes to the cluster head. The cluster head then collates them together in a report and forwards a copy of it to the base station. The cluster head also sends the trust value that it has assigned to its peer cluster heads to the base station. The scheme calculates trust at three levels: the node level, the cluster head level and the base station level.

1) Node level Trust: The past interaction evaluation of the node level trust is based on a time window. The time window consists of several time units, and the interactions that occur in each time unit are recorded. After a unit of time elapses, the window slides one time unit to the right by dropping the interactions done during the earliest unit. Thus, as time progresses, the window forgets the experiences of previous units but adds the experiences of the newest time unit.

With this time window information, the time-based past interaction trust value (T_{xy}) of node y at node x that lies between 0 and 100 is defined as

$$T_{xy} = \left[\frac{100(S_{xy})^2}{(S_{xy} + U_{xy})(S_{xy} + 1)} \right] \quad (1)$$

where [.] is the nearest integer function, S_{xy} is the total number of successful interactions of node x with y during time units, and U_{xy} is the total number of unsuccessful interactions of node x with y during time units.

After calculating the trust value, a node will quantify the trust into the three states as follows:

$$M_p(T_{xy}) = \left\{ \begin{array}{ll} \textit{trusted} & 100 - f \leq T_{xy} \leq 100 \\ \textit{uncertain} & 50 - g \leq T_{xy} \leq 100 - f \\ \textit{untrusted} & 0 \leq T_{xy} \leq 50 - g \end{array} \right\} \quad (2)$$

where f represents half of the average values of all trusted nodes, and g represents one third of the average values of all untrusted nodes. The values of f and g are calculated as follows:

$$f_{j+1} = \left\{ \begin{array}{cc} \left[\frac{1}{2} \left(\frac{\sum_{r<R_x} T_{xy}}{|R_r|} \right) \right] & 0 \leq |R_r| \leq n-1 \\ f_j & |R_x| = 0 \end{array} \right\} \quad (3)$$

$$g_{j+1} = \left\{ \begin{array}{cc} \left[\frac{1}{2} \left(\frac{\sum_{r<M_x} T_{xy}}{|M_r|} \right) \right] & 0 \leq |M_r| \leq n-1 \\ g_j & |M_x| = 0 \end{array} \right\} \quad (4)$$

where [.] is the nearest integer function, R_x represents the set of trustwothy nodes for node x, M_x represents the set of untrustful nodes for node x, and n is the total number of nodes (a collection of trusted, untrusted, and uncertain nodes).

Whenever a node requires peer recommendation, it will send a request to all member nodes exclusing the untrusted ones. Let us assume that j nodes are trusted or uncertain in a group. Then, node x calculates the trust value of node y as follows:

$$T_{xy} = \left[\frac{\sum_{r<R_r \cup C_r} T_{xi} * T_{iy}}{100 * j} \right]; j = |R_r \cup C_r| \leq n-2 \quad (5)$$

where [.] is the nearest integer function, T_{x_i} is the trust value of the recommender, and $T_{iy}$is the trust value of node y sent by node i. Here, T_{x_i} is acting as a weighted value of the recommender that is multiplied with the trust value T_{iy}, sent by the recommender, such that the trust value of node y should not increase beyond the trust value between node x and the recommender node i.

2) Cluster Head level Trust: At group level, the trust value is calculated by the cluster head using the trust states that other members in the group have established for a target node. Suppose that there are $n+1$ nodes in the group including the cluster head (*ch*). The cluster head will periodically broadcast the packet within the group requesting the trust state. In response, all member nodes in the group forward their trust states, s, of other member nodes to the cluster head. The variable, s, can take three possible values: *trusted*, *uncertain*, and *untrusted*. The cluster head will maintain these trust states in a matrix form, as shown below:

$$TM_{ch} = \left[\begin{array}{cccc} S_{ch,1} & S_{1,ch} & ... & S_{n,1} \\ S_{ch,2} & S_{1,2} & ... & S_{n,2} \\ \vdots & \vdots & \vdots & \vdots \\ S_{ch,n} & S_{1,n} & ... & S_{n,n-1} \end{array} \right]$$

where TM_{ch} represents the trust state matrix of cluster head ch, and $s_{ch,1}$ represents the state of node 1 at cluster head ch. The cluster head assigns a group level trust state to a node based on the relative difference in trust states received from all the members for that node.

III. Dynamic Trust Framework for WSNs

Our dynamic trust framework builds on the hierarchical architecture of WSNs to minimize the nodes' memory by storing trust information in the cluster head [9]. We employ a decaying trust function which can be used to give more weight to the most recent trust value in the overall trust value computation. We combine behaviour based trust and certificate based trust using the pre-deployment knowledge in the establishment of trust relationships. We also allow the nodes to move from one cluster to another by preserving their trust record, thereby making the scheme suitable for dynamic environments wherein the nodes move frequently.

A. Trust Management Architecture

Each cluster includes the cluster head (or the cluster leader) and a set of distinct sensors. Each sensor has two main functions: sensing and relaying. Sensors probe their environment and gather data. Then they transmit the collected information to the cluster head directly in one hop or by relaying via a

multi hop path. Sensors transmit or relay data only via short-haul radio communication. A cluster head is in charge of its cluster. It is assumed that each cluster head can reach and control all the sensors in the cluster. Each cluster head receives the information from different sensors, and then processes the data to extract relevant information, sends it to the base station (command node) via long-haul transmission.

Our framework introduces the notion of a sponsor node as shown in Figure 2. A sponsor node is the initiator of each cooperation, denoted as s. Any node can be a sponsor node based on the application. This sponsor node will find one or more other nodes to cooperate together. A target node is the node chosen by the sponsor node to cooperate for a service, denoted as t. There can be one or more target nodes for each service. Again, in our framework, we assume that the cluster head has higher computation power and memory when compared to other sensor nodes. The base station (or command node) is assumed to be totally trusted and virtually has no computational or memory constraints.

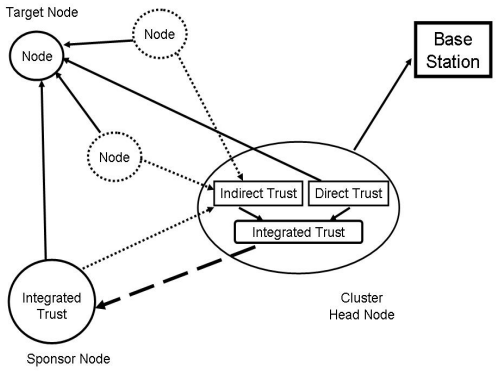

Fig. 2. Trust in WSN Architecture [9]

As shown in Figure 2, each node stores all the direct trust values of all other nodes in the same cluster. Each node then sends all the direct trust values of other nodes to the cluster head. Using the direct trust values, the cluster head calculates and stores the indirect (group) trust information for each node in the cluster. In addition, the cluster head also stores its own direct trust value for each node as well as the power (energy) level of each node residing in its cluster. The cluster head then calculates the integrated trust value for each node in the cluster based on the group and cluster head's direct trust values.

To realize an application service, a sponsor node will find one or more target nodes to cooperate together. First the sponsor node will check if it has the direct trust value of the target candidate nodes. If it has the direct trust values, it will use these values as the trust values. If not, the sponsor node will try to obtain the target node's trust value from the cluster head. After it obtains all the trust values for all target nodes, it can select the most suitable target nodes from this set based on the trust value of the node and its energy level as well as application service requirements.

In wireless sensor networks, a node can provide many application services such as furnishing observed temperature and pressure values and to accomplish this, a node, in turn could request network related services such as forwarding of packets from other peer nodes. We denote a node using the following tuple $\langle ID, A, V, T \rangle$, where *ID* denotes the identity of the node, A denotes the attribute set of node *ID*, V denotes the value set associated with the attributes, and T denotes the trust value set of the attributes. The quality of the service can be measured by the attributes (Figure 3). The better the service, the higher the trust value for each node.

To be deemed as trustworthy, a node must make a reasonable effort to perform its network related services in a dependable manner. The network services of a node can be broadly classified into the following three categories: (1) routing/forwarding related, (2) QoS related, and (3) security related. For instance, the routing and forwarding functions are governed by routing protocols. A robust routing protocol reduces packet loss rates, eliminates the possibility of having frequent route failures, and is able to cope quickly against topological changes [14]. Several attributes for routing and forwarding are highlighted in [14] such as:

- A node does not deliberately introduce latency to delay the arrival and departure of packets.
- A node does not intentionally drop packets.
- A node does not intentionally generate duplicate copies of a packet or inserts false packet/s into a packet stream with the intention of either depleting the network bandwidth or for the purposes of misleading other nodes.

As a result, the observed attributes could include the packet drop rate, the number of duplicate packets received, and latency (delay / delay jitter)

Trust can be represented in several ways. Some researchers represent trust as a range of values in real numbers varying from 0 (corresponding to complete distrust) to 1 (corresponding to complete trust) [26], [12], [27]. Some others consider trust values to be ranging from -1 (corresponding to complete distrust) to 1 (corresponding to complete trust) [28]. In [8], trust values have been represented by an integer in the interval between 0 and 100. This is because the wireless sensor networks have limited memory, transmission and power. An unsigned integer uses 1 byte while a real number uses 4 bytes.

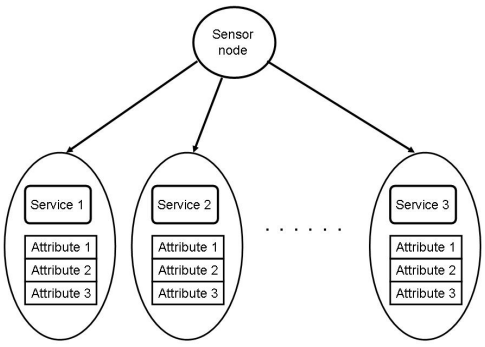

Fig. 3. Nodes and Their Attributes

This means that the representation of trust values [0, 100] can save up 75 percent of memory space. As a result, the data transmission between nodes is also reduced accordingly. Consequently, the power consumption can also be decreased. For the same reason, we also represent trust values as an integer between 0 and 100.

B. Node Level Trust Management

In our framework, the direct trust value between a sponsor node s and a target node t is denoted as $T_{s,t}$. Group trust value is denoted by G_t (calculated from the direct trust values of all the nodes in the cluster) and integrated trust value is denoted by I_t (calculated based on the direct trust value $T_{ch,t}$ of the cluster node and the group trust value G_t).

The direct trust value of a target node is calculated based on its multi-attribute trust values. The sponsor node evaluates and records the result of the cooperation with the target node. The cooperation records are listed as shown in Table I. Each attribute $(A_i, i = 1, 2...n)$ has two relevant values: the value of the successes $(S_i, i = 1, 2, ...n)$, and the value of the cumulative cooperations $(CC_i, i = 1, 2...n)$. For example, the cummulative cooperations are $CC_1 = 10$, the S_1 can be any number between 0 to 10.

TABLE I
COOPERATION RECORD TABLE

Attribute	Number of Successes	Number of Cooperations
A_1	S_1	CC_1
A_2	S_2	CC_2
...	...	...
A_n	S_n	CC_n

Based on the cooperation success records, we can calculate the trust value for attribute A_i as follows:

$$t_{A_i} = [100 * \frac{S_i}{CC_i}] \tag{6}$$

where $0 \leq t_{A_i} < 100$ and $[.]$ is the nearest integer function.

We divide time into Time Units. Individual sensor nodes keep track of cooperation records with the other sensor nodes within a certain time frame. The size of the time frame is critical for trust evaluation. The duration of this time frame can be dynamically determined and is influenced by factors such as application requirements and/or resource constraints of each sensor node. At time T_k, the trust value for attribute i is $T_{A_{ik}}$ and during the time unit t_{k+1}, the trust value is $t_{A_{ik+1}}$.

The trust value at time T_{k+1} can be calculated as follows:

$$T_{A_{ik+1}} = [(100 - \theta)/100 * T_{A_{ik}} + \theta/100 * t_{A_{ik+1}}] \tag{7}$$

where $0 \leq \theta < 100$ is the impact factor and $[.]$ is the nearest integer function. With this formula, the new trust value at time T_{k+1} results from the trust value at T_k and the trust value during the new time unit t_{k+1}.

By adopting the trust evaluation approach proposed in [29] for service oriented environments, we use the following function to compute the impact factor:

$$\theta = \lambda * f'(t_{A_{ik}}/100) \tag{8}$$

where $\lambda > 0$ is the scale control factor; $f'(x) \geq 0$ is the derivative of the function $f(x)$ (transformation of hyperbolic tangent):

$$f(x) = \frac{(e^{\alpha x} - e^{-\alpha x})}{\beta(e^{\alpha x} + e^{-\alpha x})} \ (\alpha \geq 1, \beta \geq 1) \tag{9}$$

The derivatives of the function $f(x)$ where $\alpha = 2$ and $\beta = 4, 8, 16$ are plotted in Figure 4. By adjusting the α and β values, we can define an impact factor that may lead to quicker or slower trust improvement/degradation. The smaller α and β values can lead to a quicker change in trust while larger α and β values can lead to a slower change. We can change the factor value λ to adjust this rate of change dynamically while satisfying $0 \leq \theta < 100$.

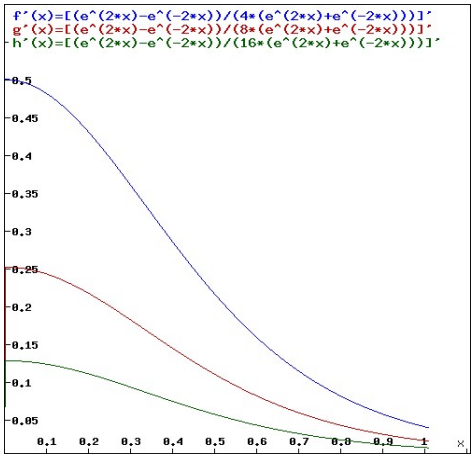

Fig. 4. The plot of the derivatives of $f(x)$ where $\alpha = 2$ and $\beta = 4, 8, 16$

Moreover, when the trust value in the latest time unit is high, the new overall trust value will change by a small amount. On the other hand, if the trust value is low, the overall trust value will go down sharply because the old trust value will decay fast. This will result in much lower trust values for those nodes which turn malicious at the next time unit.

Assuming the parameter setting of $\alpha = 2$, $\beta = 4$ and $\lambda = 100$, the lowermost line in Figure 5 plots the impact factor θ (y) as the trust value in the next time unit (x) ranges from 0 to 100. Suppose that for a given sensor node the overall trust value $T_{A_{ik}} = 90$. The topmost line then plots the next overall trust value (y) as the trust value in the next time unit (x) ranges from 0 to 100.

If the node turns malicious in the next time unit with $t_{A_{ik+1}} = 20$, then $\theta = 42.8$ (resulting from the parameter setting of $\alpha = 2$, $\beta = 4$ and $\lambda = 100$). The next trust value is $T_{A_{ik+1}} = [(100 - \theta)/100 * T_{A_{ik}} + \theta/100 * t_{A_{ik+1}}] = 0.572 * 90 + 0.428 * 20 = 60.04$, representing a decrease of almost 30 points in the overall trust value.

Now we are able to calculate the overall direct trust value for the target node t with n attributes $A_i, i = 1, , 2, ...n$ for a given sponsor node s as follows:

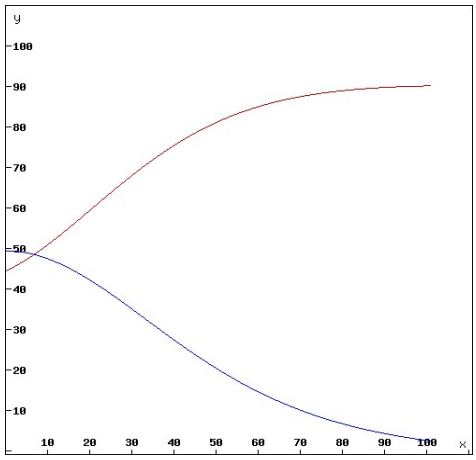

Fig. 5. The plot of θ and the next overall trust value based on the trust value in the next time unit (x) where $\alpha = 2$, $\beta = 4$ and $\lambda = 100$

$$T_{s,t} = \frac{\prod_{i=1}^{n} T_{A_i}}{\prod_{i=1}^{n} T_{A_i} + \prod_{i=1}^{n} (1 - T_{A_i})} \tag{10}$$

C. Cluster Head Level Trust Management

We assume that the cluster heads are the super nodes with more memory, higher computation power and more energy than the other member nodes. A cluster head has several roles. Within its cluster, a cluster head stores all the trust records of the nodes in its cluster, and furnishes the integrated trust value to a requesting node . Within the wireless sensor network, a cluster head also records other cluster head trust values. The cluster head is also in charge of transferring the trust record of a node when the node moves from one cluster to another. We now discuss each of these functions in more detail.

1) Intra-Cluster Trust Management: As mentioned before, a cluster head stores all trust records of the nodes , computes the integrated trust values for each node in its cluster, and provides the integrated trust value to nodes upon request.

Each node transmits the recorded trust values that it has for other target nodes to the cluster head. The cluster head stores all the records for calculating the node's group trust value. Suppose there are $n + 1$ nodes in the group including the cluster head. Each of the group member nodes forwards its direct trust values of the other member nodes to the cluster head when one of the trust values is changed. The cluster head maintains these trust values in a matrix form, as shown below:

$$T_{ch} = \begin{bmatrix} T_{1,2} & T_{2,1} & ... & T_{n,1} \\ T_{1,3} & T_{2,3} & ... & T_{n,2} \\ \vdots & \vdots & \vdots & \vdots \\ T_{1,n} & T_{2,n} & ... & T_{n,n-1} \end{bmatrix}$$

Each column in the above matrix corresponds to the vector of trust values for the corresponding sensor node. For example, $T_{1,2}$ is the node 2's direct trust value sent by node 1.

a) Integrated Trust Computation: The integrated trust value computation of a node involves three: group trust values, cluster head's trust value, the base station's trust value. The group trust values can be derived from the above matrix. The cluster head has its trust value for each node in its cluster group as well. The base station has a trust value for each node that resides in the wireless sensor network (a collection of clusters) as it is the only node in the network that has the complete knowledge necessary to evaluate and assess every function that a sensor node is capable of performing.

Each cluster head keeps track of a trust matrix:

$$T_G = \begin{bmatrix} G_1 & T_{ch,1} & T_{b,1} \\ \vdots & \vdots & \vdots \\ G_i & T_{ch,i} & T_{b,i} \\ \vdots & \vdots & \vdots \\ G_n & T_{ch,n} & T_{b,n} \end{bmatrix}$$

where $i = 1, ...n$ and the G_i, $T_{ch,i}$, $T_{b,i}$ are the group trust value of node i, the cluster head's direct trust value for i and the base station's trust value of i.

The group trust value G_i for node i is calculated as follows:

$$G_i = \frac{\prod_{j=1,...,i-1,i+1,...,n} T_{j,i}}{\prod_{j=1,...,i-1,i+1,...,n} T_{j,i} + \prod_{j=1,...,i-1,i+1,...,n} (1 - T_{j,i})} \tag{11}$$

Based on the service trust record values of all other nodes, the cluster head can then compute the integrated trust value I_i for each node as follows:

$$I_i = G_i * W_{group} + T_{ch,i} * W_{ch} + T_{b,i} * W_{base}$$

Here W_{group}, W_{ch} and W_{base} are the weights for the corresponding trust values and they can be adjusted dynamically by the cluster head. The weights must add up to 1.

b) Transfer of Integrated Trust to Sponsor Node: We require that the cluster head also keeps track of the energy levels of the sensor nodes in its cluster in a vector $[E_1, E_2, \ldots, E_n]$ which is used in service selection and load balancing. When a sponsor node does not have the direct trust value for a target node, it will request the target nodes' trust values from the cluster head. The cluster head will then send the integrated trust values for all the requested nodes, along with their energy levels. The sponsor node is then able to find the path with higher trust value to perform the required service with the node with the sufficient energy level. For example, if the sponsor node is to decide between two paths, involving A and B, and involving C and D, it will request the trust values and energy levels of A, B, C and D from the cluster head. The sponsor node is then able to calculate the trust value for each path and then choose the path with the highest trust value but it will also take into consideration the minimum trust value assigned to a node in each path along with the energy levels associated with each node.

2) Inter-Cluster Trust Management: We assume that the nodes (apart from the cluster heads) in the wireless sensor network are dynamic. They can move from one cluster group to another cluster group. We are currently in the process of developing a full trust management scheme for such inter-cluster movements. At this stage, our inter-cluster trust management framework has the following characteristics.

When an existing node leaves one group and joins a new group, the cluster head will send the records of this node to the new cluster head of the group cluster the node is going to join. The old cluster head will also send the trust related information to the base station and finally delete the nodes' records from its database.

When a new node joins a cluster, first it will check with the base station whether it knows anything about this particular node. If it does not, then we use the pre-deployment knowledge based on certificates to establish trust. Each node is assumed to have a public key based certificate and the trust association between the node and the cluster head is established using this certificate. We set the trust value based on just the certificate based trust alone to 50.

IV. Comparative Evaluation

In this section, we compare our proposed trust management framework (TMF) with the previously proposed GTMS framework in terms of computation, memory requirements and communication overhead as only these two schemes have the similar architecture (see Table II for a summary).

TABLE II
Comparison of TMF and GTMS [9]

	GTMS	TMF
Communications among nodes for trust calculation	$n*(n-m)$	1
Transmission length between nodes and cluster head	2 bits	2 bytes
Communication times for nodes and cluster head	Regular	on Demand
Communications for cluster heads and the base station	Roughly Same	Roughly Same
Memory overhead for nodes	More	Less
Memory overhead for cluster head	Less	More
Computation overhead for nodes	More	Less
Computation overhead for cluster heads	Less	More
Trust decay	Yes	Better
Energy level for nodes	No	Yes
Pre-deployment knowledge certificates	No	Yes
Enhancing cluster head trust management	No	Yes
Dynamic node movement	No	Yes
Multi-hop routing	No	Yes

a) Communication Overhead: In both GTMS and TMF, the direct trust of the node is calculated based on the past experiences without requesting any information from other nodes. With respect to indirect trust, in GTMS, the sponsor node needs to collect all the targets nodes' trust values from all other nodes in the cluster. If there are n nodes and m targets nodes, then the total number of communications can reach up to as many as $n*(n-m)$. By contrast, in our framework, the sponsor node only needs to communicate with the cluster head once to obtain the group trust value.

For communication between the cluster head and nodes in the same cluster group, in GTMS, the cluster head periodically broadcasts the request packets. All the nodes in that group will forward the trust states of other nodes to the cluster head. As there are three states (trusted, uncertain, and untrusted), only 2 bits are needed for this purpose. In our framework, we use the trust value from 0 to 100, and hence 1 byte is needed. We assume that the node will send the new trust value to the cluster head when the trust values for other nodes are changed. Hence, this will require less communication overhead. The node will also need to send its energy level to the cluster head periodically, and this will incur extra communication overhead (1 byte is required for the energy levels ranging from 0 to 100).

b) Storage Overhead: In both GTMS and TMF, the nodes need to store the successful and unsuccessful interaction numbers for each node. Additionally in GTMS, the nodes need to store indirect trust values for other peer nodes. In TMF, for a cluster group with n nodes, the cluster head needs to store an $(n-1)*(n-1)$ matrix and a $4*n$ matrix to calculate the integrated trust value of each node in the group, together with one vector of size n for the energy levels. By contrast, in GTMS, the cluster head only needs to store one $n*n$ matrix.

c) Computation Overhead: In terms of computation at the node, in both GTMS and TMF, a node needs to calculate the direct trust value based on its interaction experiences with other nodes. However, in GTMS, a node also needs to calculate the target node trust values based on the peer nodes trust value. In terms of cluster head computation, in TMF, the cluster head needs to compute the integrated trust values based on one $(n-1)*(n-1)$ matrix and one $4*n$ matrix, while, in GTMS, the cluster calculates the integrated trust value for each node based on one $(n-1)*(n-1)$ matrix.

d) Trust Value Calculation with Decay: In our framework, we calculate the new trust value at time T_{k+1} based on two trust values: the trust value at time T_k and the trust value during the last time unit. TMF has two advantages over GTMS. Firstly, the node only needs to store two trust values: the trust value at time T_k and the trust value during the last time unit. On the other hand, in GTMS, the trust value calculation is based on several previous time units. Thus, nodes need to store all the trust values in several time units. Secondly, in GTMS, each of the trust values has the same weight. By contrast, in TMF, based on the two factors λ and θ, the trust value of the most recent time unit can have more weight in the overall trust value calculation; the older trust value may take less weight in the overall trust value and decay accordingly.

e) Nodes Energy Level Balance and Others: In TMF, we also consider the energy level of nodes as an additional factor in service selection. This is because that the node with higher trust value will be employed to do more services. As a result, its energy will be consumed faster. By propagating the energy level of a target node along with its trust value, a recipient can decide whether or not to use the services of this target node. On the other hand, in GTMS, there is no such an arrangement.

f) Pre-deployment Knowledge: In TMF, we also take advantage of the pre-deployment knowledge. As the sensor networks have infrastructure components such as a base station which can perform centralized management, we can employ certificates to validate new nodes. We assume that the base station and the cluster head can perform advanced trust management because we assume that cluster heads and base station have more power for computation and communication and are trusted at a higher level.

V. Conclusions

In this paper, we have proposed a dynamic trust management framework for hierarchical wireless sensor networks. In our scheme, we have reduced the computation and communication requirements of sensor nodes in carrying out trust evaluation. Our scheme incorporates a time window and a decay function that captures the dynamic nature of trust in trust calculations. Finally, we have shown that our scheme has improved features to support mobile sensor network environments over those of the previously proposed scheme GTMS. In addition, our trust management framework has the ability to consider movement of nodes from one cluster to another. Currently, we have been developing a complete model of inter-cluster movement of nodes in mobile inter-cluster wireless sensor environments.

Acknowledgements

This research has been supported in part by an Australian Research Council (ARC) Discovery grant (DP0452628) and a Macquarie University Research Development Grant (MQRDG).

References

[1] N. Sklavos and X. Zhang, *Wireless Security and Cryptography: Specifications and Implementations*. A Taylor and Francis Group, ISBN: 084938771X: CRC-Press, 2007.

[2] S. Ganeriwal and M. Srivastava, "Reputation-based framework for high integrity sensor networks," in *Proc. ACM Workshop Security of Ad Hoc and Sensor Networks (SASN q́04)*, Oct. 2004, pp. 66–67.

[3] A. Srinivasan, J. Teitelbaum, and J. Wu, "Drbts: Distributed reputation-based beacon trust system," in *the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06)*, 2006.

[4] A. Boukerche, X. Li, and K. EL-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Computer Comm.*, pp. 2413–2427, Sept. 2007.

[5] E. Aivaloglou, S. Gritzalis, and C. Skianis, "Towards a flexible trust establishment framework for sensor networks," *Telecommun Syst*, vol. Vol. 35, pp. 207–213, 2007.

[6] ——, "Trust establishment in sensor networks: behaviour-based, certificate-based and a combinational approach," *Int. J. System of Systems Engineering*, vol. Vol. 1, Nos. 1/2, pp. 128–148, 2008.

[7] O. Mistry, A. Gürsel, and S. Sen, "Comparing trust mechanisms for monitoring aggregator nodes in sensor networks," in *Proc. of the 8th Int, Conf. on Autonomous Agents and Multiagent Systems*, May 2009, pp. 985–992.

[8] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, and S. L. andYoung Jae Song, "Group-based trust management scheme for clustered wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, pp. 1698–1712, 2009.

[9] J. Zhang, R. Shankaran, M. A. Orgun, V. Varadharajan, and A. Sattar, "A trust management architecture for hierarchical wireless sensor networks," in *Proc. of The 35th Annual IEEE Conference on Local Computer Networks, LCN 2010,*. IEEE Computer Society, 11-14 October 2010, Denver, Colorado, U.S.A. (to appear).

[10] A. Srinivasan, J. Teitelbaum, and J. Wu, "Reputation and trust-based systems for ad-hoc and sensor networks," http://www.cse.fau.edu/ jie/research/publications/, 2007.

[11] Z. Liu, A. W. Joy, and R. A. Thompson, "A dynamic trust model for mobile ad hoc networks," *Future Trends of Distributed Computing Systems, IEEE International Workshop*, vol. 0, pp. 80–85, 2004.

[12] G. Theodorakopoulos and J. S. Baras, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 318–328, 2006.

[13] V. Balakrishnan, V. Varadharajan, P. Lucs, and U. K. Tupakula, "Trust enhanced secure mobile ad hoc network routing," in *Proc. of the 21st IEEE International Conference on Advanced Information Networking and Applications Workshops(AINAW 2007)*, May 2007, pp. 27–33.

[14] R. Shankaran, V. Varadharajan, M. A. Orgun, and M. Hitchens, "Context-aware trust management for peer-to-peer mobile ad-hoc networks," in *Proc. of the 33rd Annual IEEE International Computer Software and Applications Conference, Seattle, USA, Volume 2*. IEEE Computer Society, 20-24,July 2009, pp. 188–193.

[15] E. Aivaloglou, S. Gritzalis, and C. Skianis, "Trust establishment in ad-hoc and sensor networks," in *Proc. of 1st International Workshop on Critical Information Infrastructure Security (CRITIS 06)*, LNCS 4347, 2006, pp. 179–194.

[16] M. Mejia, N. Peña, J. L. Muñoz, and O. Esparza, "A review of trust modeling in ad hoc networks," *Internet Research*, vol. Vol.2009, 2009.

[17] G. Han, L. Shu, J. Ma, J. H. Park, and J. Ni, "Power-aware and reliable sensor selection based on trust for wireless sensor networks," *Journal of Communications*, vol. 5, pp. 23–30, January 2010.

[18] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks," in *Proc. of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'2002)*, October 2002. (Forth Worth, TX), p. 129.

[19] K. Arisha, M. Youssef, and M. Younis, "Energy-aware TDMA-based MAC for sensor networks," in *the IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002)*. IEEE Computer Society, May 2002, p. 129.

[20] G. Jolly, M. C. Kuşçu, P. Kokate, and M. Younis, "A low-energy key management protocol for wireless sensor networks," in *Proc. of the 8th IEEE International Symposium on Computers and Communication (ISCC'03)*, June 2003, p. 335.

[21] M. Younis, K. Ghumman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered sensor networks," *IEEE Trans. Parallel and Distrib. Sys*, vol. 17, pp. 865 – 882, 2006.

[22] M. Eltoweissy, M. Moharrum, and R. Mukkamala, "Dynamic key management in sensor networks," *IEEE Communications Magazine*, vol. 44, pp. 122–130, 2006.

[23] G. Gupta and M. Younis, "Performance evaluation of load-balanced clustering of wireless sensor networks," in *the Proc. of the 10th IEEE International Conference on Telecommunications (ICT'2003)*, Feb. 2003, pp. 1577–1581.

[24] ——, "Performance evaluation of load- balanced clustering of wireless sensor networks," in *the 10th International Conference on Telecommunications (ICT'2003)*, Tahiti, Papeete, French Polynesia, February 2003, pp. 1577–1581.

[25] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 366–379, Oct.-Dec. 2004.

[26] Y. Sun, W. Yu, Z. Han, and K. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE Journal on Selected Areas in Communication*, vol. 24, pp. 305–317, Feb. 2006.

[27] H. Jameel, L. X. Hung, U. Kalim, A. Sajjad, S. Lee, and Y.-K. Lee, "A trust model for ubiquitous systems based on vectors of trust values," in *the Proc. of the Seventh IEEE International Symposium on Multimedia*. IEEE Computer Society, 2005, pp. 674–679.

[28] A. Pirzada and C. McDonald, "Establishing trust in pure ad-hoc networks," in *Proc. 27th Australasian Computer Science Conf. (ACSC' 04)*, June 2004, pp. 47–54.

[29] Y. Wang, D. S. Wong, K.-J. Lin, and V. Varadharajan, "The design of a rule-based and event-driven trust management framework," in *Proc. of IEEE International Conference on e-Business Engineering*. IEEE Computer Society, 2007, pp. 97–104.